

Installation and Deployment Guide

Netscape Certificate Management System

Version 4.2

151-09680-01

April 2000

Copyright © 2000 Sun Microsystems, Inc. Some preexisting portions Copyright © 2000 Netscape Communications Corporation. All rights reserved.

Sun, Sun Microsystems, the Sun logo, iPlanet, the iPlanet logo, Java, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Netscape and the Netscape N logo are registered trademarks of Netscape Communications Corporation in the U.S. and other countries. Other Netscape logos, product names, and service names are also trademarks of Netscape Communications Corporation, which may be registered in other countries.

This product contains Validation Authority (VA™) software from ValiCert, Inc. Copyright © 2000 ValiCert, Inc. All rights reserved.

Federal Acquisitions: Commercial Software—Government Users Subject to Standard License Terms and Conditions

The product described in this document is distributed under licenses restricting its use, copying, distribution, and decompilation. No part of the product or this document may be reproduced in any form by any means without prior written authorization of the Sun-Netscape Alliance and its licensors, if any.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright © 2000 Sun Microsystems, Inc. Pour certaines parties préexistantes, Copyright © 2000 Netscape Communication Corp. Tous droits réservés.

Sun, Sun Microsystems, iPlanet, the iPlanet logo, Java, et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-Unis et d'autres pays.

Netscape et the Netscape N logos sont des marques déposées de Netscape Communications Corporation aux États-Unis et d'autres pays. Les autres logos, les noms de produit, et les noms de service de Netscape sont des marques déposées de Netscape Communications Corporation dans certains autres pays.

Ce produit contient des logiciels Validation Authority (VA™) de ValiCert, Inc. Copyright © 2000 ValiCert, Inc. Tous droits réservés.

Le produit décrit dans ce document est distribué selon des conditions de licence qui en restreignent l'utilisation, la copie, la distribution et la décompilation. Aucune partie de ce produit ni de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit sans l'autorisation écrite préalable de l'Alliance Sun-Netscape et, le cas échéant, de ses bailleurs de licence.

CETTE DOCUMENTATION EST FOURNIE "EN L'ÉTAT", ET TOUTES CONDITIONS EXPRESSES OU IMPLICITES, TOUTES REPRÉSENTATIONS ET TOUTES GARANTIES, Y COMPRIS TOUTE GARANTIE IMPLICITE D'APTITUDE À LA VENTE, OU À UN BUT PARTICULIER OU DE NON CONTREFAÇON SONT EXCLUES, EXCEPTÉ DANS LA MESURE OÙ DE TELLES EXCLUSIONS SERAIENT CONTRAIRES À LA LOI.

Contents

About This Guide	13
What You Should Already Know	13
What's in This Guide	14
Conventions Used in This Guide	15
Where to Go for Related Information	16
Part 1 Overview and Demo Installation	
Chapter 1 Introduction to Certificate Management System	21
System Overview	22
Public-Key Infrastructure	23
Subsystems of Certificate Management System	24
Basic System Configuration	26
Authentication and Policy Modules	29
Authentication Modules	29
Policy Modules	30
Steps in End-Entity Enrollment	30
Some Enrollment Scenarios	33
Firewall Considerations	34
Extranet/E-Commerce:	
Acme Sales Corp.	35
Enrolling Existing Customers	36
Enrolling New Customers	38
Enrolling Extranet Users	40
PIN Registration: Atlas Manufacturing	42
VPN Client Enrollment and Revocation	44
Router Enrollment and Revocation	46
End Entities and Life-Cycle Management	48
Life-Cycle Management Formats and Protocols	48
Access to Subsystems	50

HTML Forms for End Users	52
Netscape Personal Security Manager	53
Summary of System Features	55
Authentication Modules	55
Policy Modules	56
Job Scheduler Plug-Ins	61
Event-Driven Notifications	61
Registration Manager	62
Certificate Manager	62
Signing Algorithms	63
Certificate Revocation Lists	63
Data Recovery Manager	64
Command-Line Utilities	68
System Architecture	70
PKCS #11	72
NSS	73
JSS and the Java/JNI Layer	73
Middleware/JDK 1.1.6 Layers	73
Authentication and Policy Modules	74
Standards Summary	74
Certificate Management Formats and Protocols	74
Security and Directory Protocols	75
Chapter 2 Default Demo Installation	77
System Requirements	78
Operating System and Software Required	78
Platform Requirements	78
UNIX Platform Requirements	78
Windows NT Platform Requirements	79
Other Requirements	79
Overview of the Default Demo	80
Demo Passwords	83
Installing the Default Demo	84
Step 1. Run the Installation Script - Unix	85

Step 1. Run the Installation Script - Windows NT	87
Step 2. Run the Installation Wizard	89
Step 3. Get the First User Certificate	93
Enrolling for the First Agent Certificate	93
If You Need the First Agent Form Again	95
Using the Default Demo	95
Verify the Installation	96
Viewing Issued Certificates From the Agent Gateway	97
Enrolling for a Certificate From the End-Entity Gateway	98
Finding and Approving a Certificate Request	99
Setting Your Browser to Use the Agent Certificate	100
Testing Your New Certificate	100
Create a Policy	101
Configuring an RSA Key Length Policy	102
Use an LDAP Directory	103
Enable Directory-Based Authentication	103
Add a User to the Directory	104
Enroll with Directory-Based Authentication	105
Publish Certificates in a Directory	106
Configure the Publishing Destination	107
Set Rules for Publishing Certificates	108
Update the Publishing Directory	109
Send Renewal Reminders	111
Configuring a Mail Server for CMS	111
Configuring CMS to Send Renewal Reminders	112

Part 2 Planning and Installation

Chapter 3 Planning Your Deployment	117
Topology Decisions	118
Server Groups and CMS Instances	118
Single Certificate Manager	119
Certificate Manager and Registration Manager	120
Certificate Manager and Data Recovery Manager	123
Certificate Manager, Data Recovery Manager, and Registration Manager	125

Cloned Certificate Manager	127
Certificate Authority Decisions	128
CA's Distinguished Name	128
CA Signing Key Type and Length	129
CA Signing Certificate's Validity Period	129
Self-Signed Root Versus Subordinate CA	130
CAs and Certificate Extensions	130
CA Certificate Renewal or Reissuance	131
Cryptographic Token Decisions	132
Publishing Decisions	133
Subsystem Certificate Decisions	134
SSL Server Certificates	135
Certificate Manager Certificates	135
Registration Manager Certificates	136
Data Recovery Manager Certificate and Storage Key	136
Authentication Decisions	137
Policy Decisions	137
Deployment Strategy and Port Assignments	138
Chapter 4 Installation Worksheet	141
Information for Unix Installation Script	142
Installation Location	142
Configuration Directory Server	142
User/Group Directory Server	143
Configuration Directory Settings	144
Administration Server Information	145
Certificate Management System Identifier	145
Information for NT Installation Script	145
Installation Directory	145
Configuration Directory Server	146
User/Group Directory Server	146
Configuration Directory Settings	147
Configuration Directory Server Administrator	147
Directory Server Administration Domain	148

Directory Manager Settings	148
Administration Server Port	148
Certificate Management System Identifier	148
Initial Configuration	149
Internal Database	149
Administrator	150
Subsystems	150
Remote Certificate Manager	150
Remote Data Recovery Manager	151
Network Configuration	151
Certificate Manager Configuration	151
Server Migration from Certificate Server 1.x	152
Migration Tool Output Files	152
Token for CA Signing Certificate	152
Token for SSL Server Certificate	153
CA Signing Certificate	153
CA's Serial Number Range	153
Key-Pair Information for CA Signing Certificate	154
Subject Name for CA Signing Certificate	154
Validity Period for CA Signing Certificate	155
Extensions for CA Signing Certificate	155
CA Signing Certificate Request	157
Registration Manager Configuration	157
Registration Manager Signing Certificate Request	157
Key-Pair Information for Registration Manager Signing Certificate	158
Subject Name for Registration Manager Signing Certificate	158
Registration Manager Signing Certificate Issuer	159
Data Recovery Manager Configuration	159
Transport Certificate	160
Key-Pair Information for Transport Certificate	160
Subject Name for Transport Certificate	160
Validity Period for Transport Certificate	161
Extensions for Transport Certificate	161

Transport Certificate Request	162
Storage Key and Recovery Agent Configuration	163
Storage Key Creation	163
Data Recovery Scheme - 1	163
Data Recovery Scheme - 2	164
Cloned Certificate Manager Configuration	164
Server Migration from Certificate Server 1.x	165
Migration Tool Output Files	165
Token for CA Signing Certificate	165
Token for SSL Server Certificate	166
CA Signing Certificate	166
CA's Serial Number Range	166
Cloned Key and Certificate Material	167
SSL Server Key and Certificate	167
SSL Server Certificate Configuration	167
SSL Server Certificate	168
Key-Pair Information for SSL Server Certificate	168
Subject Name for SSL Server Certificate	168
Validity Period for SSL Server Certificate	169
Extensions for SSL Server Certificate	169
SSL Certificate Request	170
Single Sign-On Password	171
Chapter 5 Installation and Configuration	173
Installation Overview	174
Installation Stages	174
Stage 1: Running the Installation Script	175
Running the Installation Script on Unix	176
Running the Installation Script on Windows NT	179
Stage 2: Using the Installation Wizard	182
Initial Configuration	183
Certificate Manager Configuration	186
Self-Signed CA Certificate	187
Subordinate CA Certificate Request	189

Registration Manager Configuration	192
Data Recovery Manager Configuration	196
Transport Certificate from a Remote CA	196
Storage Key and Recovery Agent Configuration	200
Certificate Manager and Data Recovery Manager Configuration	200
Certificate Manager Configuration	201
Data Recovery Manager Configuration	206
Registration Manager and Data Recovery Manager Configuration	211
Registration Manager Configuration	212
Data Recovery Manager Configuration	215
Cloned Certificate Manager Configuration	219
SSL Certificate Configuration	220
SSL Server Certificate from the Local CA	220
SSL Server Certificate from a Remote CA	221
Single Signon Configuration	224
Additional Steps	225
Administrator/Agent Certificate Enrollment	226
Stage 3: Further Configuration Options	229
Stage 4: Creating Additional Instances or Certificate Manager Clones	229
First Agent for an Additional CMS Instance	230
Appendix A Migrating from Certificate Server 1.x	233
Using the Migration Tool	233
Command-Line Syntax	234
Arguments	235
The Migration Process	235
Entering Informix Database Login Information	236
Entering Key and Certificate Database Passwords	237
Exit Codes and Error Messages	238
Generated Files	241
Using Output Files on Another Platform	242
Converting Output Files	242

Importing Data to New Databases	244
Hardware, Operating System, and Version Support	245
Appendix B Certificate Extensions	247
Introduction to Certificate Extensions	247
Recommendations for Extension Use	249
Standard X.509 v3 Certificate Extensions	253
authorityInfoAccess	254
authorityKeyIdentifier	256
basicConstraints	257
certificatePolicies	258
cRLDistributionPoints	259
extKeyUsage	261
issuerAltName	263
keyUsage	264
nameConstraints	267
OCSPNocheck	268
policyConstraints	269
policyMappings	270
privateKeyUsagePeriod	271
subjectAltName	272
subjectDirectoryAttributes	273
subjectKeyIdentifier	274
Standard X.509 v3 CRL Extensions	275
Extensions for CRLs	276
authorityKeyIdentifier	276
CRLNumber	276
deltaCRLIndicator	277
issuerAltName	278
issuingDistributionPoint	278
CRL Entry Extensions	279
certificateIssuer	279
holdInstructionCode	279
invalidityDate	280

reasonCode	280
Netscape-Defined Certificate Extensions	280
netscape-cert-type	281
netscape-comment	282
Adding Extensions in Certificate Management System	282
CA Certificates and Extension Interactions	282
Appendix C Certificate Download Specification	285
Data Formats	285
Binary Formats	285
Text Formats	286
Importing Certificate Chains	286
Importing Certificates into Netscape Communicator	287
Importing Certificates into Netscape Servers	288
Object Identifiers	288
Appendix D Using SSL with iPlanet Web Server, Enterprise Edition	
4.x	291
Creating a New Server	292
Obtaining a Server Certificate	293
Creating a Trust Database	293
Submitting a Certificate Signing Request	294
Importing the Certificate	296
Enabling SSL on the Server	299
Enabling Encryption on the Server	300
Trusting the Root CA Certificate	300
Enabling Client Authentication for All Requests	302
Specifying the Authentication Directory	302
Note for CGI Programmers	304
Modifying the Configuration File	304
Modifying the Access Control Lists	306

Testing Client Authentication	308
Appendix E Export Control Information	311
Approved Export Operations and Key Sizes	312
SSL Cipher Suite Profiles for Export	315
Glossary	317
Index	331

About This Guide

The *Installation and Deployment Guide* describes how to plan for the deployment of a public-key infrastructure (PKI) using Netscape Certificate Management System (CMS). It provides the basic information required to set up a simple default pilot, make basic deployment decisions, and install and configure CMS subsystems.

This preface has the following sections:

- What You Should Already Know
- What's in This Guide
- Conventions Used in This Guide
- Where to Go for Related Information

What You Should Already Know

This guide is intended for experienced system administrators who are planning to deploy Netscape Certificate Management System. CMS agents should refer to *Netscape Certificate Management System Agent's Guide* for information on how to perform agent tasks, such as handling certificate requests and revoking certificates.

Before reading this guide, you should be familiar with the role of Netscape Console in managing Netscape servers. For background information, see the accompanying manual, *Managing Servers with Netscape Console*.

In particular, you must be familiar with the basic concepts of public-key cryptography and the Secure Sockets Layer (SSL) protocol before you attempt to install and use Certificate Management System. These include the following topics:

- encryption and decryption
- public keys, private keys, and symmetric keys
- digital signatures
- the role of digital certificates in a public-key infrastructure (PKI)
- certificate hierarchies

What's in This Guide

- SSL cipher suites
- the purpose of and major steps in the SSL handshake

For an overview of these topics, see Appendix D and Appendix E of *Managing Servers with Netscape Console*.

What's in This Guide

This guide covers the following topics:

Part 1, Overview and Demo Installation

- Chapter 1, "Introduction to Certificate Management System." Provides an overview of the Certificate Management System architecture for creating, deploying, and managing certificates.
- Chapter 2, "Default Demo Installation." Describes how to set up a simple pilot that demonstrates the basic capabilities of a Certificate Manager with an integrated Registration Manager.

Part 2, Planning and Installation

- Chapter 3, "Planning Your Deployment." Reviews basic decisions you should make as you plan your initial deployment.
- Chapter 4, "Installation Worksheet." Provides a worksheet you can copy and use to collect the detailed information that you will need to provide during installation and configuration of individual subsystems.
- Chapter 5, "Installation and Configuration." Describes the procedure for installing CMS subsystems on the basis of the information collected in Chapter 4.

Appendixes and Glossary

- Appendix A, "Migrating from Certificate Server 1.x." Describes how to use the Migration Tool that comes with Certificate Management System.
- Appendix B, "Certificate Extensions." Summarizes the standard certificate extensions defined by X.509 version 3 and the extensions defined by Netscape before this version was finalized. Recommends extensions to use

with specific kinds of certificates, including both PKIX Part 1 recommendations and Netscape extensions that must be supported to maintain compatibility with early versions of Netscape products.

- Appendix C, “Certificate Download Specification.” Describes the data formats used by Netscape Communicator 4.x for installing certificates.
- Appendix D, “Using SSL with iPlanet Web Server, Enterprise Edition 4.x.” Explains how to set up client certificate authentication to work with Netscape Enterprise Server 3.x.
- Appendix E, “Export Control Information.” Summarizes the cryptographic operations, key lengths, and cipher suites that have received US government approval for the export version of Certificate Management System.
- Glossary. Summarizes terms used in this guide and other CMS documentation.

Conventions Used in This Guide

This guide uses the following conventions:

- `Monospaced font`
This typeface is used for text that is an executable part of a program or text that you type. It's also used for filenames, directory names, and URLs.
- *Italic*
Italic type is used for emphasis and to introduce new terms.
- Square brackets []
Square brackets enclose commands that are optional.
- Angle brackets <>
Angle brackets indicate placeholders for items that vary, such as pathnames and variable names. Replace the angle brackets and their text with text that applies to your situation.

Where to Go for Related Information

- Slash /

A slash is used to separate directories in a path. (Note that the Windows NT operating system supports both the slash and the backslash.)

This guide also contains the following special notes:

- Note** You can use Netscape Console only when Administration Server is up and running.
- Caution** A caution noted documents a potential risk of losing data, damaging software or hardware, or otherwise disrupting system performance.
- Unix** Marks text that applies only to the Unix versions of Certificate Management System.
- NT** Marks text that applies only to the Windows NT versions of Certificate Management System.

Where to Go for Related Information

This section summarizes the documentation that ships with Certificate Management System, using these conventions:

- `<server_root>` is the directory where the CMS binaries are kept (specified during installation).
- `<instance_id>` is the ID for this instance of Certificate Management System (specified during installation).

The documentation set for Certificate Management System includes the following:

- *Managing Servers with Netscape Console* provides background information on basic cryptography concepts and the role of Netscape Console.
 - For the HTML version, see `<server_root>/manual/en/admin/help/contents.htm`.
- *Netscape Certificate Management System Installation and Deployment Guide* (this guide) describes how to plan for and install Certificate Management System. To access the installation and configuration information from within the CMS Installation Wizard, click any help button.
 - The HTML version of this guide is located at `<server_root>/manual/en/cert/dep_gide/contents.htm`.

- The PDF version of this guide is located at `<server_root>/manual/en/cert/pdf/cms42install.pdf`.
- *Netscape Certificate Management System Administrator's Guide* provides detailed reference information on CMS administration interfaces. To access this information from the CMS window within Netscape Console, click any help button.
 - The HTML version of this guide is located at `<server_root>/manual/en/cert/adm_gide/contents.htm`.
 - The PDF version of this guide is located at `<server_root>/manual/en/cert/pdf/cms42adm.pdf`.
- *Netscape Certificate Management System Agent's Guide* provides detailed reference information on CMS agent interfaces. To access this information from the Agent Services pages, click any help button.
 - The HTML version of this guide is located in `<server_root>/<instance_id>/web/agent/manual/agt_gide/contents.htm`.
 - The PDF version of this guide is located at `<server_root>/manual/en/cert/pdf/cms42agent.pdf`.
- End-entity help (online only, not printed) provides detailed reference information on CMS end-entity interfaces. To access this information from the end-entity pages, click any help button.
 - The HTML version of this guide is located at `<server_root>/<instance_id>/web/ee/manual/ee_gide/contents.htm`.

Important Do not change the default location of any of the HTML files; they are used for online help. You may move the PDF files to another location.

For a complete list of all documentation that ships with Certificate Management System, including documentation for Directory Server, see Documentation Summary, located at `<server_root>/manual/index.html`.

For the latest information about Certificate Management System, including current release notes, technical notes, and deployment information, see <http://docs.ipplanet.com/docs/manuals/cms.html>.

Where to Go for Related Information

Overview and Demo Installation

A large, white, serif numeral '1' is centered within a solid gray square. The square is positioned in the upper right quadrant of the page.

Chapter 1 Introduction to Certificate Management System

Chapter 2 Default Demo Installation

Introduction to Certificate Management System

This chapter introduces Netscape Certificate Management System (CMS), a highly configurable set of software components and tools for creating, deploying, and managing certificates.

The chapter has the following sections:

- System Overview (page 22)
- Authentication and Policy Modules (page 29)
- Some Enrollment Scenarios (page 33)
- End Entities and Life-Cycle Management (page 48)
- Summary of System Features (page 55)
- System Architecture (page 70)
- Standards Summary (page 74)

This guide assumes that you are familiar with the concepts of public-key cryptography and digital certificates. For a list of key concepts and information on where to learn more about them, see “What You Should Already Know” on page 13.

System Overview

Netscape Certificate Management System provides a highly scalable, easily deployable certificate infrastructure for supporting encryption, authentication, tamper detection, and digital signatures in networked communications. It is based on open standards and protocols that include the following:

- Public-Key Cryptography Standard (PKCS) #11
- Secure Sockets Layer (SSL)
- Lightweight Directory Access Protocol (LDAP)
- X.509 certificate formats recommended by the International Telecommunications Union (ITU)
- Public-Key Infrastructure (X.509) (PKIX) standards proposed by the PKIX working group of the Internet Engineering Task Force (IETF).
- Federal Information Standards Publications (FIPS PUBS) 140-1.

Certificate Management System leverages Netscape Directory Server and Netscape Console to provide a complete, scalable, high-performance certificate management solution for extranets and intranets. Its strong support for existing and evolving standards makes Certificate Management System especially well-suited for large heterogeneous extranets that must support a variety of platforms, client and server software, hardware devices such as routers and hardware tokens, virtual private network (VPN) implementations, existing intranet security systems, and so on. It can be customized and configured to fit widely varying deployment scenarios, permitting rapid integration with existing client and server software, customer databases, security systems, and authentication procedures.

This chapter describes the basic features and capabilities of Certificate Management System. Chapter 2, “Default Demo Installation,” describes how to install a simple demo that uses some of these features.

Public-Key Infrastructure

The standards and services that facilitate the use of public-key cryptography and X.509 version 3 certificates in a networked environment are collectively called *public-key infrastructure (PKI)*. In any PKI, a *certificate authority (CA)* is a trusted entity that issues, renews, and revokes certificates. An *end entity (EE)* is a person, router, server, or other entity that uses a certificate to identify itself.

To participate in a PKI, an end entity must *enroll*, or register, in the system. The end entity typically initiates enrollment by giving the CA some form of identification and a newly generated public key. The CA uses the information provided to *authenticate*, or confirm, the identity. In some cases the CA may require human intervention, such as an interview or examination of notarized documents, to authenticate the end entity (manual approval). In other cases the information provided may be sufficient (automatic approval). In addition to authenticating the end entity, the CA uses the public key to ensure “proof of possession”—that is, cryptographic evidence that the certificate request was signed by the holder of the corresponding private key. Finally, the CA issues a certificate that associates the end entity’s identity with the public key, and signs the certificate with the CA’s own private signing key.

Netscape Certificate Management System dramatically simplifies the PKI enrollment process. Before you deploy a PKI, however, you need to make many decisions about the relationships between CAs and end entities and related policies and procedures.

End entities and CAs may be in different geographic or organizational areas or in completely different organizations that are linked through an extranet (that is, the extension of a company’s internal network, or intranet) to selected customers, suppliers, and mobile employees via the Internet. CAs may include third parties that provide services through the Internet as well as the root CAs and subordinate CAs for individual organizations. Policies and certificate content may vary from one organization to another. For all these reasons and many others, the deployment and long-term management of any large-scale PKI require careful advance planning and custom configuration.

Subsystems of Certificate Management System

To meet the widest possible range of configuration requirements, Certificate Management System permits the independent installation of three separate subsystems, or “managers,” that typically play distinct roles:

- A **Certificate Manager** functions as a root or subordinate certificate authority. This subsystem issues, renews, and revokes certificates, generates certificate revocation lists (CRLs), and can publish certificates and CRLs to an LDAP directory. It can be configured to accept requests from end entities, Registration Managers, or both, and can process requests either manually (that is, with the aid of a human being) or automatically (based entirely on customizable policies and procedures). When set up to work with a separate Registration Manager, the Certificate Manager processes requests and returns the signed certificates to the Registration Manager for distribution to the end entities. (For an overview of the role of certificate authorities and related concepts of public-key cryptography, see Appendix D of *Managing Servers with Netscape Console*.)
- A **Registration Manager** performs a subset of the end-entity tasks performed by the Certificate Manager, such as enrollment or renewal, on behalf of the Certificate Manager. A Registration Manager is typically installed on a different machine from the Certificate Manager that it serves. After the Registration Manager approves requests, it forwards them to this Certificate Manager, which trusts the Registration Manager to provide reliable authentication services and therefore trusts any signed requests it submits. The Certificate Manager processes the requests and issues the certificates. The Registration Manager then distributes the certificates to the end entities.
- A **Data Recovery Manager** performs the long-term archival and recovery of private encryption keys for end entities. A Certificate Manager or Registration Manager can be configured to archive end entities' private encryption keys with a Data Recovery Manager as part of the process of issuing new certificates. The Data Recovery Manager is useful only if end entities are encrypting data (using applications such as S/MIME email) that the organization may need to recover someday. It can be used only with client software that supports dual key pairs—that is, two separate key pairs, one for encryption and one for digital signatures. This service is available in newer clients only (including future versions of Communicator). The Data

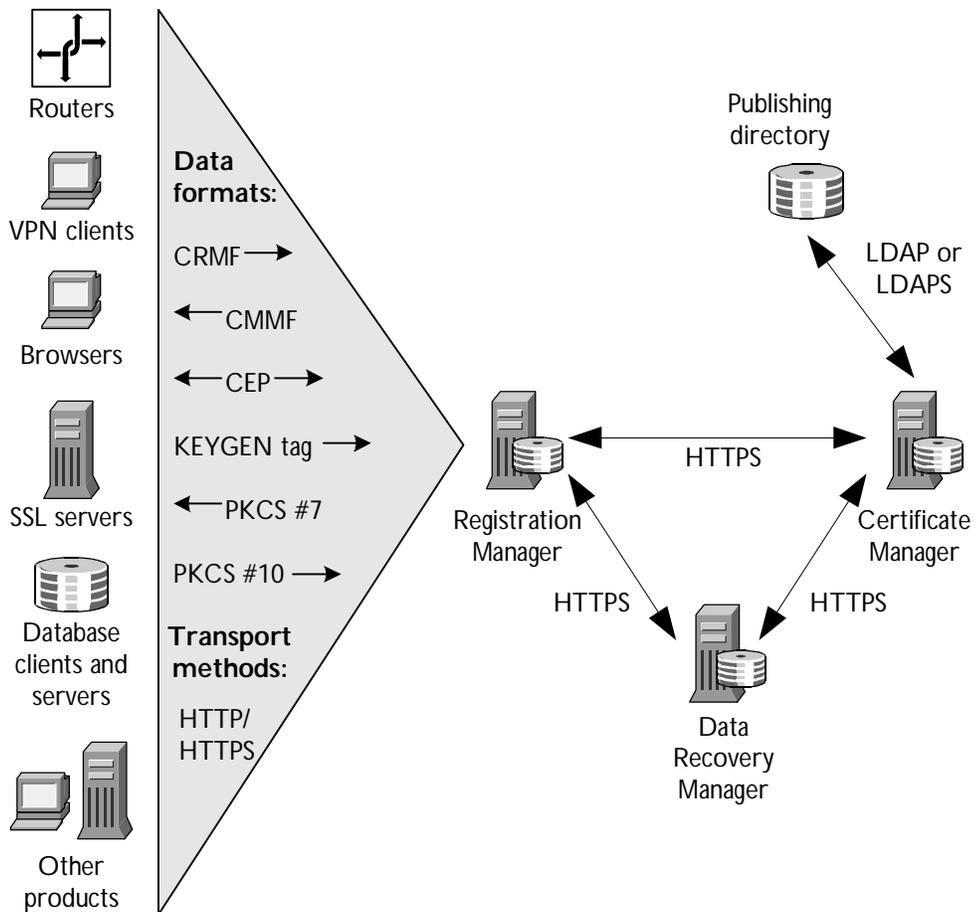
Recovery Manager archives encryption keys. It does not archive signing keys, since such archival would undermine nonrepudiation properties of dual-key certificates.

The Certificate Manager, Registration Manager, and Data Recovery Manager subsystems are all highly customizable and can be installed in a variety of configurations and physical locations. Decisions about the number of subsystems to install, where to install them, and the relationships among them and one or more public directories affect all aspects of installation and configuration. Some organizations may want to install a single Certificate Manager on one machine inside the firewall and a single Registration Manager on a separate machine outside the firewall. Others may have a single CA run by a single Certificate Manager and hundreds of Registration Managers in different geographic locations. Still others may have many different CAs or subordinate CAs, and only a few Registration Managers. For descriptions of some basic deployment options, see Chapter 3, “Planning Your Deployment.”

Basic System Configuration

Figure 1.1 illustrates some of the data formats and protocols used among the three independent CMS managers and various kinds of end entities. To keep things simple, the figure assumes that each manager is installed in a different CMS instance and on a different machine. The Registration Manager handles all interactions with different kinds of end entities, using protocols appropriate for each entity.

Figure 1.1 Basic CMS configuration and use of data formats and protocols



The end-entity data formats and transport methods shown in the figure are used to send enrollment and other requests to the Registration Manager (indicated by a right-pointing arrow) or to send responses back to the end entities (indicated by a left-pointing arrow). The end-entity data formats can be summarized as follows:

- **Certificate Request Message Format (CRMF) and Certificate Management Message Formats (CMMF).** Proposed standards from the Internet Engineering Task Force (IETF) PKIX working group that define message formats used to convey requests to a Registration Manager or Certificate Manager and to return information to end entities. CMMF will be subsumed by another proposed standard, Certificate Management Messages over Cryptographic Message Syntax (CMC), which is also supported by Certificate Management System.
- **Certificate Enrollment Protocol (CEP).** A certificate management protocol jointly developed by Cisco Systems and VeriSign, Inc. CEP governs communication between routers or VPN clients and a Registration Manager or Certificate Manager.
- **KEYGEN tag.** An HTML tag supported by Netscape browsers that generates a key pair stored in the client and formats an HTTP GET string to send off to a CA as part of the enrollment process.
- **Public-Key Cryptography Standard (PKCS) #7.** An encrypted data and message format developed by RSA Data Security to represent digital signatures, certificate chains, and encrypted data. This format is used to deliver certificates to end entities.
- **Public-Key Cryptography Standard (PKCS) #10.** A message format developed by RSA Data Security for certificate requests. This format is supported by many server products and by Microsoft Internet Explorer.

These are the standard transport methods used for all of the data formats described above:

- **Hypertext Transport Protocol (HTTP) and Hypertext Transport Protocol Secure (HTTPS).** Protocols used to communicate with web servers.

For more information about end-entity data formats and protocols used by Certificate Management System, see “End Entities and Life-Cycle Management” on page 48 and “Standards Summary” on page 74.

The Registration Manager communicates with the Data Recovery Manager and the Certificate Manager as necessary to facilitate certificate management operations such as enrollment, renewal, or key storage. When the three subsystems are installed in separate CMS instances (whether on the same machine or on different machines), they communicate with each other over HTTPS—that is, HTTP over SSL, as shown in Figure 1.1.

The Certificate Manager can publish certificates and CRLs to a public directory using LDAP or LDAP over SSL (LDAPS). It's also possible for the Registration Manager to publish certificates. However, the Certificate Manager has the complete record of issued certificates, so it is recommended that publishing tasks be performed by the Certificate Manager only. If it's necessary for some entries in a directory to be available outside the firewall, Netscape recommends using the partial replication feature of Directory Server to replicate the relevant portion of the directory to which the Certificate Manager publishes. In this guide, a directory used for publishing certificates and CRLs is called a *publishing directory*. Publishing directories can also be used for authentication.

Each CMS manager has its own internal LDAP directory for storing private information such as certificate records, key archival records, and the request queue. For example, the Certificate Manager uses its directory for storing certificates and certificate requests; the Registration Manager uses its directory for storing certificate requests (but not certificates, which are stored by the Certificate Manager only); and the Data Recovery Manager uses its directory for storing archived encryption keys. These internal directories are configured during installation. They allow Certificate Management System to leverage the scalability and industry-leading performance of Netscape Directory Server, which replaces the Relational Database Management System (RDBMS) used in Certificate Server 1.x.

Some deployments require installation of two subsystems in a single CMS instance on a single machine: either Certificate Manager and Data Recovery Manager or Registration Manager and Data Recovery Manager. For these dual-manager installations, communication between the two subsystems takes place internally (that is, within the same running process) rather than via HTTPS. (Note that a Certificate Manager performs all Registration Manager tasks, including end-entity interactions. Registration Managers are required only for remote or delegated administration of the CA.)

Throughout this guide, the term *CMS administrator* describes the person who installs and configures one or more managers and sets up privileges for the users who manage those subsystems. The users who manage day-to-day interactions of end entities with each manager, as well as other aspects of the

PKI, are called *CMS agents* collectively, or the *Certificate Manager agent*, *Registration Manager agent*, and *Data Recovery Manager agent*. The role of an agent is to approve, defer, or reject requests using Agent Services web pages served by the CMS manager for which that agent has been assigned the necessary privileges. The privileges of each agent can be confined to a specific manager or can include several different managers.

System administrators set up CMS subsystems through Netscape Console, and agents manage end-entity requests and certificates through HTML pages. For more information about facilities available to administrators and agents, see Chapter 2, “Default Demo Installation.”

Authentication and Policy Modules

Certificate Management System includes a plug-in architecture for code modules that authenticate user identities and code modules that enforce policies.

Each type of request from an end user—for certificate enrollment, renewal, revocation, or retrieval—is handled by a different *servlet*, a piece of Java code designed for that kind of request. Each servlet processes the request using the appropriate protocols (such as the `KEYGEN` HTML tag or PKCS #10) for each type of end entity. Additional servlets control interactions with administrators and agents.

Authentication Modules

An *authentication module* is a set of rules (implemented as a Java class) for authenticating an end user, server, or other entity that needs to interact with a CMS manager. (Similar rules are used to authenticate agents and administrators, but they are built into Certificate Management System instead of being implemented as plug-in modules.) With a typical end-user enrollment, the user supplies the information requested by the Registration Manager on an enrollment form, and then the servlet uses an authentication module specified within the form to validate the information and authenticate the user's identity. This simple input value makes it possible to use custom authentication for any form without changing the corresponding servlet code.

CMS managers always support client SSL certificate-based authentication (for both agents and end entities) and user-ID-based and password-based authentication for administrators. Registration Managers and Certificate Managers can also be configured to use standard CMS authentication modules that perform directory-based authentication and directory-based personal identification number (PIN) authentication for end entities. Software development kits (SDKs) are available that demonstrate how to write custom authentication modules, for example to authenticate end entities using existing customer databases or security systems.

Policy Modules

After a Registration Manager or Certificate Manager has successfully authenticated an end entity, the entity's request is passed to a policy processor, which sequentially applies a set of policy rules configured for that CMS manager. A *policy module* is a rule (implemented as a Java class) that validates the contents of a certificate request for that rule and can add or modify any part of a certificate's contents, including validity dates, name constraints, and extensions.

Here are three typical examples of the use of policies:

- A name constraints policy checks that the subject name matches a pattern, and it rejects, defers, or adjusts the subject name in the request accordingly.
- A validity constraints policy checks that the certificate validity period falls within a specified period, and it rejects, defers, or adjusts the validity period in the request accordingly.
- An extensions policy checks that a request includes a specified extension and adds the extension if it's missing.

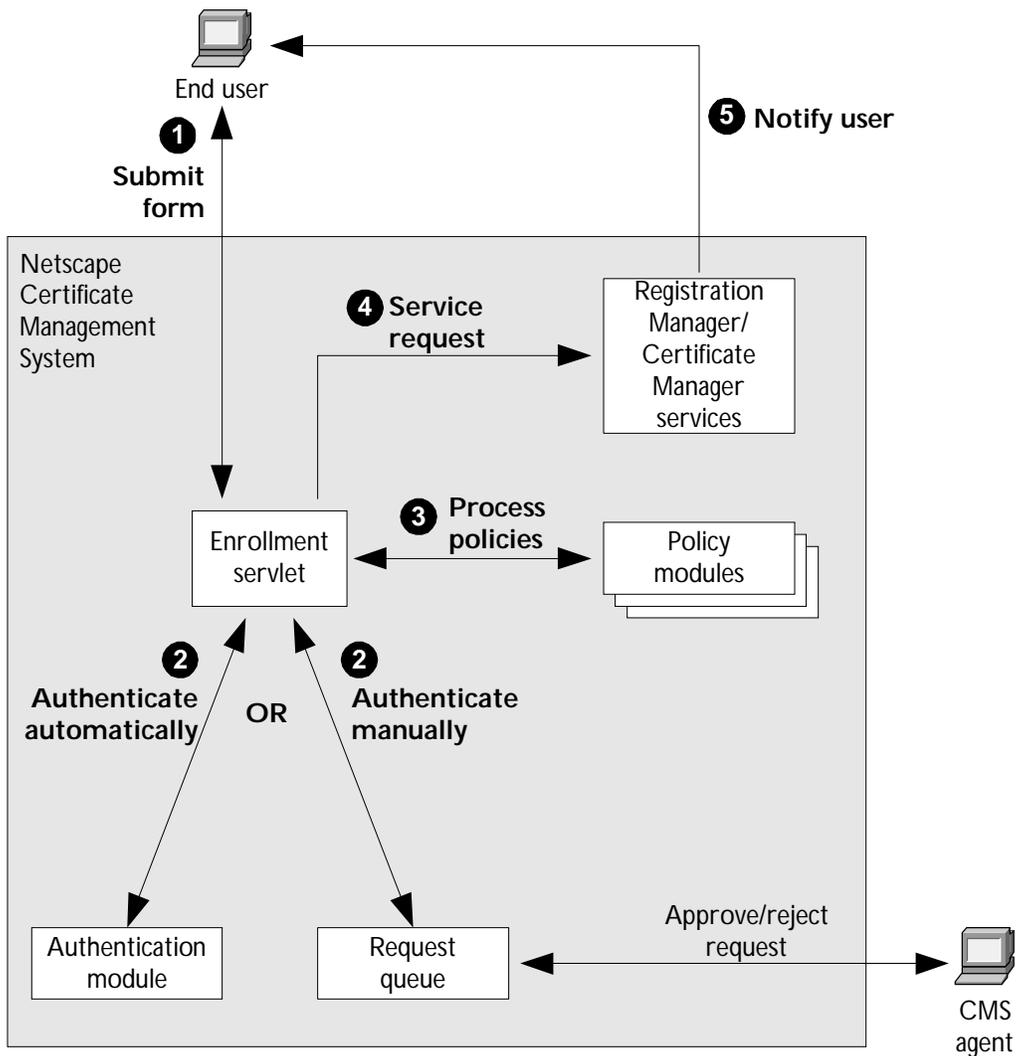
Steps in End-Entity Enrollment

The following steps take place when a Registration Manager or a Certificate Manager handles an enrollment request from an end user. Figure 1.2 shows a simplified view of how this works.

1. **Submit form.** When the user first interacts with the CMS manager (either the Registration Manager or the Certificate Manager), the user specifies the kind of request to be made, fills in the form for that request, and submits it to the servlet via HTTP or HTTPS. The servlet then processes the form. In the figure, a certificate request is being sent to an enrollment servlet. It could also be a renewal or revocation request being sent to one of the other servlets.
2. **Authenticate user.** Authentication can be either automatic or manual. If the CMS manager is configured for automatic authentication, the servlet uses the authentication module specified by the form to validate the information provided by the user. For example, the directory authentication module that comes with Certificate Management System validates the user ID and password by comparing it to the user's entry in an LDAP directory. Custom authentication modules can be used to take advantage of existing databases, security systems, or other methods of authentication. If the CMS manager is configured for manual authentication, the servlet routes the request to the request queue and informs the user (via a web page) that approval has been deferred. The request remains in the queue until an agent approves it or rejects it.
3. **Process policies.** If authentication is successful, policies specified for this CMS manager are applied to the request for the purpose of formulating the contents of the certificate to be issued and to enforce certain rules, such as name constraints. Custom policy modules can be used to enforce specialized certificate extensions and other requirements.
4. **Service request.** After policy processing, the servlet's work is finished and the CMS manager services the request (assuming that a policy has not triggered deferral)—for example, by issuing a certificate.
5. **Notify user.** If the CMS manager has been configured for automatic authentication and issuance, the manager delivers the signed certificate to the user via a web page. If the request has been deferred (for example, for manual approval) or rejected, the user is informed of the request's status. When the request has been approved and the certificate issued, the CMS manager notifies the user (for example, with an email) and provides a URL where the certificate can be picked up.

Since all three CMS managers use the same architecture for authentication and policy processing, it's possible to reuse any authentication and policy modules with any manager. For information on the relationship of policy modules to the APIs exposed by Certificate Management System, see "System Architecture" on page 70.

Figure 1.2 Roles of servlets, authentication modules, and policy modules in end-entity enrollment



Some Enrollment Scenarios

Successful PKI deployment requires flexible and easy enrollment for end entities as well as ongoing support for *certificate life-cycle management*—that is, management of each certificate from enrollment through encryption key storage (if necessary), renewal, and revocation. The preceding section describes the internal flow of control among servlets, authentication modules, and policy modules in a CMS manager (see Figure 1.2 for a summary). The examples that follow illustrate the flexibility that the CMS architecture supports among end entities, Registration Managers, Certificate Managers, and existing customer databases, security systems, and directories.

- Firewall Considerations (page 34)
- Extranet/E-Commerce: Acme Sales Corp. (page 35)
- PIN Registration: Atlas Manufacturing (page 42)
- VPN Client Enrollment and Revocation (page 44)
- Router Enrollment and Revocation (page 46)

For the sake of simplicity, these examples do not show the role of the Data Recovery Manager. For more information about data recovery, see “Data Recovery Manager” on page 64.

For more information about certificate life-cycle management, see “End Entities and Life-Cycle Management” on page 48.

Firewall Considerations

Most of the examples that follow show a Certificate Manager inside the firewall and a Registration Manager outside the firewall. Other variations are possible, but this arrangement is often appropriate. These are some of the advantages:

- The most sensitive elements of the deployment—the Certificate Manager, internal databases, directories, and so on—have the additional protection of the firewall.
- The Certificate Manager can have additional physical protection, if desired—such as storage in a locked room and agent authentication by means of smart cards.
- All communication between the Registration Manager and the Certificate Manager takes place over SSL with mutual authentication—that is, both client and server authentication via X.509 v3 certificates.
- The Registration Manager provides only a subset of the capabilities of the Certificate Manager—those required for processing end-user requests. If the Registration Manager is compromised, the Certificate Manager can revoke its signing certificate (thus invalidating all subsequent requests from that Registration Manager) and issue a new one after the problem has been addressed.

Administrative and physical arrangements are closely related to firewall issues. The flexibility of CMS deployment options makes it possible to divide functions among existing administrative groups or physical locations, requiring minimal disruption for an organization.

The examples that follow do not address the role of the Data Recovery Manager or the potential use of multiple Registration Managers and Certificate Managers. For example, in some circumstances it might make sense to have some Registration Managers outside the firewall and some inside; in other cases different CMS subsystems might be located in entirely different physical locations, each with their own firewalls.

In general, Netscape recommends that the Certificate Manager handle all certificate and CRL publishing functions. If it's necessary for some entries in a directory to be available outside the firewall, Netscape recommends using the partial replication feature of Directory Server to replicate the relevant portion of the directory.

Extranet/E-Commerce: Acme Sales Corp.

Acme Sales is a high-end mail-order catalog service that is launching an online shopping service. Many of Acme's affluent customers make very expensive purchases, so Acme has decided to use certificate-based authentication for its new web site.

Acme has 100,000 existing customers and expects to attract many new customers through its online service. The company wants to use its existing relational database to authenticate and enroll existing customers with minimal effort on their part. For new customers, Acme wants to establish a manual process entailing out-of-band credit checks (that is, checks that don't involve an electronic network), identity verification, and a personal phone call before an online certificate request can be granted. In addition, Acme plans to issue certificates to contract workers, suppliers, and employees who routinely access parts of the company's internal network by using Kerberos.

The sections that follow describe how Acme uses Certificate Management System to achieve these goals:

- Enrolling Existing Customers (page 36)
- Enrolling New Customers (page 38)
- Enrolling Extranet Users (page 40)

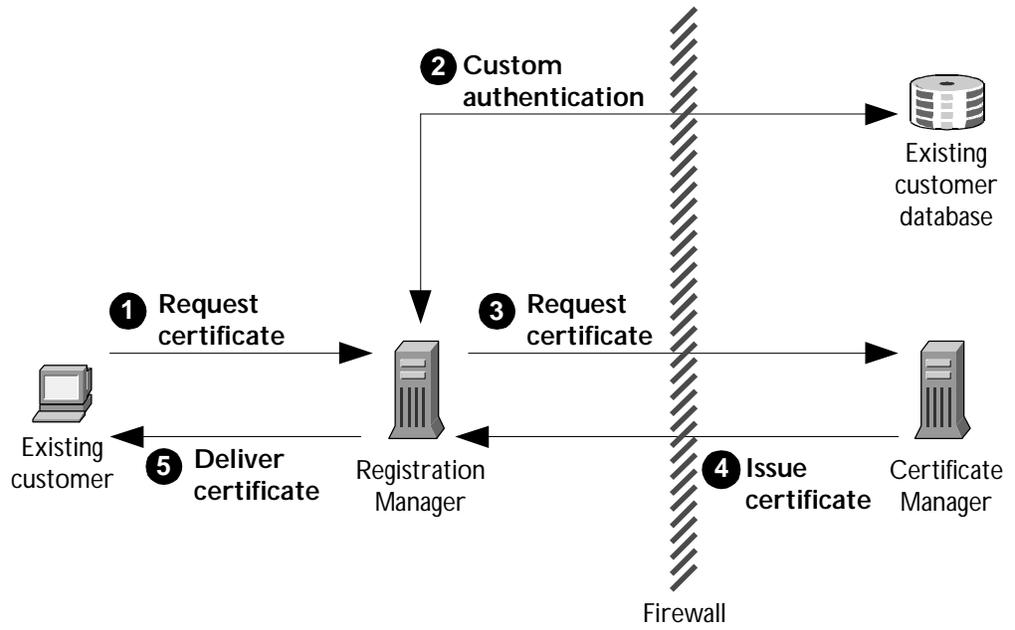
In all cases, Acme has decided to place its Certificate Manager behind the firewall and its Registration Manager outside the firewall, for reasons summarized in "Firewall Considerations" on page 34.

Enrolling Existing Customers

Acme has decided on the following process for registering its existing customers, as shown in Figure 1.3.

1. **Request certificate.** The customer fills in and submits a form (over SSL) that specifies account information and other personal details stored in the existing customer database.
2. **Custom authentication.** The Registration Manager uses a custom authentication module to verify the customer's account and status against the existing customer database.
3. **Request certificate.** If authentication against the customer database is successful, the Registration Manager performs policy processing and, if processing is successful, forwards the request to the Certificate Manager.
4. **Issue certificate.** The Certificate Manager performs its own policy processing and, if processing is successful, issues the certificate and delivers it to the Registration Manager.
5. **Deliver certificate.** If the Certificate Manager successfully issues the certificate, the Registration Manager delivers it to the end user in the same session. If the request is unsuccessful for any reason, the Registration Manager displays a web page to the customer explaining the problem and what to do about it.

Figure 1.3 Custom authentication against an existing customer database

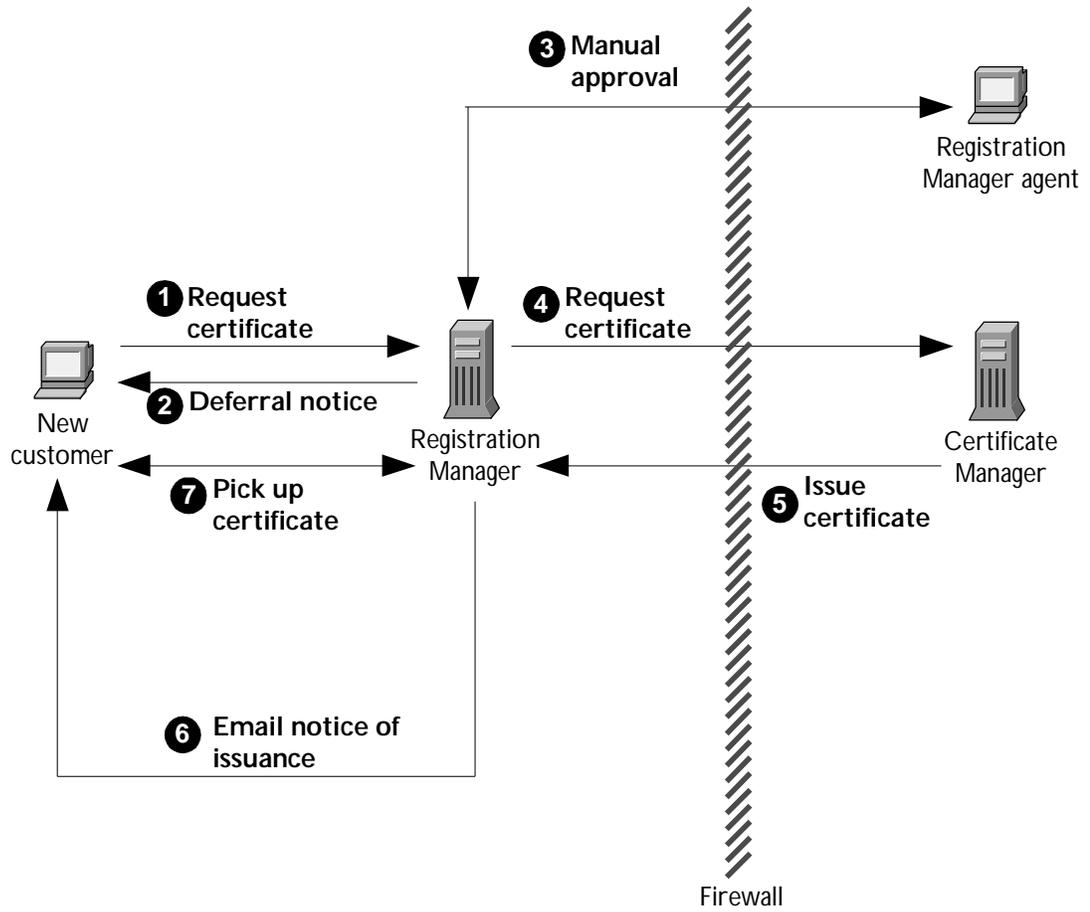


Enrolling New Customers

The following process will be used for enrolling new Acme customers. In this case, the Registration Manager uses manual authentication to validate every certificate request personally before issuing the certificate. Figure 1.4 illustrates the steps in this process.

1. **Request certificate.** The customer fills in and submits a certificate request form for new Acme customers.
2. **Deferral notice.** The Registration Manager immediately informs the customer (via a web page) that the request has been deferred and that Acme will be in touch soon. Meanwhile, the certificate request waits in a queue for attention from the Registration Manager agent.
3. **Manual approval.** The Registration Manager administrator may configure the Registration Manager to notify the agent via email whenever a new request is added to the request queue. In any case, when the agent processes the requests in the queue, he or she follows Acme's procedure for processing credit checks and validating other customer information, including making a personal phone call. If all authentication procedures are successful, the agent approves the request.
4. **Request certificate.** The Registration Manager performs policy processing and, if the processing is successful, sends the approved request to the Certificate Manager.
5. **Issue certificate.** The Certificate Manager performs its own policy processing on the request and, if processing is successful, issues the certificate and delivers it to the Registration Manager.
6. **Email notice of issuance.** The Registration Manager sends an email containing a URL to the new customer, asking the customer to pick up the certificate.
7. **Pick up certificate.** The customer goes to the specified Registration Manager URL and picks up the certificate.

Figure 1.4 Manual authentication of new customers



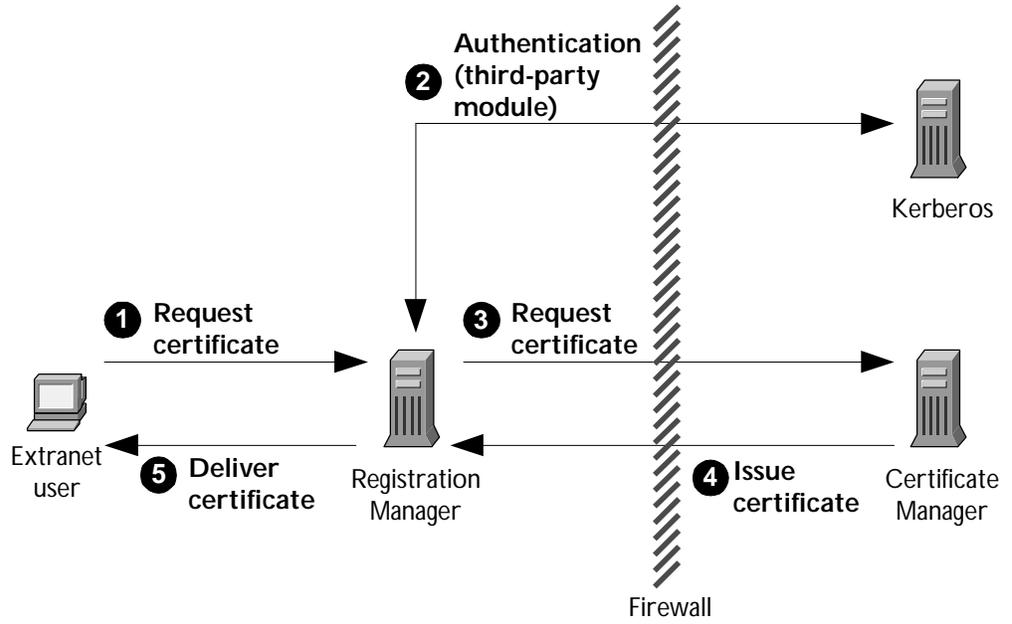
Enrolling Extranet Users

Acme wants its new, certificate-enabled extranet applications to be available to contract workers, suppliers, employees, and others who routinely access parts of the company's internal network. In general, this can be achieved by using Kerberos or other non-PKI security systems as the authentication mechanism for requesting a certificate. To authenticate them for the purposes of PKI enrollment, Acme uses a third-party authentication module from DASCOS that takes advantage of its existing Kerberos system without disturbing its current functions.

For example, to get a certificate, a contractor provides an ID and password to the Registration Manager, which uses the Kerberos system to verify them before passing on the certificate request to the Certificate Manager. This arrangement involves the following steps, illustrated in Figure 1.5. (The details of the existing security system don't matter: third-party or custom CMS authentication modules can be used for Kerberos, NIS, and many other security systems. Extranet users can continue to use applications based on the old security systems while they use their certificates to take advantage of new certificate-based applications.)

1. **Request certificate.** A user of Acme's existing extranet fills in and submits a certificate request (over SSL) using a customized form that requires a Kerberos ID and password.
2. **Authentication.** The Registration Manager uses a third-party authentication module to validate the user's identity using the existing internal Kerberos system.
3. **Request certificate.** If authentication against Kerberos is successful, the Registration Manager performs policy processing and, if processing is successful, forwards the request to the Certificate Manager.
4. **Issue certificate.** The Certificate Manager performs its own policy processing on the request and, if processing is successful, issues the certificate and delivers it to the Registration Manager.
5. **Deliver certificate.** If the Certificate Manager issues the certificate, the Registration Manager delivers it to the end user in the same session. If the request is unsuccessful for any reason, the Registration Manager displays a web page to the user explaining the problem and what to do about it.

Figure 1.5 Custom authentication against an existing Kerberos security system



PIN Registration: Atlas Manufacturing

Atlas Manufacturing has decided to put information for its employees, suppliers, dealers, and customers—a total of nearly 500,000 people, including individual consumers and employees of several dozen other companies—on an extranet. Atlas already uses Netscape Directory Server to store names, addresses, and other information about the various groups of people who will need access to the extranet. To register all these people at once, Atlas uses the directory-based PIN Generator tool that comes with Certificate Management System to generate PINs in bulk. The PINs are then stored in the directory and delivered to the end users via a batch mailer program, an employee payroll stub, a customer invoice, or some other means of physical delivery.

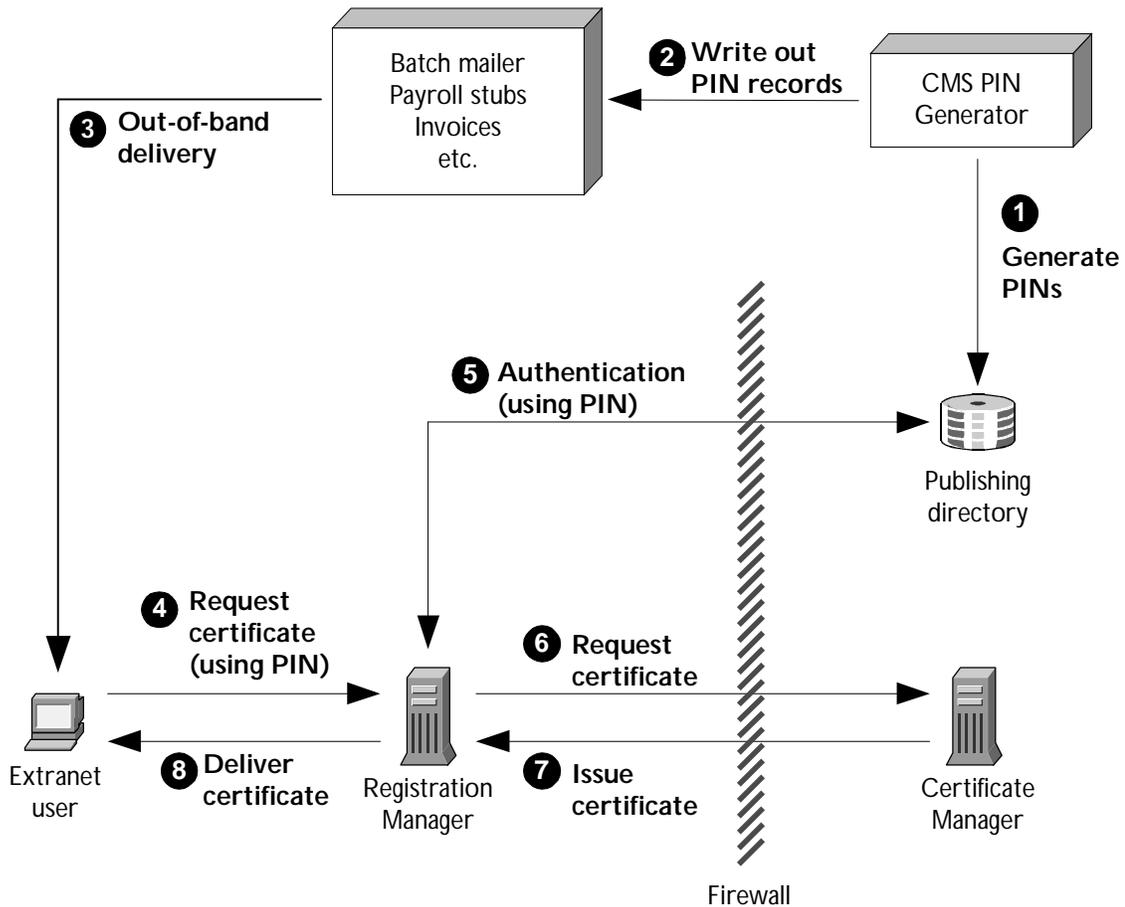
PINs are salted and hashed before storage in the directory. *Salting* refers to the inclusion of additional information from the distinguished name (DN) with the PIN to ensure unique hashing. *Hashing*, in this case, involves generating a number of fixed length from the PIN and DN information. Even if the security of the directory is breached, it is very difficult to reconstruct the PIN from the value that results from salting and hashing. When customers use the PIN to enroll in the Atlas PKI, the PIN is automatically removed from the directory. Enrollment PINs are therefore more reliable than passwords, which must be protected over a long period of time.

Acme's process involves the following steps (illustrated in Figure 1.6):

1. **Generate PINs.** The CMS administrator runs the CMS PIN Generator against the existing directory, populating each entry with a unique PIN.
2. **Write out PIN records.** The CMS administrator uses the CMS PIN Generator to write out PIN records for use by an out-of-band delivery mechanism.
3. **Out-of-band delivery.** The user receives the PIN via a batch mailing system, payroll stub, invoice form, or other out-of-band delivery mechanism.
4. **Request certificate (using PIN).** The user goes to a specified Registration Manager URL, fills in name and PIN, and submits a certificate request.
5. **Authentication (using PIN).** The Registration Manager uses the standard CMS PIN-based directory authentication module to verify the PIN against the directory.

6. **Request certificate.** If authentication against the directory is successful, the Registration Manager performs policy processing and, if this succeeds, forwards the request to the Certificate Manager.
7. **Issue certificate.** The Certificate Manager performs its own policy processing and, if all goes well, issues the certificate.
8. **Deliver certificate.** If the Certificate Manager issues the certificate, the Registration Manager delivers it to the end user in the same session. If the request is unsuccessful for any reason, the Registration Manager displays a web page to the user explaining the problem and what to do about it.

Figure 1.6 PIN-based enrollment



VPN Client Enrollment and Revocation

Virtual private network (VPN) client software runs on a user's desktop, outside the firewall, and uses the IP Key Management Protocol (IPKMP) or IP Security (IPSec) protocol to establish encrypted communication with VPN hardware that straddles the firewall. These protocols allow VPN hardware to authenticate VPN client software using the client's certificate, in much the same way that the SSL protocol allows a server to authenticate client browser software.

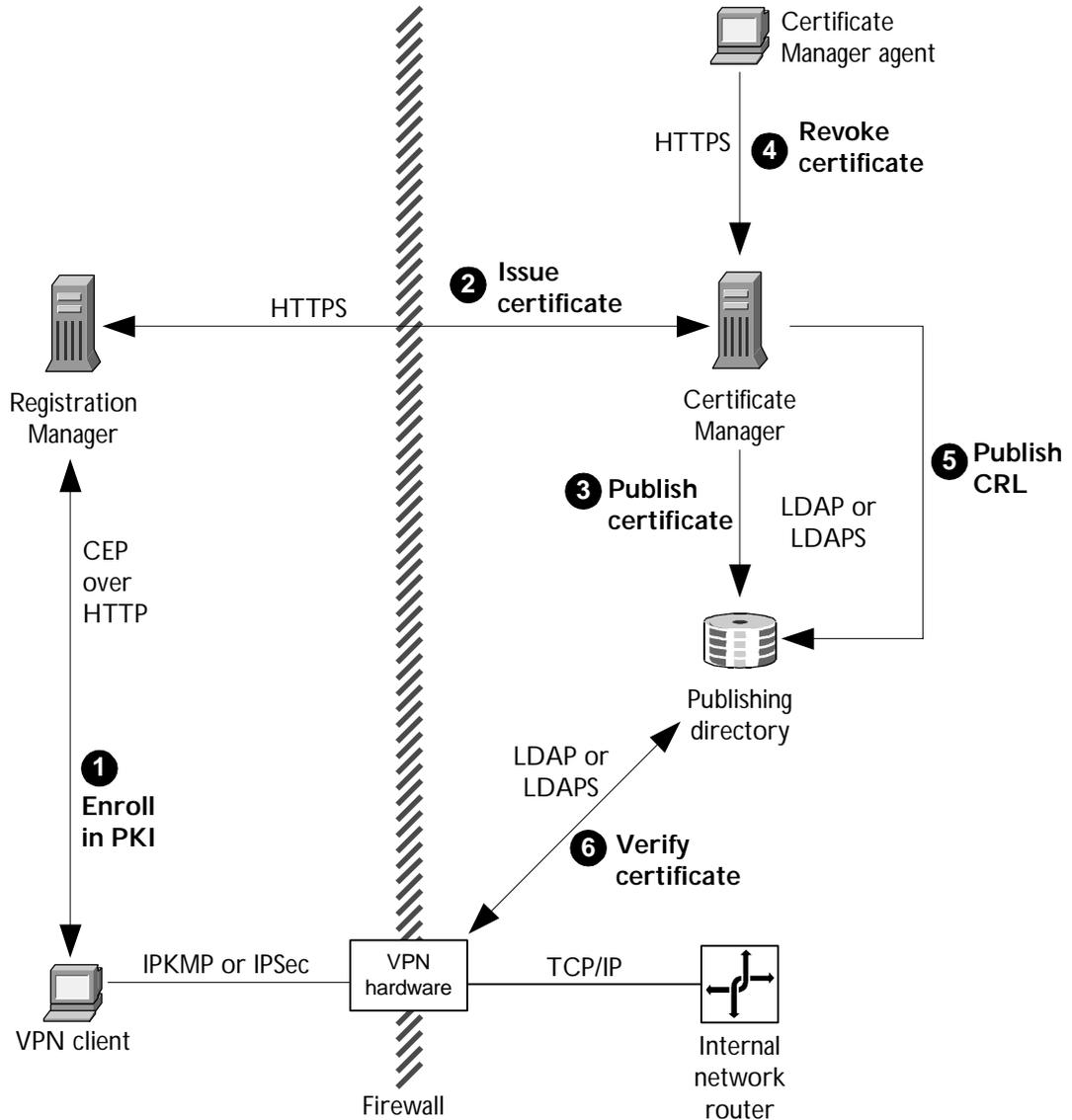
VPN client software can use several different protocols over HTTP or HTTPS to handle enrollment and other life-cycle management tasks. Certificate Management System supports the Certificate Enrollment Protocol (CEP) used by Cisco routers. CEP runs over HTTP and provides its own form of encryption.

The following steps explain how VPN client software can use the Registration Manager and Certificate Manager to enroll in a PKI and what happens when the client's certificate is revoked. These steps are shown in Figure 1.7.

1. **Enroll in PKI.** The VPN client sends a certificate request to the Registration Manager via CEP, and the Registration Manager processes the request and forwards it to the Certificate Manager inside the firewall. (Any of the authentication methods discussed in the previous sections can be used during enrollment to authenticate the client.)
2. **Issue certificate.** The Certificate Manager issues the certificate, and the Registration Manager delivers it to the VPN client. The VPN client can now authenticate itself to the VPN hardware and establish an encrypted channel using IPKMP or IPSec. All TCP/IP communication passes through this encrypted channel. From the point of view of the VPN client, it appears to be directly connected to the TCP/IP network inside the firewall.
3. **Publish certificate.** The Certificate Manager publishes the certificate to a directory (this is an optional step).
4. **Revoke certificate.** After some time has passed, the Certificate Manager agent revokes the certificate (for example, after the certificate owner leaves the company).
5. **Publish CRL.** The Certificate Manager publishes a new CRL to the directory specified as the CRL distribution point in the original certificate.

- Verify certificate.** The VPN hardware checks the CRL as part of its authentication process. Certificates listed in the CRL are not authenticated, and VPN clients presenting them cannot establish a connection.

Figure 1.7 VPN client enrollment and revocation



The certificate includes information about a CRL distribution point, which is a directory that the VPN hardware can check for the latest CRL published by the Certificate Manager.

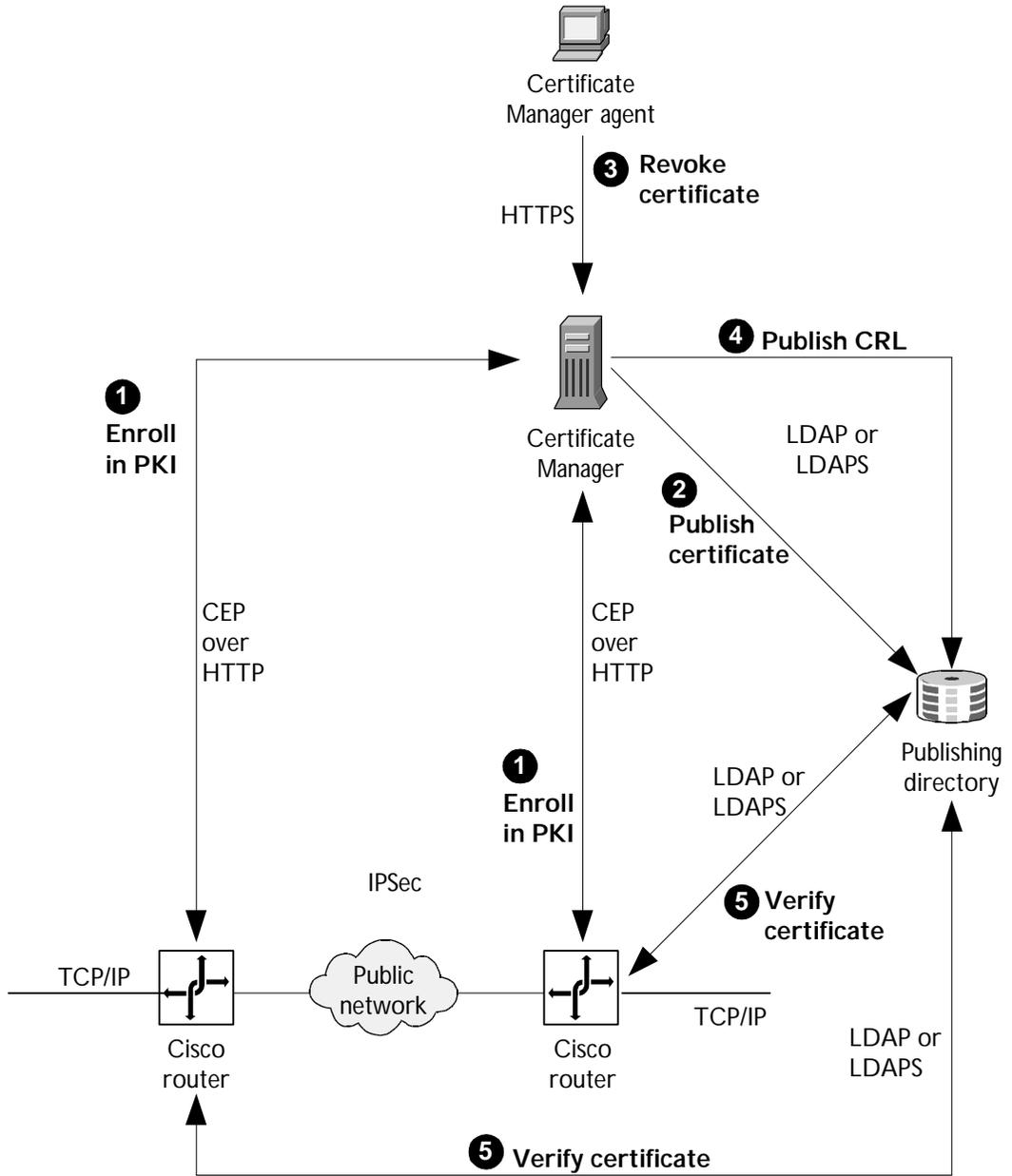
Router Enrollment and Revocation

Cisco routers support the use of certificates for authentication, encryption, and tamper detection with the IP Security (IPSec) protocol. Cisco routers also support CEP for certificate life-cycle management, as discussed in the previous section.

The following steps describe how two routers can use a Certificate Manager to enroll in a PKI and what happens when a router's certificate is revoked. These steps are shown in Figure 1.8.

1. **Enroll in PKI.** The routers each send a certificate request to the Certificate Manager via CEP, and the Certificate Manager issues them certificates. (Any of the authentication methods discussed in the previous section can be used during enrollment to authenticate the client.)
2. **Publish certificates.** As part of the issuing process, the Certificate Manager publishes the certificates to the directory. (Publishing occurs only if the router's DN exists in the publishing directory. This is important for some Cisco routers that must fetch their certificates from an LDAP directory because flash memory is not large enough to hold them.) The routers can now authenticate each other and establish an encrypted channel using IPSec. All TCP/IP communication passes through this encrypted channel. From the point of view of other connections to each router, they all appear to be sharing the same TCP/IP network.
3. **Revoke a certificate.** After some time has passed, the Certificate Manager agent revokes one of the certificates (for example, after the certificate owner leaves the company).
4. **Publish CRL.** The Certificate Manager publishes the CRL to the directory.
5. **Verify certificate.** The routers check the CRL as part of their mutual authentication process. Certificates listed in the CRL are not authenticated, and routers presenting them cannot establish a connection.

Figure 1.8 Router enrollment and revocation



End Entities and Life-Cycle Management

Certificate Management System provides default web forms for all end-entity interactions involved in managing the life cycle of a certificate. It also provides forms, collectively called *Agent Services*, for agent interactions. These forms can be used as is or customized. The Netscape Personal Security Manager is a software that improves the PKI abilities of Netscape Communicator 4.7 and later versions.

The sections that follow introduce the end-entity forms and protocols.

- Life-Cycle Management Formats and Protocols (page 48)
- Access to Subsystems (page 50)
- HTML Forms for End Users (page 52)
- Netscape Personal Security Manager (page 53)

Life-Cycle Management Formats and Protocols

The Registration Manager and Certificate Manager provide default HTML forms that use different protocols and life-cycle management procedures for different kinds of end entities. For example, end entities running Navigator 3.x and versions of Communicator earlier than 4.5 need to be presented with an enrollment form based on the use of the HTML tag `KEYGEN` to generate keys. End entities running Microsoft Internet Explorer require a form containing VBScript `XENROLL` commands. These various tags, scripts, and protocols result in enrollment messages that are sent back to the Certificate Manger or Registration Manager in a variety of nonstandard and standards-based formats.

Table 1.1 summarizes the message formats, cryptographic algorithms, and key pairs (single or dual) supported by Certificate Management System for the main categories of end-entity software. Note that, for the purposes of enrollment, CMS managers are also end entities. CMS managers installed in different instances need SSL client and SSL server certificates to identify themselves. For more information about the standards listed in Table 1.1, see “Standards Summary” on page 74.

Table 1.1 End entities, message formats, algorithms, and key pairs supported by Certificate Management System

End entity software	Enrollment message format over HTTP or HTTPS	Cryptographic algorithms	No. of key pairs
Navigator 3.x Communicator 4.0 to 4.5	KEYGEN tag	Signing and encryption: RSA Signing only: RSA, DSA	Single key pair
Internet Explorer 3.x and 4.x	PKCS #10	Signing and encryption: RSA Signing only: RSA	Single key pair
Internet Explorer 5.x	PKCS #10	Signing and encryption: RSA Signing only: RSA, DSA	Single or dual key pairs
Upcoming version of Communicator	CRMF and CMMF based on new JavaScript API	Signing and encryption: RSA Signing only: RSA, DSA	Single or dual key pairs
Netscape servers (including CMS managers) and other servers	PKCS #10	Signing and encryption: RSA	Single key pair
Cisco routers (version IOS 12.04) and VPN clients	CEP	Signing and encryption: RSA	Single key pair

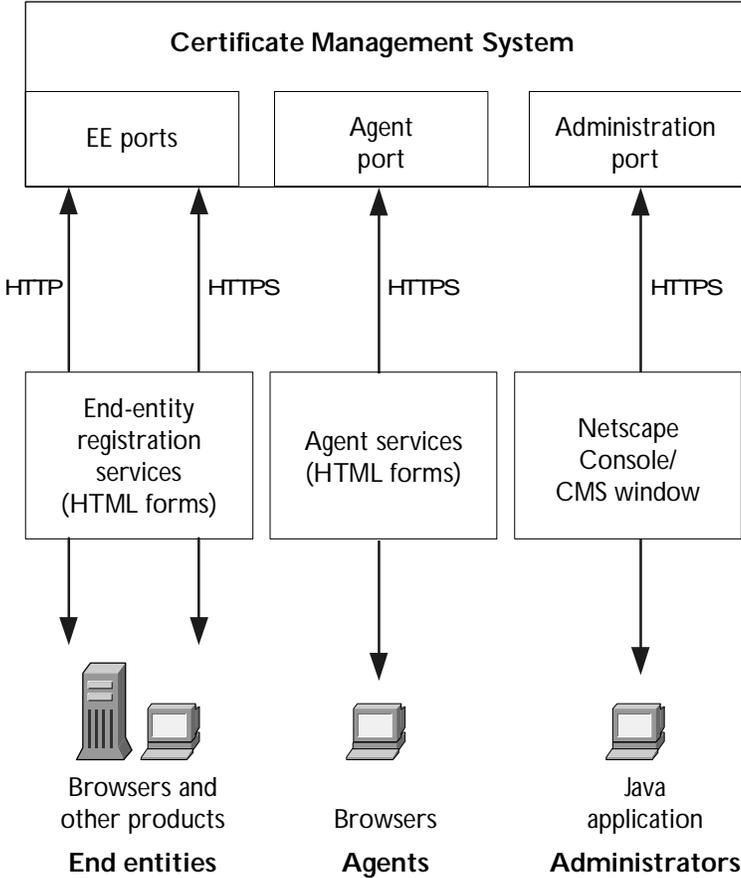
Access to Subsystems

Three kinds of entities can access CMS subsystems: administrators, agents, and end entities. Administrators are responsible for the initial setup and ongoing maintenance of the subsystems. Agents manage the day-to-day operations of each subsystem, such as responding to requests from end entities. End entities access Registration Manager or Certificate Manager subsystems to enroll in a PKI and to take part in other life-cycle management operations, such as renewal or revocation.

Figure 1.9 shows the ports used by administrators, agents, and end entities. All agent and administrator interactions with CMS subsystems occur over HTTPS. For detailed information on the use of the Agent Services interface by agents and the use of Netscape Console by administrators, see *Netscape Certificate Management System Agent's Guide* and *Netscape Certificate Management System Administrator's Guide*, respectively.

End-entity interactions can take place over HTTP or HTTPS. For example, routers using CEP, which includes its own encryption scheme, uses HTTP rather than HTTPS. For a more detailed discussion of these ports and examples of hands-on use, see Chapter 2, "Default Demo Installation."

Figure 1.9 Access ports for Certificate Management System



HTML Forms for End Users

Each type of end-entity form provided by a Registration Manager or Certificate Manager determines the type of client, such as Communicator or Internet Explorer, and presents the appropriate input page. Each form also specifies both an authentication module and an output template. The authentication module is used by the servlet to authenticate the end entity; the *output template* is an HTML page that returns information from the servlet to the end entity.

Figure 1.10 shows the default manual enrollment form as it is presented to end users running Communicator 4.5. Users can click items in the left menu and tabs to access other HTML forms. Server administrators, including CMS administrators, can also access forms for enrolling servers or subsystems. Any of these forms can be customized to reflect an organization's requirements.

Figure 1.10 Default manual enrollment form for end users



Table 1.2 shows the protocols supported by the default CMS life-cycle management servlets. Any of the HTML forms and their HTML help text can be customized. The Registration Manager also supports the creation of new forms. Some output templates can also be customized.

Table 1.2 Default CMS life-cycle management servlets and supported protocols

Life-cycle management servlet	Message syntax/procedures for end entities
Certificate enrollment form	User certificates: KEYGEN for Navigator/Communicator, VBScript/XENROLL and PKCS #10 for Internet Explorer
	Server certificates: PKCS #10 (cut and paste; also URI for Administration Server 3.5 and 4.1)
Certificate renewal form	User certificates: SSL client authentication
	Server certificates: PKCS #10 (cut and paste)
Certificate revocation form	User certificates: SSL client authentication
	Server certificates: agent initiated
Encryption key storage and recovery form	Not supported for Navigator/Communicator 4.x; CRMF for Communicator 5.0 (based on new JavaScript API).

For more information about the standards listed in Table 1.1, see “Standards Summary” on page 74.

Netscape Personal Security Manager

Netscape Personal Security Manager is a standards-based, client-independent application that performs PKI operations on behalf of Netscape Communicator 4.7 and other applications. Personal Security Manager provides advanced cryptographic capabilities while at the same time hiding the complexity of PKI operations from end users. In particular, Personal Security Manager simplifies certificate deployment with Certificate Management System by taking advantage of the following CMS features:

- One-click issuance of certificates.

End Entities and Life-Cycle Management

- Forced certificate backup by end users after certificate issuance.
- Issuance and management of separate signing and encryption certificates.
- Automatic storage of encryption private keys with the Data Recovery Manager at the time a certificate is issued, if requested by the Registration Manager.
- Automatic revocation checking each time Personal Security Manager verifies a certificate.

Behind the scenes, PSM supports the following cryptographic capabilities:

- SSL v2 and v3. SSL authentication, encryption, and tamper detection.
- S/MIME. Signed and encrypted email (using separate signing and encryption keys if desired)
- PKCS #5. Encryption for private key storage.
- PKCS #7. Signing operations.
- PKCS #11. Communication with PKCS #11 modules and associated cryptographic tokens (such as smart cards).
- PKCS #12. Export and import of certificates and associated private keys.
- CRMF/CMMF. Direct communication between Personal Security Manager and a CA, simplifying enrollment processes and making one-click issuance possible.
- Online Certificate Status Protocol (OCSP). Real-time revocation checking.

For more information or to download the latest version of Personal Security Manager, see <http://developer.ipplanet.com/docs/manuals/psm.html>. That page will also provide links to deployment information as it becomes available, including details on how to set up Certificate Management System for use with Personal Security Manager.

Summary of System Features

This section summarizes the most important out-of-the-box features that Netscape Certificate Management System offers in the following areas:

- Authentication Modules (page 55)
- Policy Modules (page 56)
- Job Scheduler Plug-Ins (page 61)
- Event-Driven Notifications (page 61)
- Registration Manager (page 62)
- Certificate Manager (page 62)
- Data Recovery Manager (page 64)
- Command-Line Utilities (page 68)

For detailed descriptions of the agent interfaces used to control these features in CMS subsystems, see *Netscape Certificate Management System Agent's Guide*.

Authentication Modules

An authentication module is a set of rules (implemented as a Java class) for authenticating an end entity, agent, administrator, or any other entity that needs to interact with a CMS manager. For an introduction to the role of authentication modules in the enrollment process, see “Authentication and Policy Modules” on page 29.

All CMS managers support client SSL certificate-based authentication (for both agents and end entities). Netscape Console supports user ID- and password-based authentication for administrators. Registration Managers and Certificate Managers also support three authentication modules out of the box:

- **Manual authentication.** Requires manual approval by an agent. This authentication module is hardwired; you cannot configure it. This ensures that when the server receives requests that lack authentication credentials, it sends them to the request queue for agent approval. It also means that if you don't configure Certificate Management System for any other authentication mechanism, the server automatically sends all certificate-related requests to a queue where they await agent approval.

- **Directory-based authentication.** Checks a user's name and password against the user's entry in a specified directory and uses the DN for that entry to formulate the subject name for the certificate.
- **Directory-based PIN authentication.** Checks a user's name, password, and a special one-time PIN against the user's entry in a specified directory and uses the DN for that entry to formulate the subject name for the certificate. The PIN is stored in salted and hashed form, and is removed after being used once to authenticate a user during enrollment.
- **NIS-based authentication.** Authenticates end users based on their user IDs and passwords stored in a NIS server. Optionally, uses an LDAP directory for formulating certificate subject names.
- **Portal-style authentication.** Checks that a user's name is unique in an LDAP directory.

When you configure authentication modules, you can specify how the DN should be used to formulate the subject name. As a result, neither the user nor the agent needs to figure out or enter the subject name—its formulation is entirely automated.

It's also possible to write custom authentication modules, for example to authenticate end entities by using existing customer databases or security systems. For example, DASCOS Inc. provides a DCE Kerberos authentication module.

Sample code provided with Certificate Management System demonstrates how to write custom authentication modules. The sample code is provided in the CMS Software Development Kit (SDK) located in the following directory:

```
<server_root>/cms_sdk/samples
```

For information about ways customized authentication modules can be used during enrollment, see "Some Enrollment Scenarios" on page 33.

Policy Modules

A policy module is a rule (implemented as a Java class) that validates the contents of a certificate request and formulates the contents of the certificate to be issued. Policy modules are also responsible for accepting, rejecting, or deferring the request. Certificate Management System policies have nothing to

do with export control policies or certificate usage policies. For an introduction to the role of policy modules in the enrollment process, see “Authentication and Policy Modules” on page 29.

Certificate Management System supports the following constraints-specific policy modules out of the box. These policies establish rules or constraints that Certificate Management System must use to evaluate an incoming request. They can be used with either a Certificate Manager or a Registration Manager.

- **DSAKeyConstraints.** Allows the server to certify only DSA keys of specified lengths.
- **IssuerConstraints.** Allows the server to check for certificates that have been issued by a particular CA.
- **KeyAlgorithmConstraints.** Allows the server to certify only those keys that are generated using one of the specified algorithms, such as RSA or DSA.
- **PinPresentConstraints.** Allows the server to check for the presence of end users' PINs in the authentication directory when they request a certificate and accordingly approves or rejects the request.
- **RenewalConstraints.** Allows or rejects requests for renewal of expired certificates.
- **RenewalValidityConstraints.** Enforces the number of days before which a currently active certificate can be renewed and a new validity period for the renewed certificate.
- **RevocationConstraints.** Allows or rejects requests for revocation of expired certificates.
- **RSAKeyConstraints.** Allows the server to certify only RSA keys of specified lengths.
- **SigningAlgorithmConstraints.** Allows the server to specify the signature algorithm to be used by the CA (a Certificate Manager) to sign certificates.
- **SubCANameConstraints.** Allows the server to check for issuer name uniqueness and prevents issuance of multiple subordinate CA certificates with same issuer names.

- **UniqueSubjectNameConstraints.** Allows the server to check for certificate subject name uniqueness and prevents issuance of multiple certificates with same subject names.
- **ValidityConstraints.** Causes the server to check whether the validity period of a certificate falls within a specified period.

Certificate Management System supports the following policy modules out of the box for formulating certificate extensions. They can be used with either a Certificate Manager or a Registration Manager.

- **AuthInfoAccessExt.** Adds the Authority Information Access extension to certificates. The extension specifies how the application validating the certificate can access information, such as on-line validation services and CA policy statements, about the CA that has issued the certificate in which the extension appears.
- **AuthorityKeyIdentifierExt.** Adds the Authority Key Identifier extension to certificates of a specified type. The Authority Key Identifier extension identifies the public key corresponding to the private key used to sign a certificate. This extension is useful when an issuer has multiple signing keys (for example, due to CA certificate renewal).
- **BasicConstraintsExt.** Adds the Basic Constraints extension to certificates of a specified type. This extension is used during the certificate chain verification process to identify CA certificates and to apply certificate chain path length constraints.
- **CertificatePoliciesExt.** Adds the Certificate Policies extension to certificates. The extension contains a sequence of one or more policy statements, each indicating the policy under which the certificate has been issued and identifying the purposes for which the certificate may be used.
- **CertificateRenewalWindowExt.** Adds the Certificate Renewal Window extension to certificates. The extension specifies how to renew a certificate automatically and when automatic renewal should be attempted.
- **CertificateScopeOfUseExt.** Adds the Certificate Scope of Use extension to SSL client certificates. This extension specifies internet addresses where the certificate can be presented for SSL client authentication. This restriction prevents any private information that might be contained in the certificate from being released to servers not explicitly contained in the scope of use.

- **CRLDistributionPointsExt.** Adds the CRL Distribution Points extension to certificates. This extension identifies one or more locations from where the application that is validating the certificate can obtain the CRL information.
- **ExtendedKeyUsageExt.** Adds the Extended Key Usage extension to certificates. The extension identifies one or more purposes—in addition to or in place of the basic purposes indicated in the key usage extension—for which the certified public key may be used.
- **GenericASN1Ext.** Adds ASN.1 type custom extension to certificates. This policy enables you to configure Certificate Management System to add custom extensions to certificates.
- **IssuerAltNameExt.** Adds the Issuer Alternative Name extension to certificates. This extension enables binding of or associating Internet style identities, such as Internet electronic mail address, a DNS name, an IP address, and a uniform resource indicator (URI), with the certificate issuer.
- **KeyUsageExt.** Adds the Key Usage extension to certificates of a specified type. This extension defines the purpose of the key contained in the certificate. The Key Usage, Extended Key Usage, Basic Constraints, and Netscape Certificate Type extensions act together to specify the purposes for which a certificate can be used.
- **NameConstraintsExt.** Adds the Name Constraints extension to certificates. The extension is used in CA certificates to indicate a name space within which subject names or subject alternative names in subsequent certificates in a certification path or chain should be located.
- **NSCCommentExt.** Adds the Netscape Certificate Comment extension to certificates. The extension can be used to include textual comments in certificates.
- **NSCertTypeExt.** Adds the Netscape Certificate Type extension to certificates of a specified type. This extension can be used to limit the purposes for which a certificate can be used. It has been replaced by the X.509 v3 extensions extKeyUsage and basicConstraints, but must still be supported in deployments that include Navigator 3.x clients.
- **OCSPNoCheckExt.** Adds the OCSP No Check extension to certificates. The extension, which should be used in OCSP responder certificates only, indicates how OCSP-compliant applications can verify the revocation status of the certificate an authorized OCSP responder uses to sign OCSP responses.

- **PolicyConstraintsExt.** Adds the Policy Constraints extension to certificates. The extension, which can be used in CA certificates only, constrains path validation in two ways. It can be used to prohibit policy mapping or to require that each certificate in a path contain an acceptable policy identifier.
- **PolicyMappingsExt.** Adds the Policy Mappings extension to certificates. The extension lists one or more pairs of OIDs, each pair identifying two policy statements of two CAs. The pairing indicates that the corresponding policies of one CA are equivalent to policies of another CA.
- **PrivateKeyUsagePeriodExt.** Adds the Private Key Usage Period extension to certificates. The extension allows the certificate issuer to specify a different validity period for the private key than the one specified for the corresponding certificate.
- **SubjectAltNameExt.** Adds the Subject Alternative Name extension to certificates of a specified type. This extension includes one or more alternative (non-X.500) names for the identity bound by the CA to the certified public key. It may be used in addition to the certificate's subject name or as a replacement for it.
- **SubjectDirectoryAttributesExt.** Adds a Subject Directory Attributes extension to certificates. The extension is used to specify any desired directory attribute values for the subject of the certificate.
- **SubjectKeyIdentifierExt.** Adds the Subject Key Identifier extension to certificates of a specified type. This extension identifies the public key certified by this certificate. It provides a way of distinguishing public keys if more than one is available for a given subject name, for example after the certificate has been renewed with a new key.

In addition to the modules listed above, sample code provided with Certificate Management System demonstrates how to support additional extensions. The sample code is provided in the CMS Software Development Kit (SDK) located in the following directory:

```
<server_root>/cms_sdk/samples
```

For detailed information about using certificate extensions, see Appendix B, "Certificate Extensions."

Job Scheduler Plug-Ins

The CMS *Job Scheduler* allows you to configure a Certificate Management System to perform a specified action at a specified time, such as informing a user of the need to renew a certificate or removing an expired certificate from the directory. The scheduler checks at specified intervals for jobs waiting to be executed; if the specified execution time has arrived, the scheduler initiates the job.

You can use standard CMS job plug-ins or write your own Java plug-in class in much the same way that you can write your own authentication and policy modules. Plug-in classes are provided out of the box for scheduling the following jobs:

- **Renewal notification.** Notifies end entities by email that their certificates are about to expire and must be renewed. This job also sends a summary of such notices to agents. Available for Certificate Manager only.
- **Request in queue.** Notifies agents at regular intervals of the state of the request queue. Alternatively, an event-driven notification can be sent whenever a request has been added to the request queue; see the next section for details. Available for Registration Manager or Certificate Manager.
- **Directory expiration update.** Updates a specified LDAP publishing directory periodically by removing expired certificates. This can be useful for end entities such as Netscape Enterprise Server 3.x that rely on the presence or absence of the certificate for authentication purposes, or if you wish to ensure that only current, valid certificates can be found in the directory. This job also sends a summary of removed certificates to agents or administrators. Available for Certificate Manager only.

Event-Driven Notifications

The Certificate Manager and Registration Manager support two kinds of event-driven notifications:

- **Request-completion status.** Automatically notifies users by email that a requested certificate has been issued or that a request has been deferred or rejected. Available for Registration Manager or Certificate Manager.

- **Request-queue status.** Automatically notifies agents by email when a request has been added to the request queue. Available for Registration Manager or Certificate Manager.

Registration Manager

A Registration Manager is a trusted subsystem to which a Certificate Manager can delegate responsibility. A Registration Manager cannot issue or revoke certificates by itself; instead, it evaluates end-entity requests and forwards them to a Certificate Manager for action, such as the issuing of a certificate.

A Registration Manager is designed to handle certificate life-cycle management tasks—that is, the tasks required to maintain a certificate throughout its life cycle, including the following:

- enrolling end entities (initial authentication and initiation to the PKI)
- enforcing policies such as request validation requirements, authentication requirements, and certificate formulation
- distributing issued certificates
- publishing issued certificates to an LDAP directory (LDAP 2.0 or higher)
- coordinating certificate renewal
- coordinating end-entity private encryption key storage with a Data Recovery Manager

A Registration Manager's default forms for end-entity interactions can be used as is or customized. For more information about default Registration Manager forms, see “End Entities and Life-Cycle Management” on page 48.

Certificate Manager

A Certificate Manager can be configured to accept requests from end entities, from Registration Managers, or from both end entities and Registration Managers. When set up to work with a remote Registration Manager, the Certificate Manager processes requests and returns the signed certificates to the Registration Manager, which distributes them to end entities.

Basic capabilities of the Certificate Manager (as distinct from the Registration Manager) include the following:

- can be configured as either a root CA or a subordinate CA

- can accept certificate requests directly from end entities and/or Registration Managers
- can issue end-entity, Registration Manager, and Certificate Manager certificates
- can issue single key-pair or dual key-pair certificates
- can notify users and administrators of approaching certificate expiration
- can renew certificates
- can revoke certificates
- can publish certificates and CRLs to an LDAP directory (LDAP 2.0 or higher)

Although it is possible to configure a Registration Manager to publish certificates to an LDAP directory, the Certificate Manager maintains a complete record of issued certificates, so it is recommended that publishing tasks be performed by the Certificate Manager only.

The Certificate Manager can issue certificates with the following characteristics:

- X.509 version 3
- internationalized subject names
- customized components in subject names
- customized extensions

Signing Algorithms

The Certificate Manager supports the following signing algorithms for both certificates and CRLs:

- RSA with MD2
- RSA with MD5
- RSA with SHA-1
- DSA with SHA-1

Certificate Revocation Lists

The Certificate Manager can issue X.509 v1 or v2 CRLs. A CRL can be automatically updated whenever a certificate is revoked or at specified intervals.

CRL extensions supported include the following:

- **Authority key identifier.** Identifies the public key to be used to validate the digital signature on the certificate.

- **CRL number.** A sequential number unique to each CRL issued by a given CRL issuer. This number allows CRL-checking software to ensure that all previous CRLs have been received.
- **Issuer alternative name.** Associates the CRL issuer with an Internet style identity, such as Internet electronic mail address, a DNS name, an IP address, or a uniform resource indicator (URI).
- **Issuing distribution point.** The URL at which this CRL is maintained.

The Delta CRL indicator extension is not supported.

CRL entry extensions supported include the following:

- **Hold instruction code.** Indicates the action to be taken for an entry that appears on the CRL because it has been placed on hold.
- **Reason code.** Indicates the reason the certificate was revoked.
- **Invalidity date.** Indicates the date on which the private key corresponding to the public key certified by the certificate was (or is suspected to have been) compromised.

Data Recovery Manager

A Data Recovery Manager provides facilities for archiving and recovering private RSA encryption keys. This crucial element of a PKI allows an authorized Data Recovery Manager agent to recover an encryption key that has been lost or corrupted. It also allows administrators to recover encryption keys for employees who have left the company or who are unavailable for some other reason. In either case, once the encryption key has been recovered, the user or administrator can use it to decrypt any data (such as saved email messages) that was encrypted with that key.

A Data Recovery Manager can be used with dual key pairs only—that is, with end entities that support a signing key pair and signing certificate and an encryption key pair and encryption certificate for each identity, and that also support archival of encryption keys. Dual key pairs allow an end entity to get a new signing certificate and signing key pair without changing the encryption certificate or encryption key pair. Similarly, an end entity or an administrator can recover a lost encryption key without changing the signing certificate or signing key pair.

The Data Recovery Manager uses two special key pairs in the process of archiving an end entity's encryption key: a transport key pair (and certificate) and a storage key pair. The end entity must also have two key pairs: a signing key pair and an encryption key pair. The roles of all these keys are summarized in Table 1.3.

Table 1.3 Key pairs used by end entities and key pairs used by the Data Recovery Manager

End-entity key pairs		Data Recovery Manager key pairs	
Signing key pair	Encryption key pair	Transport key pair	Storage key pair
Public signing key: used by recipients to validate digital signature	Public encryption key: used by others to encrypt messages sent to owner	Public transport key: used by end-entity software to encrypt the end entity's private encryption key before sending it to Certificate Management System for storage.	Public storage key: used to decrypt an end entity's stored private encryption key after m of n recovery agents have authorized the recovery operation.
Private signing key: used by owner to digitally sign messages	Private encryption key: used by owner to decrypt messages encrypted with the public key	Private transport key: used by Data Recovery Manager to decrypt an end entity's private encryption key	Private storage key: used to encrypt an end entity's private encryption key for long-term storage

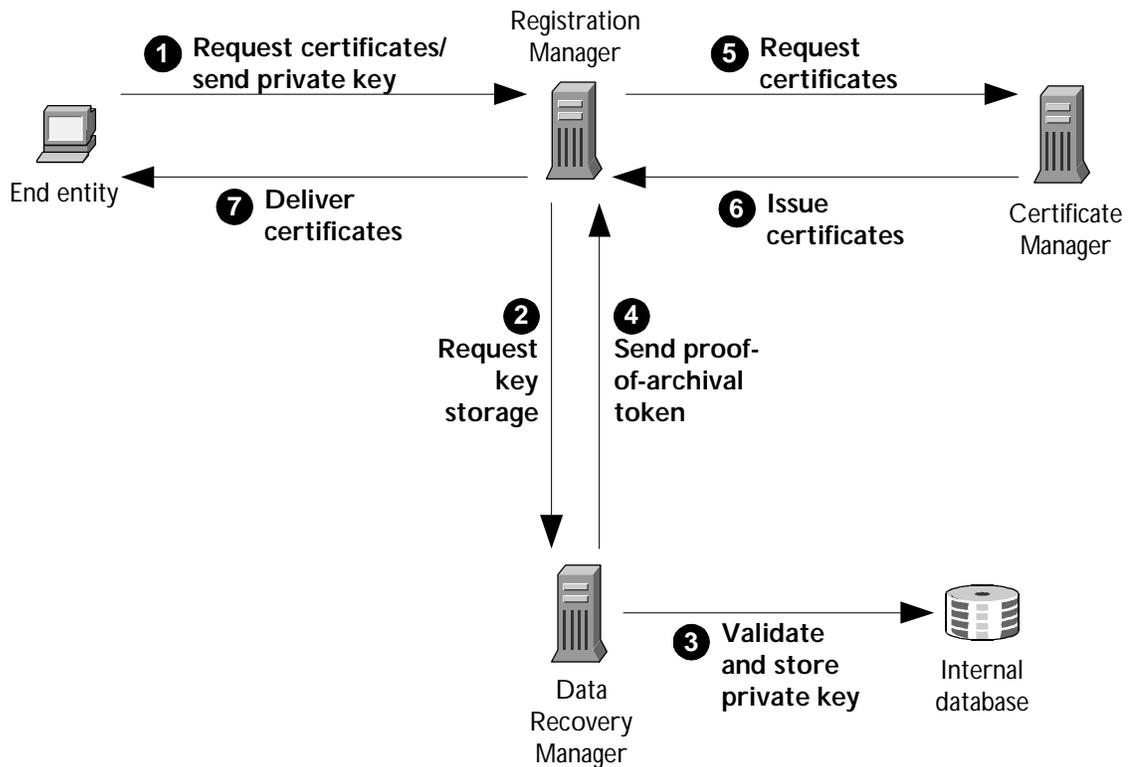
The following steps summarize the key storage process during end-entity enrollment through a Registration Manager. Figure 1.11 illustrates these steps.

1. After the user completes and submits an enrollment form, the end entity generates dual key pairs and sends two certificate requests to the Registration Manager, which detects a request for key archival and requests the private encryption key from the end entity. The end entity then encrypts (or "wraps") its newly minted private encryption key with the Data Recovery Manager's public transport key (obtained from a copy of the transport certificate embedded in the enrollment form) and sends the wrapped private key to the Registration Manager.
2. The Registration Manager sends the end entity's wrapped private encryption key to the Data Recovery Manager as part of a key storage request (which also includes the end entity's public encryption key).

Summary of System Features

3. The Data Recovery Manager uses its private transport key to decrypt the end entity's private encryption key. After confirming that the private encryption key corresponds to the end entity's public encryption key, the Data Recovery Manager encrypts the private encryption key with its private storage key and stores the private encryption key in the CMS internal database.
4. The Data Recovery Manager signs a proof-of-archival token with its private transport key and sends the token to the Registration Manager.
5. The Registration Manager verifies the token and sends the certificate requests on to the Certificate Manager.
6. The Certificate Manager issues the signing and encryption certificates and sends them back to the Registration Manager.
7. The Registration Manager delivers the certificates to the end entity.

Figure 1.11 Key storage process during end-entity enrollment



Data encrypted with the storage key can be retrieved only if m of n “split keys” are provided at the same time by m of n authorized recovery agents. By default, m and n are 2 and 3, respectively. Both values can be changed, as long as m is less than or equal to n .

The Data Recovery Manager indexes stored keys by owner name and a hash of the public key. This arrangement allows for highly efficient searching by name (all stored keys belonging to that owner are returned) or by public key (only the requested key is returned).

Command-Line Utilities

Table 1.4 summarizes the command-line utilities that are bundled with Certificate Management System. For more detailed information about these utilities, see Appendix D and the appendixes that follow in *Netscape Certificate Management System Administrator's Guide*. The binaries for the tools listed in the table are located in the directory `<serverroot>/bin/cert/tools`.

Table 1.4 Command-line utilities bundled with Certificate Management System

Name of utility	Description	Syntax/documentation
AtoB	Converts ASCII base-64 encoded data to binary base-64 encoded data.	AtoB <inputfile> <outputfile>
BtoA	Converts binary base-64 encoded data to ASCII base-64 encoded data.	BtoA <inputfile> <outputfile>
cmsbackup	Script located in the instance directory (not bin/cert/tools). Copies data and configuration files for a CMS instance, the Administration Server, and local Netscape Directory Servers into a compressed archive.	cmsbackup See Appendix B, Backing Up and Restoring Data, in <i>Netscape Certificate Management System Administrator's Guide</i> .
cmsrestore	Restores the data archived by cmsbackup from a named archive or automatically from the last known archive.	cmsrestore <archive path> automatic See Appendix B, Backing Up and Restoring Data, in <i>Netscape Certificate Management System Administrator's Guide</i> .
PrettyPrintCert	Prints the contents of a certificate stored as ASCII base-64 encoded data in a human-readable form.	PrettyPrintCert <inputfile> [<outputfile>]
PrettyPrintCrl	Prints the contents of a CRL stored as ASCII base-64 encoded data in a human-readable form.	PrettyPrintCrl <inputfile> [<outputfile>]

Table 1.4 Command-line utilities bundled with Certificate Management System (Continued)

Name of utility	Description	Syntax/documentation
<code>dumpasn1</code>	Dumps the contents of binary base-64 encoded data.	<code>dumpasn1 [-bcdfhlopsx] <file></code> To see a summary of options, run <code>dumpasn1</code> without specifying any options.
<code>certutil</code> (Certificate Database Tool)	Used to manipulate the certificate database.	See Appendix D, Certificate Database Tool, in <i>Netscape Certificate Management System Administrator's Guide</i> .
<code>keyutil</code> (Key Database Tool)	Used to manipulate the key database.	See Appendix E, Key Database Tool, in <i>Netscape Certificate Management System Administrator's Guide</i> .
<code>migrate/</code> <code>AIX/migrate</code> <code>HP-UX/migrate</code> <code>Solaris/migrate</code> <code>WINNT/migrate.exe</code> (Migration Tool)	Extracts database contents and certificate/key data from a Certificate Server 1.x installation and places the data in three platform-independent files that can then be imported into a CMS installation.	See Appendix A, "Migrating from Certificate Server 1.x," in this guide.
<code>signtool</code> (Netscape Signing Tool)	Used to digitally sign a file, including log files.	See Appendix F, Netscape Signing Tool, in <i>Netscape Certificate Management System Administrator's Guide</i> .
<code>sslstrength</code> (SSL Strength Tool)	Used to connect to an SSL server and report back the type and strength of the encryption cipher that it's using.	See Appendix G, SSL Strength Tool, in <i>Netscape Certificate Management System Administrator's Guide</i> .
<code>ssltap</code> (SSL Debugging Tool)	Used to debug SSL applications.	See Appendix H, SSL Debugging Tool, in <i>Netscape Certificate Management System Administrator's Guide</i> .
<code>setpin</code> (PIN Generator)	Used to generate PINs for end users for directory-based PIN authentication.	See Chapter 11, Using the PIN Generator Tool, in <i>Netscape Certificate Management System Administrator's Guide</i> .
<code>killproc</code>	Used to kill system processes in Windows NT	See Attending to an Unresponsive Server in Chapter 5 of <i>Netscape Certificate Management System Administrator's Guide</i> .

System Architecture

The first five tools listed in Table 1.4 (`AtoB`, `BtoA`, `PrettyPrintCert`, `PrettyPrintCrl`, and `dumpasn1`) are useful for converting back and forth between various encodings and formats you may encounter when dealing with keys and certificates.

The Certificate Database Tool, Key Database Tool, and Security Module Database Tool (which is described in Appendix B of *Managing Servers with Netscape Console*) are useful for a variety of administrative tasks that involve manipulating certificate and key databases.

The Migration Tool converts Certificate Server 1.x data for use with Certificate Management System, and the PIN Generator tool creates PINs for directory authentication.

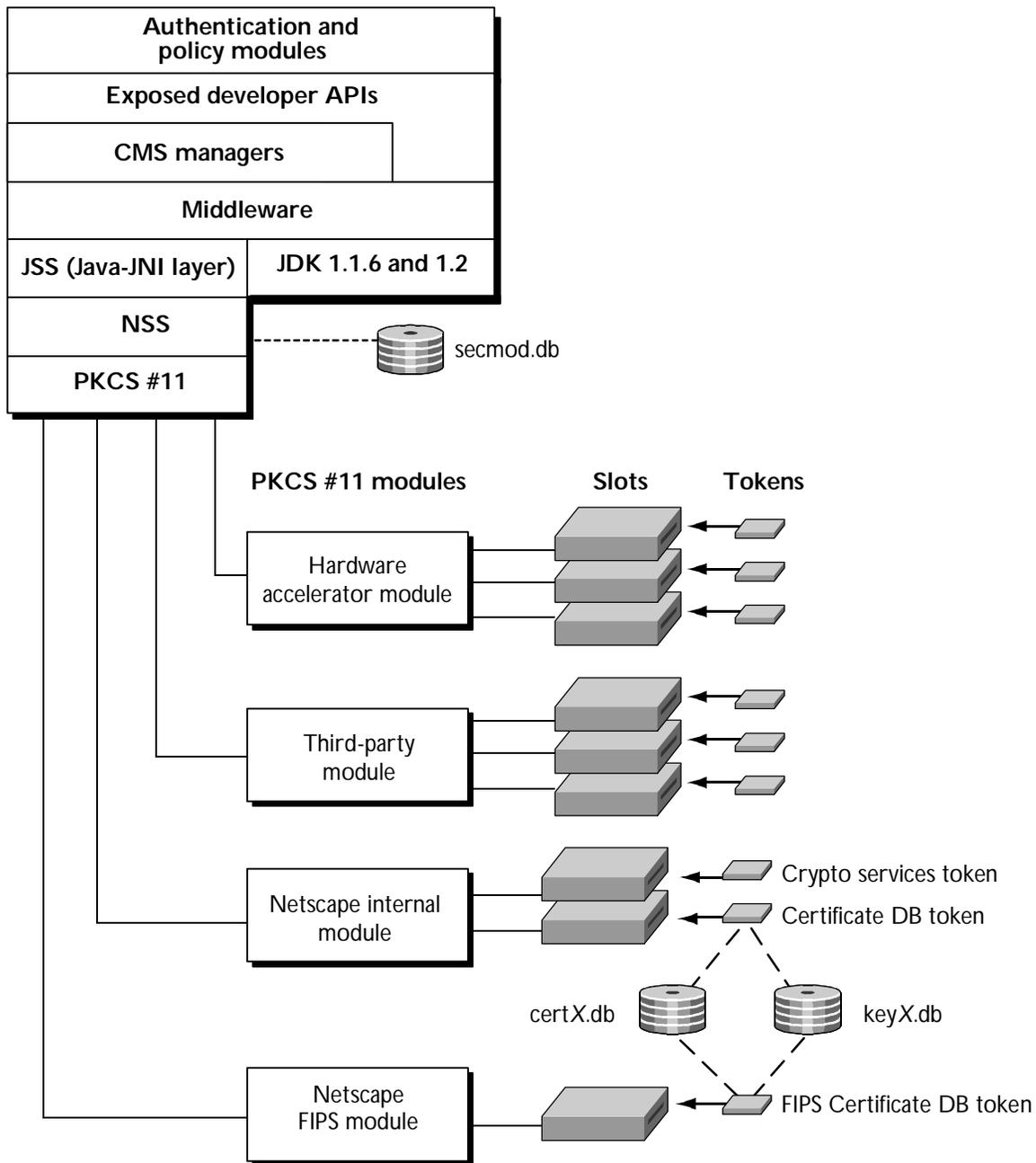
The Netscape Signing Tool associates a digital signature with any file. For information about using this tool to sign CMS logs, see “Signing Log Files” in Chapter 26 of *Netscape Certificate Management System Administrator’s Guide*.

The SSL Strength Tool and SSL Debugging Tool are useful for testing and debugging purposes.

System Architecture

Figure 1.12 shows the internal architecture of Netscape Certificate Management System. The sections that follow describe the basic elements of this architecture, starting at the bottom of the figure.

Figure 1.12 CMS architecture



PKCS #11

Public-Key Cryptography Standard (PKCS) #11 specifies an API used to communicate with devices that hold cryptographic information and perform cryptographic operations. Because it supports PKCS #11, Certificate Management System works with a wide range of hardware and software devices intended for such purposes.

One or more PKCS #11 modules must be available to any CMS subsystem instance. As shown in Figure 1.12, a *PKCS #11 module* (also called a *cryptographic module* or *cryptographic service provider*) manages cryptographic services such as encryption and decryption via the PKCS #11 interface. PKCS #11 modules can be thought of as drivers for cryptographic devices that can be implemented in either hardware or software. Netscape provides a built-in PKCS #11 module with Certificate Management System.

A PKCS #11 module always has one or more *slots*, which can be implemented as physical hardware slots in some form of physical reader (for example, for smart cards) or as conceptual slots in software. Each slot for a PKCS #11 module can in turn contain a *token*, which is the hardware or software device that actually provides cryptographic services and optionally stores certificates and keys.

Netscape provides two built-in modules with Certificate Management System:

- **Default Netscape Internal PKCS #11 Module.** This comes with two built-in tokens:
 - The Internal Crypto Services token performs all cryptographic operations, such as encryption, decryption, and hashing.
 - The Internal Key Storage token (“Certificate DB token” in Figure 1.12) handles all communication with the certificate and key database files (called `certX.db` and `keyX.db`, respectively, where *X* is a version number) that store certificates and keys.
- **FIPS 140-1 module.** This module complies with the FIPS 140-1 government standard for implementations of cryptographic modules. Many products sold to the US government must comply with one or more of the FIPS standards. The FIPS 140-1 module includes a single, built-in FIPS 140-1 Certificate DB token (see Figure 1.12), which handles both cryptographic operations and communication with the `certX.db` and `keyX.db` files.

Any PKCS #11 module can be used with Certificate Management System. The server uses a file called `secmod.db` to keep track of the modules that are available. You can modify this file with the Security Module Database Tool (for details, see Appendix B, “Administration Server Command Line Tools,” in *Managing Servers with Netscape Console*). For example, you need to modify `secmod.db` if you are installing hardware accelerators for use in signing operations.

NSS

Netscape Security Services (NSS) is a set of libraries designed to support cross-platform development of security-enabled communications applications. Applications built with the NSS libraries support the SSL protocol for authentication, tamper detection, and encryption as well as the PKCS #11 interface for cryptographic token interfaces. Netscape uses NSS to support these features in a wide range of products, including Certificate Management System.

As shown in Figure 1.12, NSS communicates with PKCS #11 modules through the PKCS #11 interface and in turn provides the foundation for Java Security Services and higher Java layers.

JSS and the Java/JNI Layer

Java Security Services (JSS) provides a Java interface for security operations performed by NSS. JSS and higher levels of the Certificate Management System architecture are built with the Java Native Interface (JNI), which provides binary compatibility across different versions of the Java Virtual Machine (JVM). This design allows customized subsystem services to be compiled and built just once and run on a range of platforms.

Middleware/JDK 1.1.6 Layers

A middleware layer above JSS and the Java/JNI layer provides a range of services required by the Registration Manager, Certificate Manager, and Data Recovery Manager. The middleware layer is based on Java Development Kit (JDK) 1.1.6, and it underlies both the manager subsystems and the APIs

available to third-party developers for building custom authentication and policy modules. The default authentication and policy modules provided with Certificate Management System are built from the same Java classes.

Authentication and Policy Modules

The top layer of Figure 1.12 consists of authentication and policy modules. Several default modules ship with Certificate Management System; third parties can create their own custom modules using the APIs provided above the middleware and subsystem layers. Modules for all three subsystems work the same way and are interchangeable.

Standards Summary

This section summarizes the standard message formats and protocols supported by Certificate Management System.

Certificate Management Formats and Protocols

Certificate Management System supports the following certificate management formats and protocols. For more details about the proposed PKIX standards listed here, see <http://www.ietf.org/html.charters/pkix-charter.html> (under Internet Drafts).

- **Certificate Enrollment Protocol (CEP)**. A certificate management protocol jointly developed by Cisco Systems and VeriSign, Inc. CEP is an early implementation of CMC (described later in this list). CEP specifies how a device communicates with a CA, including how to retrieve the CA's public key, how to enroll a device with the CA, and how to retrieve a CRL. CEP uses PKCS #7 and PKCS #10.
- **Certificate Request Message Format (CRMF)**. A message format used to convey a request for a certificate to a Registration Manager or Certificate Manager. A proposed standard from the Internet Engineering Task Force (IETF) PKIX working group.

- **Certificate Management Message Formats (CMMF).** Message formats used to convey certificate requests and revocation requests from end entities to a Registration Manager or Certificate Manager and to send a variety of information to end entities. A proposed standard from the IETF PKIX working group. CMMF is subsumed by another proposed standard, CMC (next item).
- **Certificate Management Messages over CMS (CMC).** A general interface to public-key certification products based on CMS and PKCS #10, including a certificate enrollment protocol for DSA-signed certificates with Diffie-Hellman public keys. A proposed standard from the IETF PKIX working group. CMC incorporates CRMF and CMMF. Future versions of Certificate Management System will support this standard as it is finalized.
- **Cryptographic Message Syntax (CMS).** A superset of PKCS #7 syntax used for digital signatures and encryption. A proposed standard from the IETF PKIX working group.
- **PKIX Certificate and CRL Profile (PKIX Part 1).** The first part of the four-part standard under development by the IETF for a public-key infrastructure for the Internet. Part 1 deals with specifications for certificates and CRLs. Certificate Management System will support the other PKIX parts as they are finalized. For more information about PKIX Part 1, see <ftp://ftp.isi.edu/in-notes/rfc2459.txt>.

Security and Directory Protocols

Certificate Management System supports the following security and directory protocols:

- **FIPS PUBS 140-1.** Federal Information Standards Publications (FIPS PUBS) 140-1 is a US government standard for implementations of cryptographic modules—that is, hardware or software that encrypts and decrypts data or performs other cryptographic operations (such as creating or verifying digital signatures).
- **Hypertext Transport Protocol (HTTP) and Hypertext Transport Protocol Secure (HTTPS).** Protocols used to communicate with web servers.

- **KEYGEN tag.** An HTML tag supported by Netscape browsers that generates a key pair for use with a certificate. For more information, see <http://www.netscape.com/eng/security/comm4-keygen.html>.
- **Lightweight Directory Access Protocol (LDAP) v2, v3.** A directory service protocol designed to run over TCP/IP and across multiple platforms. LDAP is a simplified version of Directory Access Protocol (DAP), used to access X.500 directories. LDAP is under IETF change control and has evolved to meet Internet requirements.
- **Public-Key Cryptography Standard (PKCS) #7.** An encrypted data and message format developed by RSA Data Security to represent digital signatures, certificate chains, and encrypted data. This format is used to deliver certificates to end entities.
- **Public-Key Cryptography Standard (PKCS) #10.** A message format developed by RSA Data Security for certificate requests. This format is supported by many server products and by Microsoft Internet Explorer.
- **Public-Key Cryptography Standard (PKCS) #11.** Specifies an API used to communicate with devices such as hardware tokens that hold cryptographic information and perform cryptographic operations.
- **X.509 v1, v3.** Digital certificate formats recommended by the International Telecommunications Union (ITU).
- **Secure Sockets Layer (SSL) 2.0, 3.0.** A set of rules governing server authentication, client authentication, and encrypted communication between servers and clients.

Default Demo Installation

This chapter describes how to set up a simple installation that demonstrates the basic capabilities of a Certificate Manager with an integrated Registration Manager. It is intended for administrators who are already familiar with PKI concepts. An experienced administrator should be able to install and set up the default demo in less than an hour, then use it to try out basic Netscape Certificate Management System procedures.

Warning This chapter describes how to install a Certificate Manager for demonstration purposes only. The steps described require that you accept most of the default values suggested at each stage of installation and configuration. Before you attempt to install more sophisticated pilots or a full-scale deployment, you should read Chapter 3, “Planning Your Deployment,” and the chapters that follow.

This chapter has the following sections:

- System Requirements (page 78)
- Overview of the Default Demo (page 80)
- Installing the Default Demo (page 84)
- Using the Default Demo (page 95)

System Requirements

This section summarizes the basic software and hardware requirements for any machine on which you intend to install Certificate Management System instances and related software:

- Operating System and Software Required (page 78)
- Platform Requirements (page 78)
- Other Requirements (page 79)

Operating System and Software Required

Operating systems supported:

- Windows NT 4.0 with Service Pack 4 or 5
- Solaris 2.6 and Solaris 7
- OSF/1 v4.0D
- AIX 4.3

Other required software:

- Netscape Administration Server 4.2 (included)
- Netscape Directory Server 4.12 (included)
- Browser software that supports SSL

Platform Requirements

Each platform has slightly different requirements. In addition to the requirements listed below, make sure you have ample swap space or virtual memory allocated for the system on which you intend to install Certificate Management System.

UNIX Platform Requirements

RAM: 128 MB (recommended)

Hard disk storage space of approximately 250 MB total, broken down as follows:

- Total transient space required during installation: approximately 100 MB
- Hard disk storage space for installation:
 - space required for setup, configuration, and running the server: approximately 100 MB
 - additional space to allow for database growth in pilot deployment: approximately 50 MB (this may be reduced to 10 MB for default demo installation)
 - total disk storage space for installation: approximately 150 MB

Windows NT Platform Requirements

NT Service Pack 4 or 5

128 MB of RAM (recommended)

Pentium 166 or faster

Hard disk storage space of approximately 250 MB total, broken down as follows:

- Total transient space required during installation: approximately 100 MB
- Hard disk storage space for installation:
 - space required for setup, configuration, and starting the server: approximately 100 MB
 - additional space to allow for database growth during deployment: approximately 50 MB (this may be reduced to 10 MB for default demo installation)
 - total disk storage space for installation: approximately 150 MB.

Other Requirements

- On Unix systems, you must install as `root` in order to use well-known port numbers (such as 443) that are less than 1024. If you do not plan to use port numbers less than 1024, you do not need to install as `root`. If you plan

to run the administration server as root, you should also install as root and specify the default user and group, `nobody`, as the system id for other server processes.

Overview of the Default Demo

The default demo installation described in this chapter is intended to provide a quick, hands-on experience of the basic Certificate Management System interfaces. It is intended for demonstration purposes only and relies on a number of default settings that may not be appropriate for a mission-critical installation. Before you attempt to install more sophisticated pilots or a full-scale deployment, read Chapter 3, “Planning Your Deployment,” and the chapters that follow.

The default demo installation includes the following Netscape software:

- **Netscape Console.** Netscape Console is described in a separate guide, *Managing Servers with Netscape Console*. It is a stand-alone Java application used to manage Netscape server instances with the aid of a configuration directory and a user directory. For this demo, Netscape Console controls just the server instances listed here; the configuration and user directories are combined in a single Netscape Directory Server instance. In real deployments, Netscape Console can be set up to control a variety of servers in different instances and on different machines that are registered with a single configuration directory, which could potentially be separate from the user directory.
- **Netscape Administration Server.** This lightweight HTTP server acts as the back end to Netscape Console. An instance of Administration Server manages operation requests involving any Netscape servers installed in the same server root, or *server group*, and invokes CGI programs to perform these operations. For this demo, a single Administration Server instance provides administrative access to the Directory Server instance and Certificate Manager instance listed below—the only other server instances in the same server group.
- **Configuration and User Directory (Netscape Directory Server).** This is an instance of Netscape Directory Server with two subtrees. The user subtree keeps track of users and groups and their privileges (for the Administration Server, not for CMS). The configuration subtree keeps track of the location on the network of Netscape servers. For this demo, the

configuration subtree keeps track of itself, the Administration Server instance, the single instance of Certificate Management System, and a separate instance of Netscape Directory Server that serves as the internal database for Certificate Management System. For this demo, the user subtree is also used as the user and group directory for directory-based authentication and publishing.

- **Certificate Manager.** For this demo, the single instance of Netscape Certificate Management System contains a Certificate Manager that is configured to perform registration tasks as well as CA tasks.
- **Internal Database (Netscape Directory Server) for Certificate Management System.** For each instance of Certificate Management System you install an instance of Netscape Directory Server that acts as the internal database for certificate and request information.

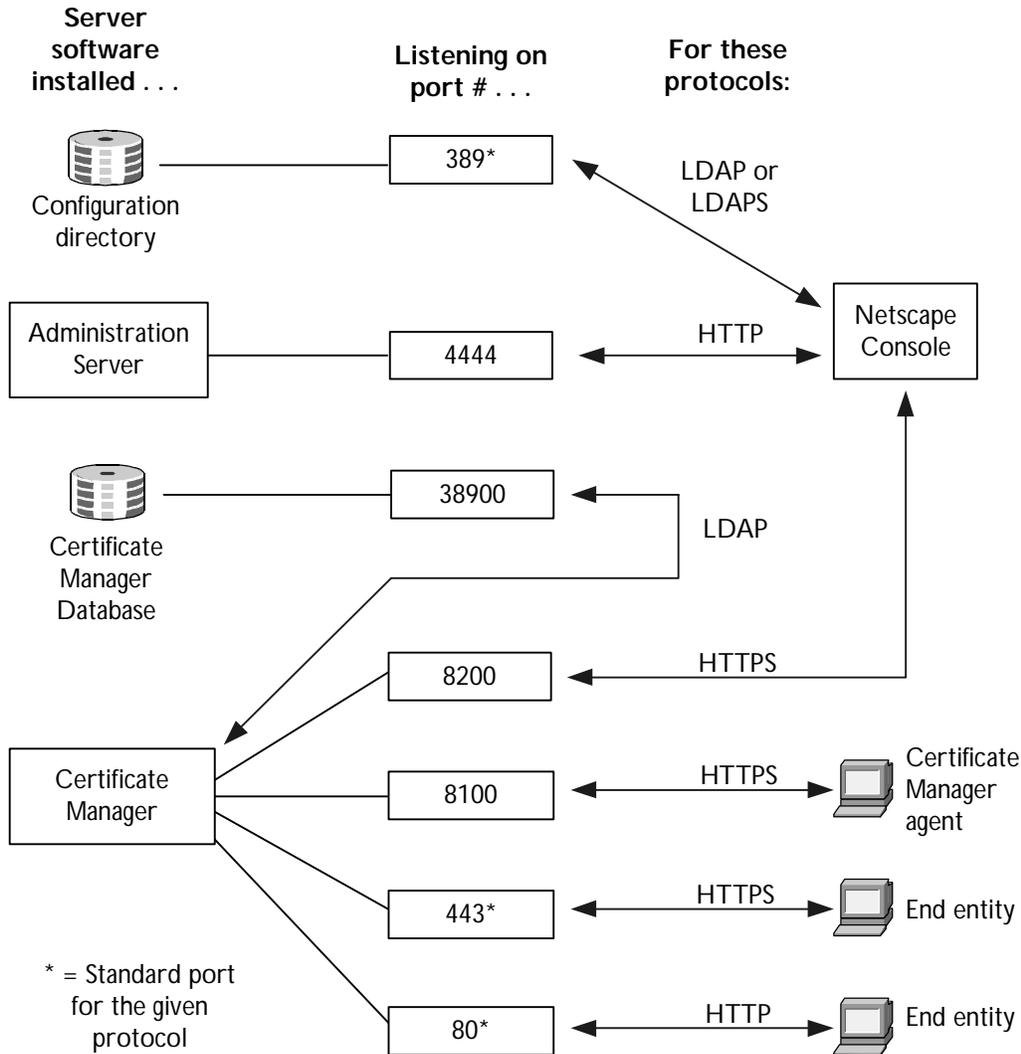
You use the main window of Netscape Console to perform basic tasks such as starting and stopping a server. To manage any server controlled by Netscape Console (in this case, just Directory Server and the Certificate Manager), first locate it on the left side of the main Netscape Console window, then double-click the icon to open a separate administrative window for that server.

Netscape Console uses the configuration directory for information on the locations and contents of server groups on the network. It also interacts with the Administration Server for each server group to perform some tasks, such as managing SSL encryption settings. However, to manage settings displayed in the Netscape Console window for a particular Certificate Management System instance, Netscape Console acts directly on a configuration file stored with that instance. (For more information about the configuration file, see *Netscape Certificate Management System Administrator's Guide*.)

As you proceed with the default demo installation and configuration, you will be asked to assign several port numbers, names, and passwords. Figure 2.1 shows the four main software elements of the demo and the port numbers and protocols they use for different purposes. Using the default ports for the end-entity URLs helps users because they will not need to remember port numbers; any HTTPS request will try port 443 if no port is specified in the URL.

Overview of the Default Demo

Figure 2.1 Software installed and port numbers assigned for the default demo



You will also be asked to provide additional information, such as the name of each server instance to be installed, the names and passwords of various types of administrators, and information related to the CA signing certificate and SSL server certificate that the Certificate Manager must have available before it can begin operation.

To keep things simple for the default demo, most of the information requested during installation is set either to a default or to some arbitrary, convenient value. Before you attempt to install more sophisticated pilots or a full-scale deployment, you should read Chapter 3, “Planning Your Deployment,” and the chapters that follow to determine the precise names and settings that are appropriate for your situation.

Another difference between the default demo and more sophisticated installations is that the Directory Server instance, in addition to providing both the configuration directory and the user directory, is also used to publish and test certificates you issue with the Certificate Manager instance. In a real-world deployment, the Directory Server Instance used for configuration and for users is unlikely to be used for publishing.

Demo Passwords

The demo that you install is a real CA that can issue certificates. Even if you plan to remove it after testing, you should maintain the security of the demo system. For this reason, the installation procedure does not give specific passwords for each administrative user. However, to avoid confusion, the passwords that you will need are identified here and are later referred to by this identification. If you make a list of the passwords you decide on, be sure to keep the list secure.

You will need to provide the following passwords during the installation process:

- | | |
|---------------------------------------|--|
| <code><admin password></code> | Administrator for both Administration Server and its configuration directory. Use this password to start Netscape Console and the Installation Wizard. |
| <code><dir mgr password></code> | Manager for the configuration directory. (This password must be at least eight characters.) |

Installing the Default Demo

<code><intdb password></code>	Administrator for the CMS internal database (an instance of Directory Server). This password is kept and protected in a special cache that you access with the <code><single-signon password></code> .
<code><CMS password></code>	CMS administrator. Use this password to access Netscape Console's CMS window.
<code><token password></code>	Password for the CMS key database. This password is kept and protected in a special cache that you access with the <code><single-signon password></code> .
<code><single-signon password></code>	This password protects the <code><intdb password></code> and <code><token password></code> . Use this password to start Certificate Management System.

Installing the Default Demo

The installation script installs and starts an Administration Server and a Directory Server; the process is slightly different for Windows NT and Unix systems. The Installation Wizard, which is the same on both systems, installs Certificate Management System itself and creates the system's certificates. When you have finished installing the files, you start Certificate Management System and enroll for the initial administrator-agent certificate, which you then use to verify that the system is properly installed and functions correctly.

The steps of this installation procedure are described in the following sections:

- Step 1. Run the Installation Script - Unix (page 85) or Step 1. Run the Installation Script - Windows NT (page 87)
- Step 2. Run the Installation Wizard (page 89)
- Step 3. Get the First User Certificate (page 93)

Step 1. Run the Installation Script - Unix

These instructions assume that you have the initial distribution of Certificate Management System available, either on a CD or on your hard disk.

If you are using a Windows NT system, see “Step 1. Run the Installation Script - Windows NT” on page 87.

To run the installation script, change to the distribution directory (where you have downloaded the distribution files) and execute the file `setup`.

In the instructions that follow, the question that appears at the bottom of each setup screen is in boldface, followed by the action you should take.

1. **Would you like to continue with setup? [Yes]:** Press Enter.
2. **Do you agree to the license terms? [No]:** Type `yes` and press Enter.
3. **Select the items you would like to install [1]:** Press Enter.
4. **Server root [/usr/netscape/server4]:** Press Enter to accept the default server root directory. (If you are not installing as `root`, you probably will not have permission to create directories in `/usr` so you will have to choose another location.)
5. **Specify the components you wish to install [All]:** Press Enter to accept the default.
6. **Specify the components you wish to install [1,2,3]:** Press Enter to accept the default server product components.
7. **Specify the components you wish to install [1,2]:** Press Enter to accept the default Directory Suite components.
8. **Specify the components you wish to install [1,2]:** Press Enter to accept the default Administration Services components.
9. **Specify the components you wish to install [1, 2]:** Press Enter to accept the default CMS components.
10. **Computer name [myhost.mydomain.com]:** Press Enter to install on the local machine.

11. **System User [nobody]:** Enter the user that the configuration/user Directory Server process will run as. Where your system supports it, accept the default user, `nobody`, creating that user as necessary.
12. **System Group [nobody]:** Enter the group that the configuration/user Directory Server process will run as. Where your system supports it, accept the default group, `nobody`, creating that group as necessary.
13. **Do you want to register this software with an existing Netscape configuration directory server? [No]:** Press Enter to install a new configuration directory.
14. **Do you want to use another directory to store your data? [No]:** Press Enter to use the new configuration directory as your user/group directory.
15. **Directory server network port [389]:** Press enter to accept the default, 389. If you are not installing as root or if 389 is in use, the default will be a random number; you may want to change this number to something easy to remember, such as 38989.
16. **Directory server identifier [myhost]:** Type `configdir` as the unique identifier for the configuration directory, and press Enter.
17. **Netscape configuration directory server administrator ID [admin]:** Press Enter to accept the default, then enter the `<admin password>`.
18. **Suffix [o=mydomain.com]:** Press Enter to accept the default.
19. **Directory Manager DN [cn=Directory Manager]:** Press Enter to accept the default, then enter the `<dir mgr password>`.
20. **Administration Domain [mydomain.com]:** Press Enter to accept the default.
21. **Administration port [random #]:** Type `4444` and press Enter.
22. **Run Administration Server as [root]:** Press Enter to accept the default.
23. **Netscape Certificate Management System Server identifier [localhost]:** Type `cmsdemo` and press Enter. After the script copies the files and updates the system, which may take a few minutes, press Enter to continue.

The first phase of the installation is now complete. The installation script has installed Netscape Console, installed and started an Administration Server and its configuration directory, and copied the files for Certificate Management System. You are now ready to configure the Certificate Management System instance by running the Installation Wizard.

Step 1. Run the Installation Script - Windows NT

These instructions assume that you have the initial distribution of Certificate Management System available, either on a CD or on your hard disk.

If you are using a Unix system, see “Step 1. Run the Installation Script - Unix” on page 85.

1. To run the installation script, open the distribution directory for the system software you are using and double-click the file `setup.exe`.
In the instructions that follow, the name that appears in the title bar of each setup screen is in bold, followed by a description of the action you should take.
2. **Welcome.** Click Next.
3. **Software License Agreement.** Click Yes.
4. **Select Server or Console Installation.** Leave the default setting (Netscape Servers) selected and click Next.
5. **Choose Installation Directory.** Leave the default setting (C:\Netscape\Server4) selected and click Next.
6. **Select Products.** Leave all four components selected and click Next.
7. **Directory Server 4.12.** Leave the default setting (“This instance will be the configuration directory server”) selected and click Next.
8. **Directory Server 4.12.** Leave the default setting (“Store data in this directory server”) selected and click Next.

9. **Directory Server 4.12 Server Settings.** Type the following values, then click Next:
 - Server identifier:** configdir
 - Server port:** Accept the default, which should be 389
 - Suffix:** Accept the default, which should be your company's domain name, in the form o=mydomain.com.
10. **Directory Server 4.12 Netscape Configuration Directory Server Administrator.** Type the following values, then click Next:
 - Configuration Directory Administrator ID:** admin
 - Password:** <admin password>
 - Password (again):** <admin password>
11. **Directory Server 4.12 Administration Domain.** Accept the default, which should be your company's domain name, in the form mydomain.com.
12. **Directory Server 4.12 Directory Manager Settings.** Type the following values, then click Next:
 - Directory Manager DN:** cn=Directory Manager
 - Password:** <dir mgr password>
 - Password (again):** <dir mgr password>
13. **Administration Server Port Selection.** Type the value 4444 and click Next.
14. **Netscape Certificate Management System Server identifier.** Type the value cmsdemo and click Next.
15. **Configuration Summary.** Click Next.
16. **Setup.** At this point, the installation script extracts and installs the binaries for all of the servers in the server root directory and creates and starts instances of the Administration Server and Directory Server. This process may take a few minutes.
17. **Setup Complete.** Leave the default setting ("Restart my computer now") and click Finish.

The first phase of the installation is now complete. The installation script has installed Netscape Console, installed and started an Administration Server and its configuration directory, and copied the files for Certificate Management System. You are now ready to complete the installation of Certificate Management System by running the Installation Wizard.

Step 2. Run the Installation Wizard

To begin running the Installation Wizard, follow these steps:

1. Start Netscape Console:
 - On a Windows NT system, click Start, then choose Programs, then Netscape Server Family, then Netscape Console 4.2. Alternatively, click the Netscape Console shortcut in the Netscape Server Family directory that opens on your desktop after setup completes.
 - On a Unix system, open a command shell, change to the directory `/usr/netscape/server4`, and execute the file `startconsole`.

2. Log in as `admin`, giving the password `<admin password>`.

The main window of Netscape Console appears.

If the Administration URL is not filled in, enter `http://<myhost>:4444/`

3. In the navigation tree at the left, open your computer, then open Server Group.
4. Select `cert-cmsdemo`.
5. In the Netscape Certificate Management System panel at the right, click Open.

After a few moments, the Installation Wizard appears. You use the wizard to get the initial certificates and set the initial configuration for this demo instance of Certificate Management System.

Installing the Default Demo

In the instructions that follow, the panel title that appears below the title bar for each screen is in boldface, followed by the action you should take.

1. **Introduction.** Click Next.
2. **Internal Database.** Type the following values, then click Next:

Instance ID: Accept the default (cmsdemo-db).

Port number: Accept the default (38900).

Directory Manager DN: cn=internal directory manager

Password: <intdb password>

Password (again): <intdb password>

At this point the system creates the internal database, which can take some time.

3. **Administrator.** Type the following values, then click Next:

Administrator ID: CMSadmin

Full name: Accept the default value.

Password: <CMS password>

Password (again): <CMS password>

4. **Subsystems.** Click Next to accept the default selection (Certificate Manager only).
5. **Remote Data Recovery Manager.** Click Next to accept the default selection (No).

At this point the system configures the internal database, which can take some time.

6. **Network Configuration.** Select the Enable checkbox to enable the non-SSL end-entity gateway, then accept the default values listed below. If one of the default ports is unavailable, a different, randomly selected port will appear in the form.

SSL administration port: 8200

SSL agent port: 8100

SSL end-entity port: 443

Enable: Select this checkbox to enable the non-SSL end-entity gateway.

Non-SSL end-entity port: 80

7. **Server Migration from Certificate Server 1.x - Step 1.** Click Next to accept the default selection (No).
8. **CA's serial number range.** Click Next to accept the default (start at 0x1 with no upper limit).
9. **CA Signing Certificate.** Click Next to accept the default selection (Create self-signed CA certificate).
10. **Key-Pair Information for Certificate Manager CA Signing Certificate.** Type the following values, then click Next:
 - Token:** Accept the default value (Internal).
 - Password:** <token password>
 - Password (again):** <token password>
 - Key type:** Accept the default value (RSA).
 - Key length:** Select 1024 and leave the custom key-length field blank.
11. **Message Digest Algorithm.** Click Next to accept the default (SHA1).
12. **Subject Name for Certificate Manager CA Signing Certificate.** Type the following values, then click Next:
 - Common name (CN=):** Demo CA
 - Organization Unit (OU=):** CMS Demo
 - Organization (O=):** *name of your company*
 - Locality (L=):** *name of your locality*
 - State (ST=):** *name of your state, province, or territory*
 - Country (C=):** *two-letter code for your country*
13. **Validity Period for Certificate Manager CA Signing Certificate.** Modify year and month values of "Expire on" date to allow a validity period of one month from the installation date, then click Next.
14. **Certificate Extensions for Certificate Manager CA Signing Certificate.** Click Next to accept the default selections.
15. **Certificate Manager CA Signing Certificate Creation.** Click Next.
16. **SSL Server Certificate.** Click Next to accept the default selection (Sign SSL certificate with my CA signing certificate).

17. **Key-Pair Information for Server SSL Certificate.** Change the Key length to 1024, accept the default values for other fields, then click Next.
18. **Message Digest Algorithm.** Click Next to accept the default (SHA1).
19. **Subject Name for SSL Server Certificate.** Type the following values, then click Next.

Common name (CN=): *your local host name, in the form
mymachine.mydomain.com*

Organization Unit (OU=): CMS Demo

Organization (O=): *name of your company*

Locality (L=): *name of your locality*

State (ST=): *name of your state, province, or territory*

Country (C=): *two-letter code for your country*

20. **Validity Period for SSL Server Certificate.** Modify year and month values of “Expire on” date to allow a validity period of one month from the installation date, then click Next.
21. **Certificate Extensions for SSL Server Certificate.** Click Next to accept the default selections.
22. **SSL Server Certificate Creation.** Click Next.

The generation of the certificate can take some time.

23. **Set Up Single Signon Password.** Type the following values, then click Next:

Single signon password: <single-signon password>

Single signon password (again): <single-signon password>

24. **Configuration Status.** Click Done.

Certificate Management System starts automatically.

The installation and configuration of Certificate Management System is now complete, and the Certificate Manager is running.

The user interface of Certificate Management System is available through the web gateways whose ports you specified during installation. You can access them directly in a web browser by going to those ports using the appropriate protocol.

- The SSL agent gateway URL is:
`https://myhost.mydomain.com:8100`
- The SSL end-user gateway URL is:
`https://myhost.mydomain.com:443`
- The non-SSL end-user gateway URL is:
`http://myhost.mydomain.com:80`

Step 3. Get the First User Certificate

After you complete configuration of Certificate Management System with the Installation Wizard, you must enroll for a certificate for the first agent. This is the first user certificate that Certificate Management System issues.

The initial user is both an administrator and an agent. This person can use Netscape Console to create additional agents with the appropriate user privileges and use Agent Services to issue them certificates. Since there is no agent yet to approve the request, a special enrollment form allows you to get this first certificate automatically.

After you submit this initial Administrator/Agent Certificate Enrollment form, it is automatically disabled, so that no one else can acquire a certificate without agent approval or some form of automated authentication. The system automatically adds the initial user to the list of agents.

Enrolling for the First Agent Certificate

To enroll for the first agent certificate, you should be working at the computer you intend to use as the agent, so that the new certificate will be installed in the browser you will be using to access the Agent Services pages. Follow these steps:

1. Open a web browser window.
2. Go to the URL for the SSL agent port (8100).

For example:

```
https://myhost.mydomain.com:8100
```

The first time you access this port, the system opens the Administrator/Agent Certificate Enrollment form.

Because you have accessed an SSL port, Certificate Management System presents its SSL server certificate to your browser for authentication. This is the SSL server certificate that you just created during installation. Because you just created it, it is not on your list of trusted certificates. A series of dialog boxes now appears that lets you add the CMS server certificate to your list of trusted certificates.

3. Complete the dialog boxes as instructed (the exact procedure depends on the browser you are using).
4. In the Administrator/Agent Certificate Enrollment form, enroll for a client SSL certificate as the system's first privileged user by entering the following information:

Authentication Information

User ID: CMSadmin

Password: <CMS password>

Subject Name

Full name: CMS Administrator

Login name: CMSadmin

Email address: *your email address*

Organization unit: CMS Demo

Organization: *name of your company*

User's Key Length Information

Key Length: Select 1024 (High Grade)

Note that the validity period of this initial agent certificate is hard-coded as one year.

5. Click Submit.
6. Follow the instructions your browser presents as it generates a key pair.
If authentication is successful, the new certificate will be imported into your browser. You should make a backup copy of the certificate.

Now you have a client authentication certificate in the name CMS Administrator. This special user name, which you specified as the initial administrator for Certificate Management System during installation, has now been designated as the first agent. The certificate you just created allows you to access the Agent Services pages. As an agent, you can approve enrollment

requests and start issuing new certificates. To access the CMS windows in Netscape Console, you use the CMS administrator user ID and the CMS password.

If You Need the First Agent Form Again

After you submit the initial Administrator/Agent Certificate Enrollment form, it is no longer available from the agent port. If something goes wrong and you are unable to obtain the initial agent certificate, you must reset a parameter in the configuration file to make the initial Administrator/Agent Certificate Enrollment form available again. Follow these steps:

1. In the left frame of Netscape Console, open `cert-cmsdemo`.
The server requests your `<CMS password>`.
2. Click the icon labeled Stop the Server.
3. Go to the directory `<server root>/cert-cmsdemo/config`, open the file `CMS.cfg` in a text editor, and find the following line:

```
agentGateway.enableAdminEnroll=false
```
4. Change `false` to `true`, and save the file.
5. Start the server from the CMS window where you stopped it.
Alternatively, right-click on `cert-cmsdemo` in the left frame and choose Start Server.
6. Enter your `<single-signon password>`.
The next time you access `https://myhost.mydomain.com:8100`, the Administrative Enrollment form will be available again.

Using the Default Demo

You have now performed a basic installation and can use the installed demo Certificate Manager to issue certificates. This section provides the following exercises with which you can test the installation and practice using the system:

- **“Verify the Installation,”** (page 96): Accessing the various web gateways and using the default versions of the forms to enroll for and issue a certificate.
- **“Create a Policy,”** (page 101): Configuring the Certificate Manager to reject certificate requests that do not use at least 1024-bit key lengths.
- **“Use an LDAP Directory,”** (page 103): Adding a user to the configuration directory you just installed and using directory-based authentication to enroll as that user.
- **“Publish Certificates in a Directory,”** (page 106): Publishing client certificates to the directory.
- **“Send Renewal Reminders,”** (page 111): Configuring the Certificate Manager to send out automatic renewal reminders to entities whose certificates will be expiring soon.

Verify the Installation

To verify that the installation is correct and complete, you will access each of the different gateways for the various user interface pages: the SSL and non-SSL end-user pages, and the Agent Services pages for the Certificate Manager. You will use each set of pages to perform a basic task.

- In “Viewing Issued Certificates From the Agent Gateway,” you will view a list of the certificates that the demo CA has issued so far.
- In “Enrolling for a Certificate From the End-Entity Gateway,” you will enroll for a certificate by using the manual enrollment procedure.
- In “Finding and Approving a Certificate Request,” you will approve the new certificate enrollment request and issue a new agent certificate.
- In “Testing Your New Certificate,” you will use the new agent certificate to access the agent gateway.

Note In a real installation, you would probably not give users access to both gateways or to all the enrollment choices and other possible actions in the pages. You access both end-user gateways here simply for testing purposes, not because these particular actions need to be performed from these locations.

Viewing Issued Certificates From the Agent Gateway

1. In a web browser window, use HTTPS to go to the URL for the SSL agent port that you specified. For example:

`https://myhost.mydomain.com:8100`

2. Because this is an SSL connection, you are prompted to present your client SSL certificate for authentication. Choose the certificate you received on initial enrollment.

The Agent Services entry page appears.



3. Click Services Summary.

The Services Summary page appears, giving you access to all the gateways.



4. Click End Users Services.

The Enrollment tab for the non-SSL end-entity gateway appears.

5. Click the Retrieval tab.

The form that appears is for the first option, List Certificates.

6. Type 0x0 into the field labeled “Lowest serial number,” then click Find to list the certificates that the Certificate Manager has issued so far.

If you followed the instructions in this chapter exactly, you should see three certificates listed: the CA signing certificate (CN=Demo CA), the Certificate Manager SSL server certificate (CN=<your hostname>), and your initial agent certificate (CN=CMS administrator).

7. Use the browser’s Back button to go back to the Services Summary page. (For example, when using Communicator, press and hold the mouse button while it’s over the Back button, then choose Index from the pop-up menu.)

Enrolling for a Certificate From the End-Entity Gateway

After following the previous procedure, your browser will be at the Services Summary page. Follow this procedure to submit an enrollment request through the end-entity gateway.

1. Click SSL End-Users Services.

The Enrollment tab for the SSL end-entity gateway appears.

2. Use the Manual User Enrollment form that appears to enroll for a certificate.

For Full Name, type the name `user1`, so you will recognize this certificate as distinct from your administrator’s certificate. When you have finished filling it out, submit the form.

3. Follow the instructions your browser presents as it generates a key pair.

After the key pair has been generated, the Certificate Manager displays a notice that the certificate request has been submitted, including a request ID.

4. Use the browser’s Back button to go back to the Services Summary page. (For example, when using Communicator, press and hold the mouse button while it’s over the Back button, then choose Index from the pop-up menu.)

Finding and Approving a Certificate Request

After following the previous procedure, your browser will be at the Services Summary page. Follow this procedure to approve the enrollment request you just submitted. This procedure will issue a certificate from the request that can be used as an agent certificate.

1. Click Agent Services, then click Certificate Manager Agent Services.

To access this page, your browser must present your client SSL certificate to authenticate your identity.

2. If a dialog box appears requesting that you select a certificate, select the certificate name that begins with CMS Administrator.

The first form for the Agent Services gateway appears—the List Requests form.

3. Select “Show enrollment requests” for Request Type.

4. Select “Show Pending Requests” for Request status, and then click Find.

One request should be returned: the request you just made through the SSL end-user gateway, which is marked as pending.

5. Click the Details button next to the pending request.

6. Scroll down to the last section of the Request Details form, labeled Privileges.

7. Select the checkbox labeled “This certificate is for a Certificate Manager agent,” then type a user ID for the new agent.

This user ID can be the same (User1) that you specified in the certificate request, or it can be some other ID that you want to use to identify this agent in the CMS window of Netscape Console, such as Agent1.

8. At the bottom of the form, select “Accept this request” and click Do It.

The certificate is issued immediately. The Request Details form is replaced by a form announcing that the certificate has been generated, along with its serial number.

9. Click Show Certificate to view the new certificate.

At the bottom of the page is a button labeled Import Your Certificate. Normally, you would mail this page to the requestor, or the Certificate Manager would mail the requestor an automatic notification containing the certificate and instructions.

10. Since you made the request yourself from this computer, go ahead and click Import Your Certificate to import the certificate into your browser.

You have now designated `User1` as an agent. Since you have already issued a certificate in the name of `User1`, you can now present that certificate to access the Agent Services pages. `User1` is an agent, but not an administrator; as `User1`, you can manage certificate requests, but you cannot access Netscape Console's CMS window to configure the system.

Setting Your Browser to Use the Agent Certificate

To verify that the `User1` certificate really can access the agent pages, you must first set your browser to use the `User1` certificate to identify you to web sites. To do this in Communicator 4.x, for example, follow these steps:

1. Click the Security button in the Navigation toolbar near the top of the window.
2. Click Navigator in the left-hand frame.
3. From the drop-down list labeled "Certificate to identify you to a web site," select your `User1` certificate.
4. Click OK.

Testing Your New Certificate

Clear the browsers cached security information so that it will ask for a new certificate when you view the agent gateway.

1. Go to any other web page that is not part of Agent Services (such as `http://home.netscape.com`).
2. Return to the Agent Services pages at the URL for the SSL agent port that you specified. For example:

`https://myhost.mydomain.com:8100`

You should be able to access the Agent Services pages without any difficulty, as long as you are using the same computer from which you requested and imported the `User1` certificate.

Before you continue, you might want to try accessing the new installation from another computer and with a different login. Try enrolling for user certificates from there, using both the SSL and non-SSL end-user gateways. If you wish, you can also enroll for additional agent certificates. You will have to return to the computer from which you requested and imported your `CMSAdmin` and `User1` certificates to access the Agent Services pages and approve the requests.

Create a Policy

Policies are rules that you define that are applied to requests before a certificate is issued. Certificate Management System provides configurable policies that allow you to enforce your organization's requirements for certificates. You can configure different policies to be applied to different requests based on criteria such as the type of request or which Registration or Certificate Manager received the request. You can find out more about policies in Chapter 16, "Introduction to Policy," in *Netscape Certificate Management System Administrator's Guide*.

In a real PKI deployment, you would probably formulate your policies before installing any software, and configure how the policies will be implemented before issuing any certificates. For this demonstration, you will implement a simple but very useful rule before you start issuing certificates.

You will create a policy that requires all certificate requests use RSA key pairs that are 1024-bit or longer. This ensures that all of the certificates you issue meet a minimum level of security. Later, you will try to enroll for a certificate using a shorter-length key pair (512 bits) to show how the request is rejected automatically by the policy.

Policies do not always result in acceptance or rejection: they can also be used to modify certificate attributes such as the validity period or certificate extensions. In the "Create a Policy" exercise, you create a policy that will reject requests that do not have at least 1024-bit keys. In the "Use an LDAP Directory" exercise, you will try to enroll using a 512-bit key to see how the policy works.

Configuring an RSA Key Length Policy

1. Start Netscape Console:
 - On a Windows NT system, click Start, then choose Programs, then Netscape Server Family, then Netscape Console 4.2.
 - On a Unix system, open a command shell, change to the directory `/usr/netscape/server4`, and execute the file `startconsole`.
2. Log in as `admin`, giving the password `<admin password>`.

The main window of Netscape Console appears.
3. In the navigation tree on the left, open your computer, then open Server Group.
4. Select CMS (`cert-cmsdemo`).
5. In the Netscape Certificate Management System panel at the right, click Open.
6. Log in as `CMSadmin`, giving the password `<CMS password>`.

Netscape Console's CMS window appears, showing the Tasks tab.
7. In the CMS window, click the Configuration tab.
8. In the navigation tree on the left, open the Certificate Manager folder and click Policies.
9. From the list of policies in the Policy Rules Management tab, select `RSAKeyRule` (the second policy in the list) then click Edit/View.
10. In the Policy Editor dialog box, provide the following information:

minSize: 1024
maxSize: 2048
exponents: accept the default setting
enable: true
predicate: certType==client

The `predicate` indicates that this policy will be applied to certificate requests for client certificates only. The `minSize` sets the minimum allowed length for the RSA key pair used to generate the request; requests with shorter RSA keys will be rejected. The policy is turned on for all requests to this Certificate Manager by setting `enabled` to `true`.

11. Click OK to save the changes. The `RSARule` should now be listed as enabled in the Policy Rules Management tab.

That is all you need to do. The policy will now be enforced on all requests for client certificates. You will see how this policy works in the next part of the demonstration when you enroll for a client certificate.

Use an LDAP Directory

To test using Certificate Management System with an LDAP directory, you will use Netscape Console's CMS window to enable directory-based authentication using the configuration directory that you installed with the demo. You will add a user (`User2`) to the directory, then enroll for a certificate as `User2`, using directory-based enrollment.

You will first try to enroll using 512-bit keys; the enrollment will fail because of the policy requiring 1024-bit keys. After you submit a new request with a 1024-bit key, Certificate Management System should authenticate the user information in the directory and issue the certificate automatically.

To use directory-based authentication to enroll entities,

- Enable Directory-Based Authentication (page 103)
- Add a User to the Directory (page 104)
- Enroll with Directory-Based Authentication (page 105)

Enable Directory-Based Authentication

To enable directory-based authentication for the Certificate Manager:

1. If the CMS console window is not still open, start Netscape Console again (or go back to the main window) and open the window for CMS.
2. In the CMS console window, select the Configuration tab, then select Authentication in the navigation tree.

3. On the Authentication Instance tab, click Add.
4. In the Select Authentication Plugin Implementation dialog box, select UidPwdDirAuth and click Next.
5. In the Authentication Instance Editor dialog box, provide the following information:

Authentication Instance ID: UserDirEnrollment

dnpattern: cn=\$attr.cn,c=US

ldapStringAttributes: Leave blank

ldapByteAttributes: Leave blank

ldap.ldapconn.host: *your host name*

ldap.ldapconn.port: 389

ldap.ldapconn.secureConn: false

ldap.ldapconn.version: 2

ldap.basedn: o=*mydomain.com*

ldap.minConns: 3

ldap.maxConns: 5

6. Click OK.

Note If you leave the `dnpattern` field blank, the `dnpattern` used by default is `E=$attr.mail,CN=$attr.cn,O=dn.o,C=$dn.c`. This pattern works well with Communicator and other browsers. For the demo, you used a simpler `dnpattern` to avoid configuring other things. The simpler pattern should not be used for a real deployment. End-entity certificates for use with S/MIME may not work correctly if the `E` attribute is not present. Certificate display will not work correctly if the `C` and `O` attributes are left out.

Add a User to the Directory

The users and groups of your organization are kept in the organization's global directory. Since you are using the configuration directory that you installed with the demo to simulate such a global directory, you must add a user to the configuration directory's user and groups subtree. (Notice that this is a different operation from adding a user or group to the Certificate Manager's internal database.)

To add a user to the configuration directory's subtree for users and groups:

1. Start Netscape Console again, or go back to the main window.
2. Select the Users and Groups tab and click Create (in the lower right corner).
3. In the Select Organization Unit dialog box, select People and click OK.
4. In the Create User dialog box fill out the required fields as follows:

First Name: User
Last Name: Two
Full Name: User Two
User ID: User2
Password: <User2 password>
Confirm password: <User2 password>
E-Mail: *your email address*

5. Click OK.

You can see that User Two has been added to the list of users.

Enroll with Directory-Based Authentication

Now that there is a user in the authentication directory, you can test directory-based authentication. In order to show the key length policy working, you will request the certificate using a 512-bit key first, then change the request to use a 1024-bit key.

1. Open a browser and go to the SSL end-user gateway:

`https://mymachine.mydomain.com:443`

2. In the Enrollment panel under User Enrollment, click Directory-based.
3. Fill out the enrollment form as follows:

User ID: User2
Password: <User2 password>
Key Length: select 512 (Low Grade)

4. Click Submit.

A dialog box asks whether to generate a private key.

5. Click OK, and provide your key database password if requested.

After the key is generated, your browser submits the certificate request to the Certificate Manager. The Certificate Manager verifies the request against all applicable policies (including the RSA key length policy for client certificates you configured earlier). The response from the server will be a Request Rejected page explaining that the request violated the `RSakeyRule` policy.

6. Use your browser's Back button to return to the Directory-based enrollment form. If the identity information is no longer present, enter the User ID and Password again.
7. Change the Key Length setting to 1024 (High Grade), and click Submit. A dialog box asks whether to generate a private key.
8. Click OK, and provide your key database password if requested.

The new certificate is issued immediately and installed in your browser.

Next, you will configure Certificate Management System to publish (in the directory) the certificate you just issued.

Publish Certificates in a Directory

In any PKI there are things that you need to publish to make them available to entities. Certificate revocation lists (CRLs), for example, can be made available at a well known URL so that clients and servers can check them as needed instead of fetching and storing the list every time it is updated. In a PKI where people need to exchange encrypted files or email, you do not want each person to have to store everyone else's public key; instead, you can publish certificates to a directory or database and allow users to look up public keys as needed.

In this example, you will configure a Certificate Manager to publish new certificates to an existing directory (the configuration directory that Netscape Console uses).

To publish certificates to a directory, you must configure information about the destination directory, configure the rules for publishing to it, then update the directory. Updating the directory publishes certificates that were issued before publishing was enabled; certificates issued later will be published automatically as they are issued.

Before you change the configuration you should understand the basics of the flexible components that make up the Certificate Management System publishing system: mappers, publishers, and rules.

Mappers translate objects (such as certificates) in the internal database into some other form for publishing. You will configure an LDAP mapper to translate the user name in a client certificate request to a distinguished name (DN) in the publishing directory.

Publishers are objects that actually publish the data. You will not configure the publisher here, but the `LdapUserCertPublisher` finds the DN that the mapper produces and adds a `certificate` attribute to its entry. The value of the attribute, of course, is the client certificate (in a binary form).

Rules coordinate the use of a mapper with a publisher for objects that meet certain conditions. The conditions may simply require a certain type of object (such as a client certificate). A condition may also assert some additional requirement (a predicate) that must be true about that type of object in order to invoke the rule. You will not configure any rules in this example. By default, the Certificate Manager uses a rule to coordinate the `LdapUserCertMap` and the `LdapUserCertPublisher` for publishing client certificates.

Configure the Publishing Destination

To enable publishing and configure the directory where certificates will be published:

1. If the CMS window is not still open, start Netscape Console again (or go back to the main Console window) and open the window for CMS.
2. Open the Certificate Manager folder and select Publishing.
3. Check the Enable Publishing checkbox then the Enable LDAP Publishing checkbox.

The Destination area becomes editable.

4. Enter information in the Destination area to identify the directory to which you want to publish (use the configuration directory, where User Two's entry is stored):

Host Name: *your local host name, in the form
mymachine.mydomain.com*
Port Number: 389
Directory Manager DN: cn=Directory Manager
Password: <dir mgr password>
Password (again): <dir mgr password>
Version: 2
Authentication: Basic authentication

5. Click Save.

A dialog box appears that indicates whether CMS is able to connect, authenticate, and bind to the directory.

If your configuration is not successful, make sure that the entries you make in the Destination area correspond to how you configured the Configuration Directory Server when you ran the `setup` program.

Directory publishing is now enabled. Certificate Management System will publish any new certificates to the directory according to the publication rules. The next step is to set those rules.

Set Rules for Publishing Certificates

In this section, you configure CMS to map client certificates to `People` entries in the `o=mydomain.com` directory tree using the user ID from the certificate request.

To configure CMS to publish user certificates to an LDAP directory:

1. Open the CMS console window and select the Configuration tab.
2. Open the Certificate Manager folder and double-click Publishing.
3. Below Publishing in the navigation tree, click Mappers.
4. In the Mappers Management tab, select `LdapUserCertMap` and click Edit/View.
5. Change the `dnPattern` parameter value to `UID=$req.UID, OU=people, O=mydomain.com`.

This pattern will cause the mapper to formulate a DN using the user ID from the certificate request (the data entered in the User ID field on the end entity enrollment form) and fixed values for OU and O.

6. Click OK.

CMS can now publish user certificates in the configuration directory. You do not need to configure the Publisher or Rule. If you want to see more about how the rule works, look at the `LdapUserCertRule` under Rules (using the Edit/View button) and the `LdapUserCertPublisher` under Publishers.

Update the Publishing Directory

Your Certificate Manager is now configured to automatically publish newly issued client certificates. If you want to experience this, you can follow the instructions in “Add a User to the Directory” and “Enroll with Directory-Based Authentication” again to add a new user and enroll for a certificate.

Use the procedure in this example to view the new user’s directory entry and see the certificate published automatically (certificates are published every 20 minutes, so you may need to wait a few minutes before a new certificate is published).

In the example here, you conclude by manually updating the directory with the issued (but unpublished) certificate for User Two. You will look at User Two’s directory entry before and after publishing to see how the entry changes.

To view the directory entry for User Two:

1. Go to the Netscape Console main window, select the configuration directory (`configdir`) in the navigation tree, and then click Open.
2. Click the Directory tab.

The directory information trees are represented in the navigation tree on the left.

3. Open the entry for your domain (for example, `mcom.com`).
4. Select the People node in the entry for your domain.

The right side of the window lists the People entries. (If you have followed the examples, User Two will be the only entry.)

5. Double-click the User Two entry to open the Edit Entry dialog box.

6. Click Advanced at the bottom of the dialog box to see all of the attributes for User Two in the Property Editor dialog box.

User Two has attributes for Email address, First name, etc., but no certificate.

7. Click Cancel to close the Property Editor dialog box, but leave the Edit Entry dialog box open if you can: you will open the Property Editor again after you manually publish certificates.

To publish certificates to the directory manually:

1. In a browser, go to the URL for the SSL agent port. For example:

```
https://myhost.mydomain.com:8100/
```

If you are asked to select a certificate for client authentication, be sure not to choose the certificate for User Two since that user does not have administrative privileges.

2. Select Certificate Manager Agent Services.
3. Select Update Directory Server from the list on the left.
4. Check the first checkbox, labeled "Update everything in the database to the directory," then click Update Directory.

After a few seconds a results page displays. Most of the entries will indicate failures because in this example you did not configure publishing rules for most of the object types in the internal database.

The third item in the list should read "Valid certificates have been published in the directory." This means that publishing client certificates was successful.

5. Return to the Edit Entry dialog for User Two (repeat the previous procedure if necessary) and click Advanced to open the Property Editor.

The first attribute listed is now the Certificate for User Two. The certificate is in an unreadable binary form, so you do not see any actual data.

You have successfully configured the Certificate Manager to publish client certificates to an LDAP directory.

Send Renewal Reminders

Certificate Management System provides a facility for scheduling automatic jobs. The jobs facility can help you manage the certificate lifecycle by automating processes such as removing revoked certificates from your data store or notifying end-entities when their certificates are about to expire.

This exercise will show you how to use the jobs facility to send out automatic renewal reminders to entities. You will configure CMS to send email to entities starting 400 days before the certificate expires. In a real deployment, of course, you would probably not start reminding certificate holders to renew until 30 days before expiration. You will see the email that is sent to a certificate holder and a summary report of all notices that can be sent to a CMS agent.

To complete this exercise, you need to have access to a host that can receive Simple Mail Transfer Protocol (SMTP) requests and send mail. By default, CMS configures `localhost` (the machine on which it is running) as the mail server. Many UNIX hosts run SMTP daemons (such as `sendmail`) in their default configurations, so in UNIX you may not need to change the CMS defaults. Windows NT systems, however, do not typically run SMTP daemons by default and you will probably need to configure the SMTP settings in CMS.

If you are sure that the machine on which CMS is running is also capable of receiving SMTP requests on port 25, skip to “Configuring CMS to Send Renewal Reminders.”

Otherwise, find out the name of host that can accept SMTP requests and follow the next procedure, “Configuring a Mail Server for CMS,” to configure CMS.

Configuring a Mail Server for CMS

To configure the server from which CMS can send mail:

1. Open the CMS console window and select the Configuration tab.
2. Click the SMTP tab.
3. Type the hostname of your mail server in the “Server name” field.
4. Enter the port number your server uses for SMTP in the Port Number field.

If you are certain that your server uses a port number other than 25 for SMTP, enter it in the “Port number” field. However, it is unlikely that any server uses a different number for the well-known SMTP service.

5. Click Save.

Configuring CMS to Send Renewal Reminders

To configure Certificate Management System to send renewal reminders:

1. Open the CMS console window and select the Configuration tab.
2. Open Job Scheduler in the navigation tree.
3. Select Jobs.
4. Select certRenewalNotifier in the Job Instance tab.
5. Click Edit/View.

The Job Instance Editor dialog box displays. By default this job is enabled and scheduled to notify end-entities 30 days before their certificates expire. You will change the settings so that renewal notices begin 400 days before the certificate expires (so you will get notices for the certificates issued during this demonstration). You will also send notices every minute (instead of every day) so that you get an immediate message, and send a summary report to yourself.

6. Make sure the following parameters have the listed values:

enabled: true
cron: * * * * * (include spaces between the asterisks)
notifyTriggerOffset: 400
senderEmail: *your email address*
summary.enabled: true
summary.recipientEmail: *your email address*
summary.senderEmail: *your email address*

7. Click OK.
8. Select Job Scheduler in the Configuration tab’s navigation tree.

The next step will turn on the Job Scheduler. Once the scheduler is enabled you will receive at least two email messages every minute. Make sure you turn off the Job Scheduler after a few minutes to avoid a flood of email messages.

9. Select the Enable Jobs Scheduler checkbox.

10. Click Save.

You should begin receiving email after one minute.

11. After the scheduler has been running for a few minutes, deselect the Enable Jobs Scheduler checkbox.

12. Click Save.

13. Check your email.

You will have at least two messages.

Messages with the subject “Certificate Renewal Notification” are examples of notices sent to end entities. By default, these are sent to the address in the email (E) attribute in the certificate subject. These messages explain that the certificate is going to expire on a certain date, and they provide a URL for an end-entity gateway where the certificate can be renewed.

Messages with the subject “Certificate Renewal Notification Summary” are examples of the summary report sent to the address in the job’s `summaryRecipientEmail` parameter (usually a CMS agent). These messages list all of the certificates that are about to expire (according to the job’s `notifyTriggerOffset` parameter) and whether or not the Certificate Manager succeeded in sending a renewal notice.

The message content, format, and subject are all customizable, so in a real deployment you can create messages that better suit your organization.

You have now completed the default demo. Before you attempt to install more sophisticated pilots or a full-scale deployment, you should read Chapter 3, “Planning Your Deployment,” and the chapters that follow.

After you are finished using the demonstration installation, remove it from your system. See Chapter 4, “Installing and Uninstalling CMS Instances” in *Netscape Certificate Management System Administrator’s Guide* for complete uninstallation instructions.

Using the Default Demo

Planning and Installation



Chapter 3 Planning Your Deployment

Chapter 4 Installation Worksheet

Chapter 5 Installation and Configuration

Planning Your Deployment

Before installing Netscape Certificate Management System in any real-life deployment, you first need to plan all aspects of the proposed installation. It's important to consider all potential issues carefully before installation. Omissions or faulty assumptions in the planning process can cause severe problems later.

This chapter provides an overview of the most important decisions you need to make. Many of these decisions are interdependent; for example, the question of whether a Certificate Manager is subordinate affects its distinguished name (DN) as well as its validity period, extensions, and place in the CA hierarchy.

As you begin to make decisions about your deployment strategy, you can use Chapter 4, "Installation Worksheet," to collect the detailed information you must supply during the installation and configuration of individual subsystems.

This chapter has the following sections:

- Topology Decisions (page 118)
- Certificate Authority Decisions (page 128)
- Cryptographic Token Decisions (page 132)
- Publishing Decisions (page 133)
- Subsystem Certificate Decisions (page 134)
- Authentication Decisions (page 137)
- Policy Decisions (page 137)
- Deployment Strategy and Port Assignments (page 138)

Topology Decisions

Certificate Management System allows you to install the Certificate Manager, Registration Manager, and Data Recovery Manager in many different configurations.

Since CAs can delegate some responsibilities to subordinate CAs, a Certificate Manager might delegate responsibilities to one or more levels of subordinate Certificate Managers. Therefore many complex variations are possible. You should carefully consider the appropriate topology for your deployment before you make any other deployment plans.

The sections that follow describe the simplest arrangements:

- Server Groups and CMS Instances (page 118)
- Single Certificate Manager (page 119)
- Certificate Manager and Registration Manager (page 120)
- Certificate Manager and Data Recovery Manager (page 123)
- Certificate Manager, Data Recovery Manager, and Registration Manager (page 125)
- Cloned Certificate Manager

Server Groups and CMS Instances

As described in *Managing Servers with Netscape Console*, Netscape servers installed in a single server root directory are called a *server group* and are managed by a single instance of Netscape Administration Server. A single CMS instance in a server group can contain a single subsystem instance of any kind, or either of the following combinations:

- one Certificate Manager and one Data Recovery Manager
- one Registration Manager and one Data Recovery Manager

The only combination that is not permitted in a single CMS instance is a Certificate Manager with a Registration Manager. A Certificate Manager in a single instance can be configured to provide Registration Manager capabilities as well; a separate subsystem isn't needed.

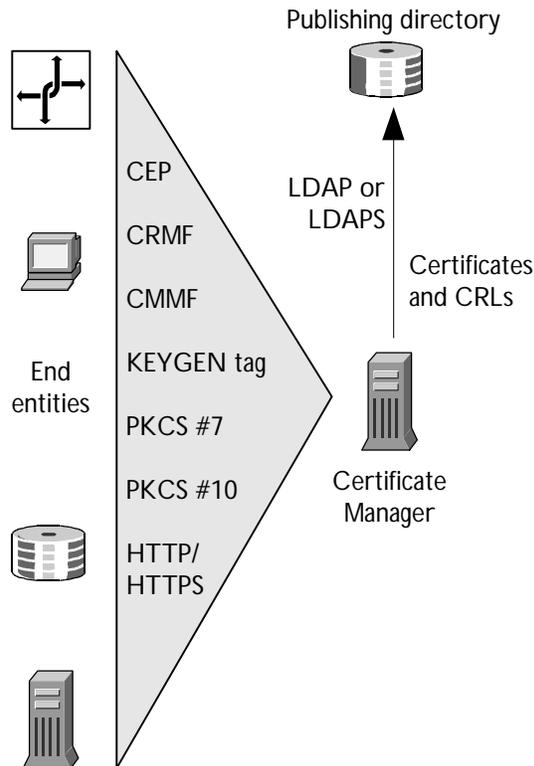
A Certificate Manager and a Registration Manager are always permitted in separate instances, whether the instances are in the same server group, in separate server groups on the same machine, or in separate server groups on separate machines.

Single Certificate Manager

Some deployments may require only a single Certificate Manager that handles all end-entity interactions and provides no key archival and recovery capabilities. This Certificate Manager can use a signing certificate issued by a public certificate authority or its own self-signed CA signing certificate to sign all the certificates it issues.

Figure 3.1 shows the relationships among a single Certificate Manager, end entities, and a publishing directory. The Certificate Manager can publish both end-entity certificates and CRLs to a directory.

Figure 3.1 Single root Certificate Manager



The arrangement shown in Figure 3.1 is equivalent to the capabilities provided by Netscape Certificate Server 1.x—with the addition of new Certificate Management System features such as Digital Signature Algorithm (DSA) signing, support for PKCS #11, and support for a wider variety of end-entity protocols.

Certificate Manager and Registration Manager

Many organizations need to separate the role of the Registration Manager from the role of the Certificate Manager. This separation can be useful, for example, if different groups of end entities are subject to different authentication policies or work in different geographic locations.

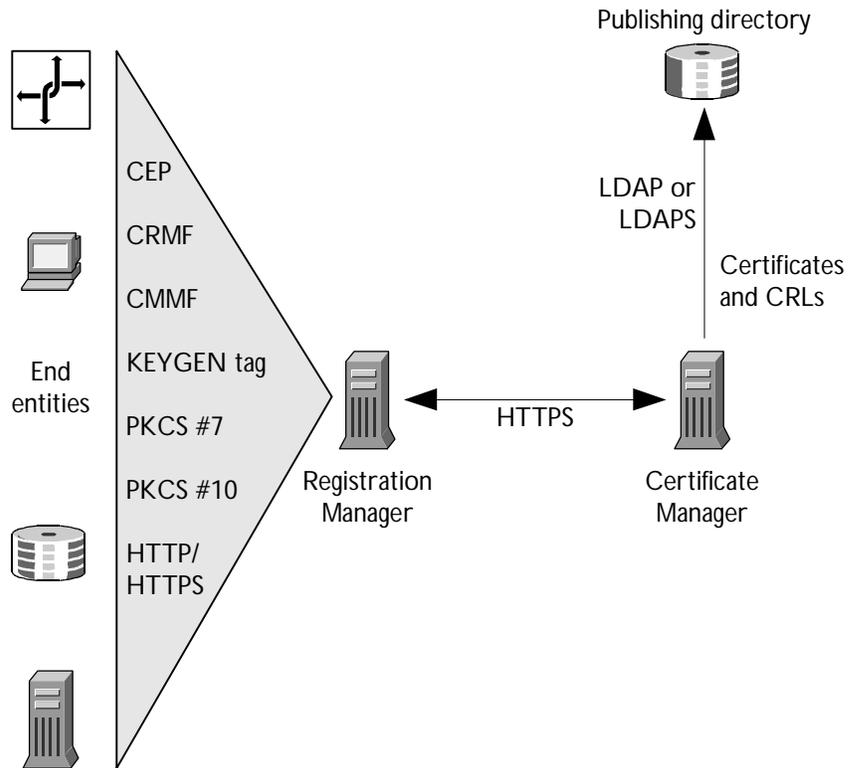
Each group of end entities interacts with a designated Registration Manager that processes requests from end entities and sends them to a Certificate Manager. The Certificate Manager can accept requests from both end entities and Registration Managers. For example, end entities at the home office might deal directly with the Certificate Manager, while end entities at a branch office might deal with their own Registration Manager. Alternatively, the Certificate Manager might be configured to accept requests only from Registration Managers.

A single CMS instance cannot contain both a Certificate Manager and a Registration Manager. A Certificate Manager that needs to interact with end entities other than Registration Managers provides all Registration Manager capabilities itself.

A Registration Manager can be installed in one CMS instance and its related Certificate Manager in another CMS instance. The separate instances can be located in the same server group, in different server groups on the same machine, or on different machines.

Figure 3.2 shows a Registration Manager and its Certificate Manager in separate instances on separate machines. All communication between the Certificate Manager and the Registration Manager takes place over HTTPS.

Figure 3.2 Certificate Manager and Registration Manager in different instances



In many organizations, it may be desirable to deploy multiple Registration Managers that all communicate with a single Certificate Manager. Each separate Registration Manager, for example, might handle all end-entity interactions in a particular geographic area or within an organizational group.

Decisions about the number of, locations of, and relationships among Certificate Managers and Registration Managers depend on many factors. These include firewall considerations, the physical security required for each subsystem, the physical location of the end entities that the Registration Manager is intended to serve, and the physical location of the Certificate Manager agent, Registration Manager agent, and other persons responsible for administering the Certificate Manager and Registration Manager.

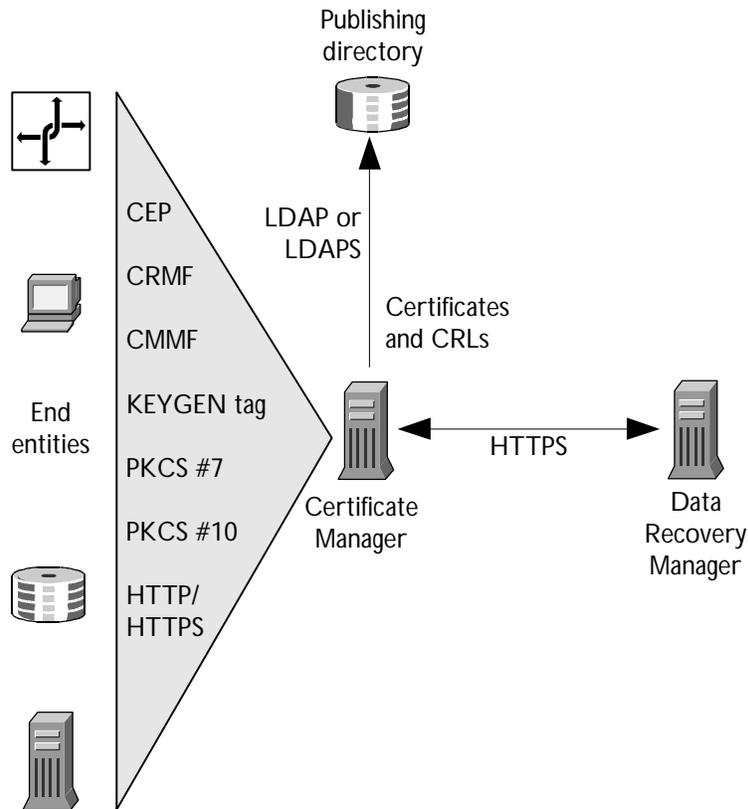
Certificate Manager and Data Recovery Manager

If an organization requires key archival and recovery capabilities—for example, if encrypted mail is widely used and the organization risks data loss if it is unable to recover encryption keys—it can install a Data Recovery Manager. This can be done without regard for the presence or absence of a separate Registration Manager.

For example, to add key storage and recovery to the scenario sketched in Figure 3.1, a Data Recovery Manager can be installed either in the same CMS instance in which the Certificate Manager is installed or in a different CMS instance (which can be located in the same server group on the same machine, in a different server group on the same machine, or on a different machine.)

Figure 3.3 shows a Data Recovery Manager in a separate CMS instance. In this case all communication between the Certificate Manager and the Data Recovery Manager takes place over HTTPS. If the Data Recovery Manager and the Certificate Manager are part of the same CMS instance, all communication takes place internally and the two subsystems do not require separate host names.

Figure 3.3 Certificate Manager and Data Recovery Manager in different instances



The Data Recovery Manager is intended for use with private encryption keys only. Therefore end entities must be using either a browser that supports dual keys or a browser that is using Personal Security Manager, which supports dual keys.

The decision to keep the Data Recovery Manager in the same instance as the Certificate Manager or in a different instance (most likely on a different machine) depends on many factors. These include firewall considerations, the physical security required for each subsystem, and the physical location of the Certificate Manager agent, Data Recovery Manager agent, and other persons responsible for administering the Certificate Manager and recovering keys.

Like a Certificate Manager subsystem, a Data Recovery Manager has special physical security requirements, since a compromised Data Recovery Manager would have devastating security consequences for your entire PKI. You may therefore want to keep the Data Recovery Manager in a special locked room or building, a choice that can affect your deployment strategy.

Certificate Manager, Data Recovery Manager, and Registration Manager

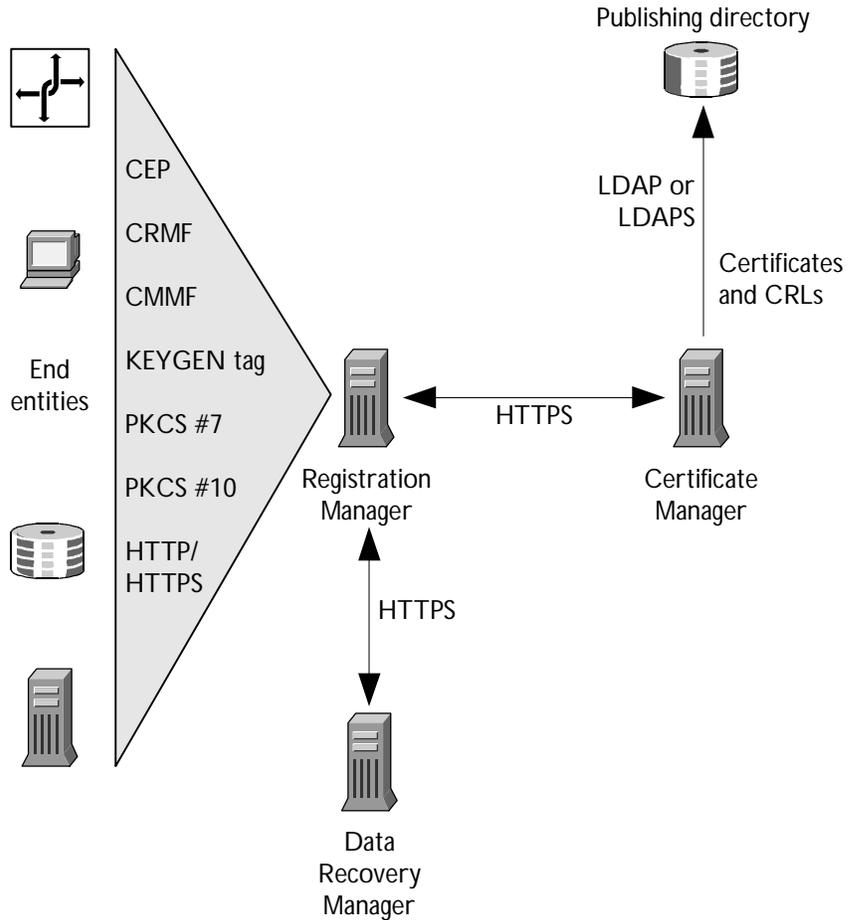
The three CMS subsystems can be deployed in many different relationships. Figure 3.4 illustrates some of the issues involved in deploying all three subsystems by showing the relationships among a single Certificate Manager, a single Registration Manager, and a single Data Recovery Manager, each installed in a different CMS instance on a different machine.

The Registration Manager handles all end-entity interactions and communicates with the Certificate Manager and the Data Recovery Manager over HTTPS. The Registration Manager is configured to request the end entity's private encryption key (in encrypted form) and send it to the Data Recovery Manager during the enrollment process. Before the Registration Manager sends the certificate request to the Certificate Manager for processing, the Registration Manager must receive verification from the Data Recovery Manager that the private key has been received and stored and that it corresponds to the end entity's public key.

Both the Registration Manager and the Certificate Manager can be configured to enable or disable LDAP publishing or to publish to separate directories. However, the Certificate Manager has the complete record of issued certificates, so it is recommended that publishing tasks be performed by the Certificate Manager only, as shown in the figure.

Many other combinations are possible. For example, the Data Recovery Manager and the Certificate Manager might be in the same instance; there might be multiple Registration Managers in different instances, all dealing with the same Data Recovery Manager and Certificate Manager; or the Certificate Manager might also handle some end-entity interactions. It's also possible to set up both Certificate Managers and Registration Managers such that each has a hierarchy of subordinate managers.

Figure 3.4 Certificate Manager, Registration Manager, and Data Recovery Manager in separate instances



Note The current design of Netscape Certificate Management System assumes that most deployments will rely on a single Data Recovery Manager (associated with either a Registration Manager or a Certificate Manager). However, it is also possible to write custom policies that support multiple Data Recovery Managers. This might be useful, for example, for subordinate CAs that issue certificates for completely independent organizations.

You can choose to install either a Certificate Manager and Data Recovery Manager or a Registration Manager and Data Recovery Manager in a single instance. There is not need to install a Certificate Manager and Registration Manager in the same instance; instead, a single Certificate Manager can be configured to perform all Registration Manager functions.

When subsystems are installed in the same instance, the connections between them are internal. Both subsystems must share the same host name, and the overall number of SSL server certificates can be reduced (see “Subsystem Certificate Decisions” on page 134).

Cloned Certificate Manager

A cloned Certificate Manager is a CMS server instance that uses the same CA signing key and certificate as another Certificate Manager. Each Certificate Manager issues certificates with serial numbers in a restricted range so that all of the servers together act as a single Certificate Authority (operating in several server processes).

Cloning requires somewhat more management and administrative effort and it creates more potential areas where the CA could become compromised, so it should only be used when absolutely necessary.

The advantage of cloning is the ability to distribute the Certificate Manager’s load across several processes or even several physical machines. For a CA that has high enrollment demand, the distribution gained from cloning allows more certificates to be signed and issued in a given time interval.

To create a cloned Certificate Manager, you must first install and configure at least one Certificate Manager and specify a definite upper and lower bound for the serial numbers it will use. You then install or create a new instance of a Certificate Manager (but do not configure it). Before configuring the clone, you copy the certificate and key database files from the original Certificate Manager to the new Certificate Manager’s `config` directory. If these databases are present, the Configuration Wizard will recognize that you are creating a clone and confirm that you want to reuse the CA’s signing key and certificate (if the clone is on the same server, you can also reuse the SSL server certificate).

If you store the CA key material on a hardware token, you will have to follow the hardware vendor’s instructions for copying the key material to a hardware device accessible to the clone.

A cloned Certificate Manager will have all the same features, agent gateway functions, and end entity gateway functions that a normal Certificate Manager has. You can then configure Registration Managers that point to different Certificate Manager servers but that appear to be serviced by the same CA.

Certificate Authority Decisions

This section covers some of the critical decisions you need to make about your certificate authority:

- CA's Distinguished Name (page 128)
- CA Signing Key Type and Length (page 129)
- CA Signing Certificate's Validity Period (page 129)
- Self-Signed Root Versus Subordinate CA (page 130)
- CAs and Certificate Extensions (page 130)
- CA Certificate Renewal or Reissuance (page 131)

CA's Distinguished Name

The core elements of a CA consist of a signing unit and the Certificate Manager's own identity. The signing unit digitally signs certificates requested by end entities that use a specified enrollment process to establish their identities. Regardless of how related Registration Managers or Data Recovery Managers are configured, any Certificate Manager must have its own distinguished name (DN), which is listed in every certificate it issues.

Like any other X.509 version 3 certificate, a CA certificate binds a DN to a public key. A DN is a series of name-value pairs that in combination uniquely identify an entity. For example, the following DN might be used to identify a hypothetical Certificate Manager for MyCorp:

```
cn=MyCA, o=MyCorp., ou=engineering, c=US
```

Many combinations of name-value pairs are possible for the Certificate Manager's DN. The DN must be unique and readily identifiable, since any end entity can examine it. For more information about DNs, see *Managing Servers with Netscape Console*.

CA Signing Key Type and Length

If you wish, you can import the signing key and certificate used in a Certificate Server 1.x installation rather than generating a new signing key pair. For information on how to do this, see Appendix A, “Migrating from Certificate Server 1.x.”

If you decide to generate a new signing key, one of the first decisions you need to make is whether to use the RSA or DSA algorithm. If you use DSA, the software can generate and verify the PQG value. PQG values are used to create the DSA signing key pair. For more information about the way they are used, see <http://www.itl.nist.gov/div897/pubs/fip186.htm>.

In general, longer keys are considered to be cryptographically stronger than shorter keys. However, longer keys also require more time for signing operations. (Certificate Manager CA signing keys up to 4096 bits in length are not subject to export restrictions.)

Many people no longer consider an RSA key length of 512 bits to be cryptographically strong. Export and other regulations permitting, it may be a good rule of thumb to start with 1024 bits and consider increasing the length to 2048 bits for certificates that provide access to highly sensitive data or services. However, the question of key length has no simple answers. Every organization must make its own decision based on its own security requirements. For more information on key length and encryption strength, see Appendix D of *Managing Servers with Netscape Console*.

CA Signing Certificate's Validity Period

Every certificate, including a Certificate Manager signing certificate, must have a validity period. Certificate Management System does not restrict the validity period you can specify. In general it's a good idea to specify as long a validity period as possible, depending on your plans for certificate renewal, the place of the CA in the certificate hierarchy, and the requirements of any public CAs that you may want to include in your PKI.

Self-Signed Root Versus Subordinate CA

For the purposes of an initial pilot, it is easiest to make the CA a self-signed root, so that you won't need to apply to a third party and wait for the certificate to be issued. Before deploying a full-blown PKI, however, you will need to consider this question carefully.

If you want your CA to chain up to a third-party public CA, you must carefully consider the restrictions that public CAs place on the kinds of certificates your CA can issue and the nature of the certificate chain. For example, a CA that chains up to a third-party CA might be restricted to issuing only Secure Multipurpose Internet Mail Extensions (S/MIME) and SSL client authentication certificates—not SSL server certificates. In addition, a CA that chains up to a third-party CA might not be allowed to have any subordinate CAs and might have to obey certain restrictions on its use of certificate extensions. These and other restrictions may be acceptable for some PKI deployments but not for others.

One benefit of chaining up to a public CA is that the third party is responsible for getting the root CA certificate into the browser or other end-entity software. This can be a major advantage if you are deploying an extranet that involves certificates used by different companies whose browsers you cannot control. Alternatively, if you create your own CA hierarchy from scratch, you are responsible for getting your root certificate into all the browsers used with the certificates you issue. You can accomplish this task within an intranet by using tools such as Mission Control Desktop or with the aid of Personal Security Manager, but extranet deployments can be more complicated.

CAs and Certificate Extensions

An X.509 v3 certificate contains an extensions field that permits any number of additional fields to be added to the certificate. Certificate extensions provide a way of adding information such as alternative subject names, policy information, and usage restrictions to certificates. The X.509 v3 standard defines a number of extensions for various purposes. Certificate Management System provides policy modules that you can use to implement many of the standard extensions.

Before the X.509 v3 standard was finalized, Netscape and other companies had to address certain issues, such as usage restrictions, with their own extension definitions. Therefore, to maintain compatibility with older versions of browsers that were released before the X.509 v3 specification was finalized, certain kinds of certificates should include some of the Netscape extensions. Certificate Management System provides policy modules that you can use to implement essential Netscape extensions.

The Internet Engineering Task Force (IETF), which controls many of the standards that underlie the Internet, is currently developing public-key infrastructure X.509 (PKIX) standards. These proposed standards further refine the X.509 v3 approach to extensions for use on the Internet. PKIX working group recommendations should also be taken into account when planning extensions for CA certificates, subordinate CA certificates, and end-entity certificates.

For more detailed information about extensions and recommendations for specific types of certificates, see Appendix B, "Certificate Extensions."

CA Certificate Renewal or Reissuance

When a CA signing certificate expires, all certificates signed with the CA's corresponding signing key become invalid. End entities use information in the CA certificate to verify the certificate's authenticity. If the CA certificate itself has expired, applications cannot chain the certificate to a trusted CA.

There are two ways of dealing with CA certificate expiration:

- **Renewing a CA certificate** involves issuing a new CA certificate with the same name and public and private key material as the old CA certificate, but with an extended validity period. As long as the new CA certificate is distributed to all users well before the old CA certificate expires, this approach allows certificates issued under the old CA certificate to continue working for the full duration of their validity periods. However, because of potential conflicts between the old CA certificate and the new CA certificate, this approach requires special care with early versions of Communicator 4.x.
- **Reissuing a CA certificate** involves issuing a new CA certificate with a new name, public and private key material, and validity period. This approach avoids some of the problems associated with renewing a CA

certificate, but it requires more work for both administrators and users to implement. All certificates issued by the old CA, including those that have not yet expired, must be renewed by the new CA.

There are advantages and disadvantages to each approach. Correct use of extensions, for example the `authorityKeyIdentifier` extension, can also affect the transition from an old CA certificate to a new one. You should begin planning for CA renewal or reissuance before you install any CMS managers; consider any ramifications your planned procedures may have for extensions, policies, and other aspects of your initial PKI deployment.

For a discussion of CA certificate expiration issues in the context of Certificate Server 1.x, see <http://help.netscape.com/products/server/certificate/cacertdoc/>. Many of the same issues apply to Certificate Management System.

For detailed information on certificate extensions, see Appendix B, “Certificate Extensions” on page 247.

Cryptographic Token Decisions

As explained in “PKCS #11” on page 72, one or more PKCS #11 modules must be available to any CMS subsystem instance. A PKCS #11 module, which can be implemented in either software or hardware, manages cryptographic services such as encryption and decryption. Netscape provides a built-in PKCS #11 module with Certificate Management System.

A PKCS #11 module always has one or more slots, which can be implemented as physical hardware slots in some form of physical reader (for example, for smart cards) or as conceptual slots in software. Each slot for a PKCS #11 module can in turn contain a token, which is the hardware or software device that actually provides cryptographic services and optionally stores certificates and keys.

As shown in Figure 1.12, the built-in PKCS #11 module for Certificate Management System includes two tokens, one for cryptographic operations and one for manipulating the key and certificate databases. You can accelerate cryptographic operations such as the signing of new certificates by using third-party hardware tokens and accelerator boards. CMS support for PKCS #11 also allows you to store critical keys, such as the root CA signing key, on smart cards or other hardware tokens to facilitate strong physical security measures.

Hardware products compatible with Certificate Management System are available from nCipher (<http://www.ncipher.com>) and Chrysalis-ITS (<http://www.chrysalis-its.com>).

If you decide to test or deploy hardware acceleration and storage devices, consult the vendor's installation instructions.

Publishing Decisions

Any Certificate Manager or Registration Manager that publishes certificates to a directory must specify the host name and port for the directory and indicate whether communication should take place over SSL. It must also specify how the subsystem should identify itself to the directory—by using password-based authentication or SSL client authentication. Finally, the directory itself must be configured (typically by the directory administrator) to authenticate the subsystem in the specified manner.

Although it's possible to configure the Registration Manager to publish certificates, the Certificate Manager has the complete record of issued certificates, so it is recommended that publishing tasks be performed by the Certificate Manager only. If it's necessary for some entries in a directory to be available outside the firewall, Netscape recommends using the partial replication feature of Directory Server to replicate the relevant portion of the directory to which the Certificate Manager publishes.

This guide assumes that you have already deployed an LDAP-compliant directory server (LDAP 2.0 or higher) for your enterprise; it does not cover directory planning and configuration. For information on Netscape Directory Server deployment, see the documentation that comes with that product.

Configuration of the publishing or corporate directory should take place before you install any Certificate Management System subsystems. Configuration details that the directory administrator may need to take care of include the following:

- If the authentication mechanism uses a DN (identifying the directory subtree in which the subsystem can publish certificates) and password, the directory administrator needs to set up a corresponding access control list (ACL).

- If authentication is based on SSL client authentication, the directory administrator needs to create an entry in the directory's `certmap.conf` file. The `certmap.conf` entry maps the DN in the subsystem's client certificate to a directory entry that specifies write permission to the appropriate portion of the directory tree.
- If you intend to publish certificates to the directory, the directory administrator needs to have an entry for each user to whom you intend to issue a certificate, and the directory schema must include a location to which the certificate should be published. If you want to publish the CA certificate, you will also need an entry for the CA.

Certificate Managers can also publish CRLs to the directory; this also requires an entry for the CA.

If you intend to use SSL authentication, both the directory and the Certificate Manager or Registration Manager must be configured appropriately for SSL.

For detailed information on LDAP publishing, see Part 7, "LDAP Publishing," in *Netscape Certificate Management System Administrator's Guide*.

Subsystem Certificate Decisions

Using a self-signed signing certificate for the Certificate Manager simplifies the deployment of an initial pilot. You can install the Certificate Manager without having to apply to a public certificate authority and waiting for it to issue, sign, and return your CA signing certificate. Your own Certificate Manager can then issue all the other certificates required for your pilot. However, taking this approach means that end entities outside your organization will not recognize your Certificate Manager unless you distribute the root Certificate Manager certificate to them.

The certificates and keys you need for each subsystem depend in part on whether the subsystems are in the same or different CMS instances. Subsystems installed together in the same instance use internal connectors to communicate and therefore don't need separate SSL certificates to authenticate each other.

When two CMS subsystems are installed in a single instance, they normally share a single SSL server certificate. If one or more subsystems are installed in a separate instance from the other subsystems, each instance requires a separate SSL server certificate.

In addition to any SSL server certificates, the Certificate Manager and the Registration Manager each requires its own signing certificate, and the Data Recovery Manager needs its own transport certificate and storage key.

SSL Server Certificates

Each CMS instance requires a single SSL server certificate. If you install two managers in the same instance—that is, a Certificate Manager or Registration Manager and a Data Recovery Manager—both managers share the same SSL server certificate.

Certificate Manager Certificates

Every Certificate Manager must have a CA signing certificate whose public key corresponds to the private key the Certificate Manager uses to sign the certificates it issues. This certificate is also used for SSL client authentication to the publishing directory (LDAP over SSL) if the Certificate Manager is set up to publish certificates or CRLs.

If the Certificate Manager is acting as a root CA, the CA certificate must be installed and trusted by each client that needs to validate certificates issued by the root Certificate Manager. In the context of a PKI, *trust* refers to the relationship between the user of a certificate and the CA that issued the certificate. If you trust a CA, you can generally trust valid certificates issued by that CA. It's possible to control which CAs the client or server software trusts and which it doesn't, and for what kinds of certificates, by means of settings within the software.

The Certificate Manager also requires an SSL server certificate. The Certificate Manager's SSL server certificate (or certificates) can be unique to the Certificate Manager or, if a Data Recovery Manager is installed in the same instance, shared with it.

Registration Manager Certificates

Every Registration Manager subsystem must have a signing certificate whose public key corresponds to the private key the Registration Manager uses to sign end-entity certificate requests before sending them to the Certificate Manager. Signed requests give the Certificate Manager persistent proof that a particular Registration Manager processed the request. If the Registration Manager is set up to publish certificates or CRLs, its signing certificate is also used for SSL client authentication to the publishing directory (LDAP over SSL).

The Registration Manager also requires at least one SSL server certificate. The Registration Manager's SSL server certificate (or certificates) can be unique to the Registration Manager or, if a Data Recovery Manager is installed in the same instance, shared with it.

Data Recovery Manager Certificate and Storage Key

The Data Recovery Manager needs a transport certificate and a storage key:

- The Data Recovery Manager transport certificate has a public key used by end-entity software to encrypt the private encryption key belonging to an end entity so that it can be sent (via the Registration Manager) to the Data Recovery Manager. The public key also corresponds to the private key used by the Data Recovery Manager to sign the proof-of-archival token it sends to the Registration Manager after storing an end entity's encryption key.
- The Data Recovery Manager storage key is used by the Data Recovery Manager to encrypt the end entity's encryption key (after it has been decrypted with the Data Recovery Manager's private transport key) before the Data Recovery Manager stores the encryption key in the local directory. Data encrypted with the storage key can be retrieved only if m of n "split keys" are provided at the same time by m of n authorized agents.

The Data Recovery Manager also requires at least one SSL server certificate. The Data Recovery Manager's SSL server certificate (or certificates) can be unique to the Data Recovery Manager or, if another subsystem are located in the same instance, shared with that subsystem.

Authentication Decisions

The role of authentication modules in certificate enrollment is discussed in “Authentication and Policy Modules” on page 29. CMS managers use authentication modules to verify the identity of a user requesting a service, such as certificate enrollment. For example, a user can be prompted to provide a name and password, and the authentication module can check a directory entry to confirm that they are correct.

Authentication is one of the essential functions of Certificate Management System. The main purpose of a certificate is to provide a trustworthy association between the public key of the subject and the subject’s name and other attributes. Therefore the manner in which administrators, agents, and end entities are authenticated, especially for operations related to certificate enrollment, requires careful planning and control throughout the lifetime of a PKI deployment.

For examples of some different approaches to authentication during certificate enrollment, see “Some Enrollment Scenarios” on page 33.

For a detailed overview of authentication management using Certificate Management System, see Chapter 9, “Introduction to Authentication,” in *Netscape Certificate Management System Administrator’s Guide*.

Policy Decisions

The role of policies in certificate enrollment is discussed in “Authentication and Policy Modules” on page 29. CMS managers use policies to evaluate or verify incoming certificate enrollment or management requests from end entities and to determine the outcome. For example, in the case of certificate enrollment request, the outcome is the issued certificate.

Decisions regarding policies depend on both the subsystem involved and your overall topology. Whether your CA signing certificate is self-signed or not, it represents part of a certificate hierarchy. For example, a CA may be a root CA for subordinate CAs that issue certificates to different parts of a large organization, or it may be one of the subordinate CAs that chain up to an internal root CA, or it may be a linked CA that chains up to a third party.

Policies configured for a Certificate Manager apply to all certificates issued by that Certificate Manager or its subordinates. Policies configured for a Registration Manager subsystem are local to the Registration Manager. This distinction can be used to model the levels of authority within an organization. Enrollment can be fully automated by means of custom policy and authentication subsystems at the Registration Manager level.

Thus, a policy for a Certificate Manager might be that all subject names have to end with `o=Netscape`. Registration Managers for individual departments can enforce this policy and can also define their own, local naming policies, such as `ou=engineering`.

Another variation is to have the Certificate Manager enforce the companywide policies and have subordinate Certificate Managers, instead of Registration Managers, enforce the names for individual departments. Each subordinate Certificate Manager, in turn, can delegate enrollment responsibilities to multiple Registration Managers, which can be configured to apply the policies uniformly in different geographic locations.

For a detailed discussion of policy management, see Chapter 16, "Introduction to Policy," in *Netscape Certificate Management System Administrator's Guide*.

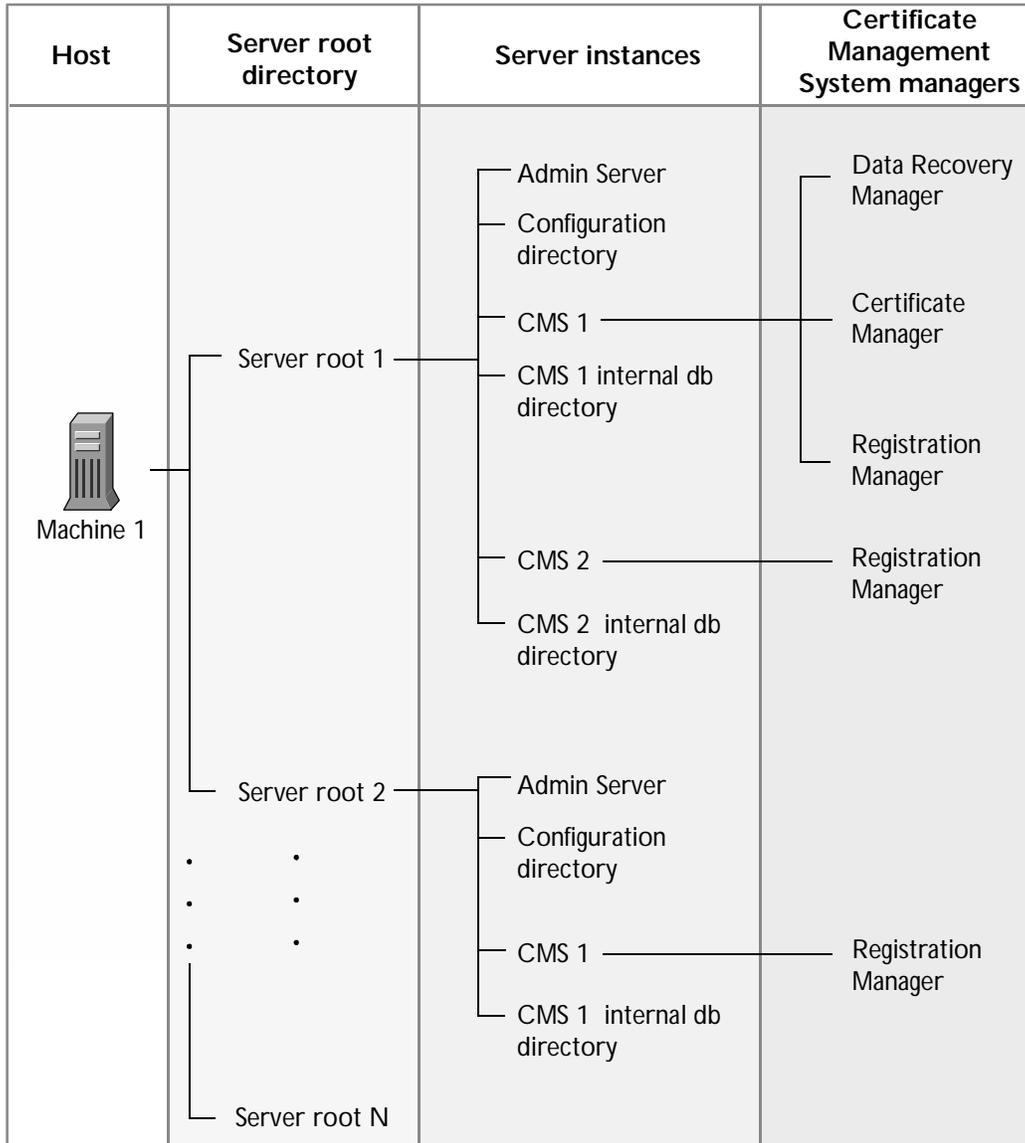
Deployment Strategy and Port Assignments

Before you install any CMS instance, you should review the decisions described in this chapter and work out the relationships between the Certificate Managers, Data Recovery Managers, and Registration Managers you want to deploy for your organization. Once you have decided which subsystems to install and where, fill out a copy of the worksheet provided in Chapter 4, "Installation Worksheet" for each separate installation.

You can create multiple instances of Certificate Management System in a single server root directory, each containing either one or two CMS managers. If you want to install CMS managers on different hosts, you must run the entire installation on each host, specifying the services for each instance of Certificate Management System. You can also perform additional complete installations on the same host in a different server root directory.

Figure 3.5 shows an example of how several CMS instances can be installed on a single host machine. (Note that on Windows NT, you can install a single server root only; multiple server roots are not permitted.)

Figure 3.5 Deploying servers on a single host



Each server root directory shown in Figure 3.5 has its own Administration Server and Netscape Console and access to a configuration directory. Each CMS instance has a corresponding instance of Directory Server that functions as the

internal database for that CMS instance. Each server root directory can have one or more instances of Certificate Management System, each with its own set of one or two subsystems and its own corresponding internal database.

Figure 2.1 on page 82 illustrates the ports used by a single Certificate Management System instance. You can also install multiple instances on a single machine, either in the same server root or as completely separate installations in separate server roots.

When you install additional CMS instances on a machine with a single IP address, you are required to specify a different set of ports for each CMS instance to listen on. That is, each CMS instance will at least four unique ports:

- Internal database port for communication with internal database
- SSL administration port for communication with Netscape Console
- SSL agent port for communication with designated agents
- At least one of these ports:
 - SSL user port for communication with end entities
 - Non-SSL user port for communication with end entities

The ports shown above are required for each CMS instance. Each server root needs two additional unique ports: one for the configuration directory and one for the administration server.

When you install additional CMS instances on a machine that has been set up with more than one IP address, you can configure each instance to listen to a specific IP address. If each instance has a different IP address, you can use the same port numbers for additional CMS instances installed on the same machine—that is, you can use one set of four or five ports for all the instances.

For more information about installing multiple CMS instances, see “Chapter 4, Installing and Uninstalling CMS Instances,” in *Netscape Certificate Management System Administrator's Guide*.

Installation Worksheet

This chapter provides a worksheet to help you prepare for installing a single instance of Netscape Certificate Management System.

Print this chapter and make as many copies as you need. Fill out one copy for each CMS instance you plan to install and refer to it during the installation and configuration process. You should fill it in after you have read Chapter 3, “Planning Your Deployment.” It is designed for easy reference while you are following the procedures described in Chapter 5, “Installation and Configuration.”

Warning Each completed worksheet contains sensitive information, such as passwords, that could severely compromise the security of your entire PKI if it falls into the wrong hands. Be sure to keep completed worksheets physically protected.

This chapter has the following sections:

- Information for Unix Installation Script (page 142)
- Information for NT Installation Script (page 145)
- Initial Configuration (page 149)
- Certificate Manager Configuration (page 151)
- Registration Manager Configuration (page 157)
- Data Recovery Manager Configuration (page 159)
- Cloned Certificate Manager Configuration
- SSL Server Certificate Configuration (page 167)
- Single Sign-On Password (page 171)

Information for Unix Installation Script

The information summarized here must be provided once for each server root installation on a Unix system.

Installation Location

To install an instance of Certificate Management System, you must also install an Administration Server and Netscape Console application and have access to a configuration and user/group directory. For more information on the Netscape server environment, see *Managing Servers with Netscape Console*.

- Installation directory
(Server root directory) _____
Enter the full pathname for the existing server root directory or for a new server root directory. For example, `/usr/netscape/server4`.
- Computer name _____
The default should be the fully qualified host name of the machine on which the installation is taking place. For example, `mydirectory.com`. Do not attempt to install remotely.

Configuration Directory Server

- System user ID _____
Enter the user ID that Directory Server will run as. The configuration directory server process runs as this user. You should run the server as a user with restricted access to other system files and resources. Where your system supports it, accept the default user `nobody`, creating that user as necessary.
- System group _____
Enter a group to which the System User ID belongs. The group should also have limited access to system resources and files. Where your system supports it, accept the default user `nobody`, creating that group as necessary.

Do you want to register this software with an existing Netscape configuration directory server?

- Yes or No. _____

If you choose No, the Installation Wizard will create a new instance of Directory Server for use as the configuration directory for this server root.

If you choose Yes, you must also supply the following information about the existing configuration directory:

- Computer name _____

The default should be the fully qualified host name of the machine on which the configuration directory is located. For example, mydirectory.com.

User/Group Directory Server

Do you want to use another directory to store your data?

- Yes or No. _____

If you choose No, the installation script either adds a user/group directory to the newly installed instance of Directory Server (if you answered no to the preceding question) or installs a new instance of Directory Server for use as a user/group directory.

If you choose Yes, you must also supply the following information:

- User directory host name _____
- User directory port _____
- Bind as _____
- User directory server suffix _____
- User directory administrator ID _____

Configuration Directory Settings

You need to provide the following information about the configuration directory, whether it is an existing one or a new one to be created by the Installation Wizard:

- Directory Server network port _____
Enter the port number for the Directory Server instance. The default is 389, if it is available, or a randomly selected number. The port number you specify must not be used for any other purpose.
- Directory Server identifier _____
This unique identifier is required for each instance of a Directory Server. For example, `configdir`.
- Configuration Directory Server Administrator ID _____
The ID for the user who will authenticate to Netscape Console with full privileges. For example, `diradmin1`.
- Configuration Directory Server Administrator Password _____
The password must be at least eight characters long.
- Suffix _____
Enter the domain name of the current host. For example, `o=mydomain.com`.
- Directory Manager DN _____
Enter the distinguished name (DN) of the directory manager for the configuration directory.
This DN can be short and does not need to conform to any suffix configured for your directory. It also should not correspond to an actual entry stored in your directory. For example, `cn=Directory Manager`.
- Directory Manager password _____
The password must be at least eight characters long.
- Administration domain _____
This domain name identifies the collection of servers that use the same configuration directory. For example, `mydomain.com`

Administration Server Information

- Administration Port _____
The default Administration Port is randomly generated. Pick a port number between 1024 and 65535 on which to run your Administration Server, or accept the default number.
- Run Administration Server as _____
Run the Administration Server as `root` if you want to be able to start and stop services and use port numbers below 1024 (for example to use port 80 for the HTTP end entity gateway).

Certificate Management System Identifier

You must specify a unique identifier for the CMS server instance that you are installing.

- Certificate Management System server identifier _____
Enter a unique identifier such as `certxx01`.

Information for NT Installation Script

The information summarized here must be provided once for each server root installation.

Installation Directory

To install an instance of Certificate Management System, you must also install an Administration Server and Netscape Console application and have access to a configuration and user/group directory. For more information on the Netscape server environment, see *Managing Servers with Netscape Console*.

- Installation directory
(Server root directory) _____

The default installation directory is C:\Netscape\Server4. If you want to use a different directory, enter the full pathname for the existing server root directory or for a new server root directory.

You cannot install more than one server root directory on a Windows NT system.

Configuration Directory Server

Choose one of these options:

- This instance will be the configuration directory server. _____
If you choose the above option, the Installation Wizard will create a new instance of Directory Server for use as the configuration directory for this server root.
- Use existing configuration directory server. _____

If you choose to use an existing configuration directory, you must supply the following information:

- Host name _____
- Port _____
- Bind as _____
- Password _____

User/Group Directory Server

Choose one of these options:

- Store data in this directory server. _____
If you choose this option, the installation script either adds a user/group directory to the newly installed instance of Directory Server (if you have already decided to install a new configuration directory) or installs a new instance of Directory Server for use as a user/group directory.
- Store data in an existing directory server _____

If you choose to use an existing directory, you must supply the following information:

- Host name _____
- Port _____
- Bind as _____
- Password _____
- Suffix _____

Configuration Directory Settings

You need to provide the following information about the configuration directory, whether it is an existing one or a new one to be created by the Installation Wizard:

- Directory Server identifier _____
This unique identifier is required for each instance of a Directory Server. For example, `configdir`.
- Directory Server network port (default is 389) _____
Enter the port number for the Directory Server instance.
- Suffix _____
If you are creating a new directory, this should be the domain name of the current host. For example, `o=mydomain.com`.

Configuration Directory Server Administrator

- Configuration Directory Server Administrator ID _____
For example, `diradmin1`.
- Configuration Directory Server Administrator Password _____

The password must be at least eight characters long.

Directory Server Administration Domain

- Administration domain _____
This domain name identifies the collection of servers that use the same configuration directory. For example, `mydomain.com`

Directory Manager Settings

- Directory manager DN _____
Enter the distinguished name (DN) of the directory manager for the configuration directory.
This DN can be short and does not need to conform to any suffix configured for your directory. It also should not correspond to an actual entry stored in your directory. For example, `cn=Directory Manager`.
- Directory Manager password _____
The password must be at least eight characters in length.

Administration Server Port

- Administration Port _____
Pick a port number between 1024 and 65535 on which to run your Administration Server, or accept the default number.

Certificate Management System Identifier

You must specify a unique identifier for the CMS server instance that you are installing.

- Certificate Management System server identifier _____
Enter a unique identifier such as `certxx01`.

Initial Configuration

For each instance of Certificate Management System that you create, you use the Installation Wizard to supply information about that instance's configuration. The information described in this section is required for each CMS instance, regardless of which subsystems you decide to install.

Internal Database

For each instance of Certificate Management System, a new instance of Netscape Directory Server is created on the local host to act as the internal (local) database. Each subsystem must have access to this local database to store certificates, certificate requests, keys, and other information. The Certificate Management System uses LDAP over SSL to communicate with its local database.

- Certificate Management System internal database instance ID _____
The default provided by the system is the CMS server identifier with the suffix `-db`; for example, `cmsdemo-db`.
- Port number _____
The default is 38900, but you may choose any value less than 65535. On Unix, you must choose a port greater than 1024 if you are not logged in as root.
- Directory Manager DN _____
The default is `CN=Directory Manager`. You can enter something more meaningful, such as `CN=Internal Directory Manager`.
- Internal database password _____

Administrator

Specify the CMS administrator. This person will be able to access the CMS window of Netscape Console and approve the first agent certificate.

- CMS Administrator ID _____
For example, CMSadmin.
- CMS Administrator full name _____
For example, Certificate Management System Administrator.
- CMS Administrator password _____

Subsystems

Choose the subsystems you will install in this instance. You can choose Certificate Manager and Data Recovery Manager together, or Data Recovery and Registration Manager together, or you can choose any individual manager, but you cannot install Certificate Manager and Registration Manager together. The Certificate Manager can be configured to perform all Registration Manager functions, so it's not necessary or possible to install both managers in the same instance.

- Certificate Manager _____
- Registration Manager _____
- Data Recovery Manager _____

Remote Certificate Manager

If you are installing a Registration Manager, you need to provide the following information about the Certificate Manager to which the Registration Manager sends certificate requests:

- Host name for remote Certificate Manager _____
- SSL agent port for remote Certificate Manager _____

Remote Data Recovery Manager

If you are installing a standalone Certificate Manager or Registration Manager, and if you have already installed a remote Data Recovery Manager that you want the new manager to use, you need to provide the following information about the Data Recovery Manager:

- Host name for remote Data Recovery Manager_____
- SSL agent port for remote Data Recovery Manager_____

Network Configuration

Enter numbers for the ports to be used for various kinds of communications. On Unix, you must be `root` to assign ports less than 1024. The default values are well-known ports, which are used only if they are not already in use. If these defaults are not available, a randomly chosen port number is given as the default.

For a discussion of port assignments, see “Deployment Strategy and Port Assignments” on page 138.

- SSL administration port (HTTPS) (default is 8200)_____
- SSL agent port (HTTPS) (default is 8100)_____
- SSL end-entity port (HTTPS) (default 443)_____
- Non-SSL end-entity port (HTTP) (default 80)_____.

Certificate Manager Configuration

This section summarizes information required to configure a Certificate Manager as a root or subordinate CA (either by itself or as part of a joint installation with a Data Recovery Manager).

Server Migration from Certificate Server 1.x

If you are importing any certificates and keys previously created with Certificate Server 1.x, you must specify where they are, how to retrieve them, and where to put them. For information about migrating these files to Certificate Management System, see Appendix A, “Migrating from Certificate Server 1.x.”

Migration Tool Output Files

- Pathname to migration output files _____
Enter the pathname to the directory where the migrate tool output files `keyscerts.dat`, `database_add.ldif`, and `database_mod.ldif` are located. All three files must be in the same directory. For example, `/eng/migrationfile/certmanage/mycompany/`.
- Password used to create `keyscerts.dat` (“transport password”) _____
Enter the transport password that you specified while exporting data with the Migration tool.

Token for CA Signing Certificate

- Token for storing the Certificate Manager CA signing certificate and private key _____
Enter either `internal` (if you plan to use the internal token) or the name of an external token. If you are using an external token, you will need to install it before you run the Installation Wizard. In the Installation Wizard, you can select from a list of all installed and available tokens. For example, `SmartCard`.
- Token password _____
The password for the token must be at least one character long.

Token for SSL Server Certificate

- Token for storing the SSL server certificate _____
Enter either `internal` (if you plan to use the internal token) or the name of an external token. If you are using an external token, you will need to install it before you run the Installation Wizard. In the Installation Wizard, you can select from a list of all installed and available tokens. For example, `SmartCard`.
- Token password _____
The password must be at least one character long.

CA Signing Certificate

When you install the Certificate Manager subsystem, you must supply information for the CA certificate that the Certificate Manager will use to sign the certificates it issues. This certificate also functions as the Certificate Manager's SSL client certificate.

CA's Serial Number Range

For most CAs, you only need to enter the starting serial number. When you configure cloned CAs, you must specify upper and lower bounds for the serial numbers on all CAs and you must make sure the ranges do not overlap.

- CA's starting serial number _____
Enter the lowest serial number available for this CA to assign to certificates it creates. You can enter the number in decimal or hexadecimal (0xnn). The default is 0x1.
- CA's ending serial number _____
Enter the highest serial number available for this CA. You can enter the number in decimal or hexadecimal (0xnn). The default is no upper limit (blank).

Key-Pair Information for CA Signing Certificate

For a discussion of related issues, see “CA Signing Key Type and Length” on page 129.

- Token for storing the Certificate Manager CA signing certificate and private key _____
Enter either `internal` (if you plan to use the internal token) or the name of an external token. If you are using an external token, you will need to install it before you run the Installation Wizard. In the Installation Wizard, you can select from a list of all installed and available tokens. For example, `SmartCard`.
- Token password _____
The password for the token must be at least one character long.
- Key type _____
RSA or DSA.
- Key length _____
Available settings for RSA are 512, 1024, 2048, or custom. Available settings for DSA are 512, 1024, or custom.
In general, longer keys are considered to be cryptographically stronger than shorter keys. However, longer keys also require more time for signing operations.
- Message Digest Algorithm (select one): SHA1___ MD2___ MD5___
Select the message digest algorithm to use for generating digital signatures on certificates.

Subject Name for CA Signing Certificate

For a discussion of issues related to the subject name, see “CA’s Distinguished Name” on page 128.

You may fill in the attribute template or simply enter the DN as a string of attribute-value pairs.

- Common Name (CN=) _____
- Organizational Unit (OU=) _____

- Organization (O=) _____
- Locality (L=) _____
- State (ST=) _____
- Country (C=) _____

A DN is a series of name-value pairs that in combination uniquely identify an entity. The subject DN identifies the CA signing certificate. You are not required to enter all the values, but must enter the Organization (O), such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. For more information about distinguished names, see Appendix A, "Distinguished Names," in *Netscape Certificate Management System Administrator's Guide*.

Validity Period for CA Signing Certificate

You can specify the validity period for a self-signed CA signing certificate only. The validity period for a subordinate CA signing certificate is determined by the issuing CA.

- Validity period _____ to _____

Enter beginning and ending dates for the certificate's validity period. The validity period for the CA signing certificate determines how soon you will have to renew the certificate, which can be a complex procedure.

Extensions for CA Signing Certificate

You can specify the extensions for a self-signed CA signing certificate only. Extensions for a subordinate CA signing certificate are specified by the issuing CA.

The default settings should work for most deployments. If necessary, you can add an additional extension by pasting its base-64 encoding in the space provided on this screen. For more information about extensions, see Appendix B, "Certificate Extensions."

Confirm that you want to include the following extensions. Check off all that apply; defaults are indicated in parentheses.

- Basic constraints (Yes) _____
 - CA (Yes) _____

Certificate Manager Configuration

- Certification path length (Null)_____

The certificate chain path length, if specified, determines the maximum number of certificates in a chain, starting with the end-entity certificate. If you do not specify this attribute, the length of the chain is unlimited.

- Netscape certificate type (Yes)_____

- SSL client (No)_____
- Object-signing (No)_____
- SSL server (No)_____
- S/MIME CA (Yes)_____
- S/MIME (No)_____
- Object-signing CA (Yes)_____
- SSL CA (Yes)_____

- Authority Key Identifier (Yes) _____

- Subject Key Identifier (Yes) _____

- Key usage (No)_____

If you decide to include the key usage extension, the following key usage bits are set by default:

- digitalSignature
- keyCertSign
- CRLSign

- Additional Extension (No)_____

To add extensions not included by default by CMS, you will need to paste the base64 encoding of a sequence of extensions into the wizard.

CA Signing Certificate Request

If you are installing a subordinate CA, you need to specify where to send your request for a CA signing certificate.

If you are submitting your certificate request to a third-party CA, follow the instructions provided by that CA.

If you are submitting your certificate request to another Certificate Manager, you need to know its URL:

- End-entity URL for issuing Certificate Manager _____
Enter the URL for the end-entity gateway of the Certificate Manager that will issue the subordinate CA's signing certificate. For example, `https://hostname:443/`.

Registration Manager Configuration

This section summarizes information required to configure a Registration Manager (either by itself or as part of a joint installation with a Data Recovery Manager).

Registration Manager Signing Certificate Request

When you install a Registration Manager subsystem, you must supply information for the certificate that the Registration Manager will use to sign certificate requests. This certificate also functions as the Registration Manager's SSL client certificate. The Installation Wizard formulates a certificate request on the basis of information you provide. It is possible for the CA that issues the certificate to overrule some of your decisions.

Key-Pair Information for Registration Manager Signing Certificate

- Token for storing the Registration Manager CA signing certificate and private key _____
Enter either `internal` (if you plan to use the internal token) or the name of an external token. If you are using an external token, you will need to install it before you run the Installation Wizard. In the Installation Wizard, you can select from a list of all installed and available tokens. For example, `SmartCard`.
- Token password _____
The password for the token must be at least one character long.
- Key type _____
RSA or DSA.
- Key length _____
Available settings for RSA are 512, 1024, 2048, or custom. Available settings for DSA are 512, 1024, or custom.
In general, longer keys are considered to be cryptographically stronger than shorter keys. However, longer keys also require more time for signing operations.
- Message Digest Algorithm (select one): SHA1___ MD2___ MD5___
Select the message digest algorithm to use for generating digital signatures on certificates.

Subject Name for Registration Manager Signing Certificate

- Common Name (CN=) _____
- Organizational Unit (OU=) _____
- Organization (O=) _____
- Locality (L=) _____
- State (ST=) _____
- Country (C=) _____

A DN is a series of name-value pairs that in combination uniquely identify an entity. The subject DN identifies the Registration Manager signing certificate. You are not required to enter all the values, but must enter the Organization (O), such as your company name. The Organization is required because its absence causes Netscape Communicator 4.x to crash. For more information about distinguished names, see Appendix A, "Distinguished Names," in *Netscape Certificate Management System Administrator's Guide*.

Registration Manager Signing Certificate Issuer

If you are submitting your certificate request to a third-party CA, follow the instructions provided by that CA.

If you are submitting your certificate request to another Certificate Manager, you need to know its URL:

- End-entity URL for issuing Certificate Manager _____

Enter the URL for the end-entity gateway of the Certificate Manager that will issue the subordinate CA's signing certificate. For example, `http://hostname:17006`.

Data Recovery Manager Configuration

This section summarizes information required to configure a Data Recovery Manager (either by itself or as part of a joint installation with a Certificate Manager or Registration Manager).

Transport Certificate

Key-Pair Information for Transport Certificate

For a discussion of issues related to key type and length, see “CA Signing Key Type and Length” on page 129.

- Token for storing the transport certificate signing certificate and private key _____
Enter either `internal` (if you plan to use the internal token) or the name of an external token. If you are using an external token, you will need to install it before you run the Installation Wizard. In the Installation Wizard, you can select from a list of all installed and available tokens. For example, `SmartCard`.
- Token password _____
The password for the token must be at least one character long.
- Key type _____
RSA or DSA.
- Key length _____
Available settings for RSA are 512, 1024, 2048, or custom. Available settings for DSA are 512, 1024, or custom.
In general, longer keys are considered to be cryptographically stronger than shorter keys. However, longer keys also require more time for signing operations.
- Message Digest Algorithm (select one): SHA1___ MD2___ MD5___
Select the message digest algorithm to use for generating digital signatures on certificates.

Subject Name for Transport Certificate

- Common Name (CN=) _____
- Organizational Unit (OU=) _____
- Organization (O=) _____
- Locality (L=) _____

- State (ST=) _____
- Country (C=) _____

A DN is a series of name-value pairs that in combination uniquely identify an entity. The subject DN identifies the transport certificate. You are not required to enter all the values, but must enter the Organization (O), such as your company name. The Organization is required because its absence causes Netscape Communicator 4.x to crash. For more information about distinguished names, see Appendix A, "Distinguished Names," in *Netscape Certificate Management System Administrator's Guide*.

Validity Period for Transport Certificate

You can specify the validity period for a transport certificate only if you are installing the Certificate Manager and Data Recovery Manager at the same time and you want the Certificate Manager that you just installed issue the transport certificate. If the transport certificate is issued by a remote CA, its validity period is determined by the issuing CA.

- Validity period _____ to _____
Enter beginning and ending dates for the transport certificate's validity period.

Extensions for Transport Certificate

You can specify the extensions for a transport certificate only if you are installing the Certificate Manager and Data Recovery Manager at the same time and you have decided to have the Certificate Manager that you just installed issue the certificate. If the transport certificate is issued by a remote CA, its extensions are determined by the issuing CA.

The default settings should work for most deployments. If necessary, you can add an additional extension by pasting its base-64 encoding in the space provided on this screen. For more information about extensions, see Appendix B, "Certificate Extensions."

Confirm that you want to include the following extensions. Check off all that apply; defaults are indicated in parentheses.

- Basic constraints (No) _____
 - CA (No) _____

- Certification path length (No)_____

The certificate chain path length, if specified, determines the maximum number of certificates in a chain, starting with the end-entity certificate. If you do not specify this attribute, the length of the chain is unlimited.

- Netscape certificate type ((No)_____

- SSL client (No)_____
- Object-signing (No)_____
- SSL server (No)_____
- S/MIME CA ((No)_____
- S?MIME (No)_____
- Object-signing CA ((No)_____
- SSL CA ((No)_____

- Authority Key Identifier (Yes) _____

- Subject Key Identifier (No)

- Key usage (No)_____

If you decide to include the key usage extension, the `keyEncipherment` key usage bit is set by default.

- Additional Extension (No)_____

To add extensions not included by default by CMS, you will need to paste the base64 encoding of a sequence of extensions into the wizard.

Transport Certificate Request

If you are obtaining your transport certificate from a remote CA, you need to know where to submit your certificate request.

If you are submitting your transport certificate request to a third-party CA, follow the instructions provided by that CA.

If you are submitting your certificate request to a CMS Certificate Manager, you need to know its URL:

- End-entity URL for issuing Certificate Manager _____
Enter the URL for the end-entity gateway of the Certificate Manager that will issue the transport certificate. For example, `http://hostname:17006`.

Storage Key and Recovery Agent Configuration

Storage Key Creation

Specify the length of the key that the Data Recovery Manager uses to encrypt end-entity encryption keys for storage.

- Storage key length _____
The options available are 512, 1024, or 2048.

Data Recovery Scheme - 1

The number of agents you enter here is determined by your organization's policies with respect to data recovery. If you enter a larger number than the default of 2 for the number of recovery agents required to recover a key, you're reducing the chances of inappropriate recovery but increasing the complexity of the recovery process.

Decide how you want to set up your m of n data recovery scheme ($n > m$):

- Number of recovery agents required to recover a key (m , default 2)

- Total number of designated recovery agents (n , default 3) _____

Data Recovery Scheme - 2

Specify user IDs and passwords for the total number of designated recovery agents (see preceding section):

- User ID_____ Password_____

Cloned Certificate Manager Configuration

This section summarizes information required to configure a clone of a Certificate Manager. You must have installed the original Certificate Manager and installed or created a new CMS instance. You must copy the `key3.db` and `cert7.db` files from the `config` directory of the original server to the `config` directory of the cloned server. If you use a hardware token for key and certificate storage, you must copy any key or certificate data from the original token to a new token accessible to the cloned Certificate Manager.

You can clone a Certificate Manager instance to have two server processes performing the same CA functions using the same keys and certificates. Each cloned Certificate Manager, including the original, must only issue certificates with serial numbers that do not conflict with the serial numbers issued by other clones. Use the CA serial number range to make sure that the serial numbers used by a clone do not overlap with the serial number range of another clone (or the original server).

If the cloned Certificate Manager has the same hostname as the original server, the clone can use the same SSL server certificate. The SSL server certificate DN contains the hostname as the common name (CN) attribute, so a clone with a different hostname must enroll for a new SSL server certificate.

Server Migration from Certificate Server 1.x

If you are importing any certificates and keys previously created with Certificate Server 1.x, you must specify where they are, how to retrieve them, and where to put them. For information about migrating these files to Certificate Management System, see Appendix A, “Migrating from Certificate Server 1.x.”

Migration Tool Output Files

- Pathname to migration output files _____
Enter the pathname to the directory where the migrate tool output files `keyscerts.dat`, `database_add.ldif`, and `database_mod.ldif` are located. All three files must be in the same directory. For example, `/eng/migrationfile/certmanage/mycompany/`.
- Password used to create `keyscerts.dat` (“transport password”) _____
Enter the transport password that you specified while exporting data with the Migration tool.

Token for CA Signing Certificate

- Token for storing the Certificate Manager CA signing certificate and private key _____
Enter either `internal` (if you plan to use the internal token) or the name of an external token. If you are using an external token, you will need to install it before you run the Installation Wizard. In the Installation Wizard, you can select from a list of all installed and available tokens. For example, `SmartCard`.

- Token password _____
The password for the token must be at least one character long.

Token for SSL Server Certificate

- Token for storing the SSL server certificate _____
Enter either `internal` (if you plan to use the internal token) or the name of an external token. If you are using an external token, you will need to install it before you run the Installation Wizard. In the Installation Wizard, you can select from a list of all installed and available tokens. For example, `SmartCard`.
- Token password _____
The password must be at least one character long.

CA Signing Certificate

When you install the Certificate Manager subsystem, you must supply information for the CA certificate that the Certificate Manager will use to sign the certificates it issues. This certificate can also function as the Certificate Manager's SSL client certificate. If the clone uses a different hostname than the original CA, you will need to generate a new SSL server certificate.

CA's Serial Number Range

For most CAs, you only need to enter the starting serial number. When you configure cloned CAs, you must specify upper and lower bounds for the serial numbers on all CAs and you must make sure the ranges do not overlap.

- CA's starting serial number _____
Enter the lowest serial number available for this CA to assign to certificates it creates. You can enter the number in decimal or hexadecimal (`0xnn`). The default is `0x1`.
- CA's ending serial number _____

Enter the highest serial number available for this CA. You can enter the number in decimal or hexadecimal (0xnn). The default is no upper limit (blank).

Cloned Key and Certificate Material

If you do not use the copied key and certificate databases, the Certificate Manager will need to generate a new signing key and certificate; consequently, it will not be a clone.

- Use existing key and certificate? _____

Answer yes, otherwise you are creating a new Certificate Manager and not a clone.

- Instance name of the original server _____
- Token name where copied keys are stored _____
- Token password _____

SSL Server Key and Certificate

If the clone uses the same hostname, you can use the same SSL server certificate and key copied from the original server. Otherwise, answer no and continue with the next section, "SSL Server Certificate Configuration."

- Use existing SSL server key and certificate? Yes____ No_____
- Instance name of the original server _____
- Token name where copied keys are stored _____
- Token password _____

SSL Server Certificate Configuration

When you install an instance of Certificate Management System, you must supply information for the SSL server certificate used by that instance to identify itself. The same SSL certificate is shared by all subsystems installed in that instance.

SSL Server Certificate

Key-Pair Information for SSL Server Certificate

- Token for storing the SSL server certificate and private key _____
Enter either `internal` (if you plan to use the internal token) or the name of an external token. If you are using an external token, you will need to install it before you run the Installation Wizard. In the Installation Wizard, you can select from a list of all installed and available tokens. For example, `SmartCard`.
- Token password _____
The password for the token must be at least one character long.
- Key type _____
RSA or DSA.
- Key length _____
For domestic versions of Certificate Management System, available settings for RSA are 512, 1024, 2048, or custom, and available settings for DSA are 512, 1024, or custom.
- Message Digest Algorithm (select one): SHA1___ MD2___ MD5___
Select the message digest algorithm to use for generating digital signatures on certificates.

Subject Name for SSL Server Certificate

- Common Name (CN=) _____
- Organizational Unit (OU=) _____
- Organization (O=) _____
- Locality (L=) _____
- State (ST=) _____
- Country (C=) _____

A DN is a series of name-value pairs that in combination uniquely identify an entity. The subject DN identifies the CA signing certificate. You are not required to enter all the values, but must enter the Organization (O), such

as your company name. The Organization is required because its absence causes Netscape Communicator 4.x to crash. For more information about distinguished names, see Appendix A, "Distinguished Names," in *Netscape Certificate Management System Administrator's Guide*.

Validity Period for SSL Server Certificate

You can specify the validity period for an SSL server certificate only if you are installing a Certificate Manager and you have decided to have that Certificate Manager issue the certificate. If the SSL server certificate is issued by a remote CA, its validity period is determined by the issuing CA.

- Validity period _____
Enter beginning and ending dates for the certificate's validity period.

Extensions for SSL Server Certificate

You can specify the extensions for an SSL server certificate only if you are installing a Certificate Manager and you have decided to have that local Certificate Manager issue the certificate. If the SSL server certificate is issued by a remote CA, its extensions are determined by the issuing CA.

The default settings should work for most deployments. If necessary, you can add an additional extension by pasting its base-64 encoding in the space provided on this screen. For more information about extensions, see Appendix B, "Certificate Extensions."

Confirm that you want to include the following extensions. Check off all that apply; defaults are indicated in parentheses.

- Basic constraints (No) _____
 - CA (Nos) _____
 - Certification path length (No) _____

The certificate chain path length, if specified, determines the maximum number of certificates in a chain, starting with the end-entity certificate. If you do not specify this attribute, the length of the chain is unlimited.

- Netscape certificate type (Yes) _____
 - SSL client (Yes) _____

SSL Server Certificate Configuration

- Object-signing (No)_____
 - SSL server (Yes)_____
 - S/MIME CA (No)_____
 - S?MIME (No)_____
 - Object-signing CA (No)_____
 - SSL CA (No)_____
 - Authority Key Identifier (Yes) _____
 - Subject Key Identifier (No)
 - Key usage (No)_____
- If you decide to include the key usage extension, the following key usage bits are set by default:
- digitalSignature
 - keyEncipherment
- Additional Extension (No)_____
- To add extensions not included by default by CMS, you will need to paste the base64 encoding of a sequence of extensions into the wizard.

SSL Certificate Request

If you are obtaining your SSL server certificate from another CA, you need to know where to submit your certificate request.

If you are submitting your certificate request to a third-party CA, follow the instructions provided by that CA.

If you are submitting your certificate request to another Certificate Manager, you need to know its URL.

- End-entity URL for issuing Certificate Manager_____

Enter the URL for the end-entity gateway of the Certificate Manager that will issue the SSL server certificate. For example, `http://hostname:17006`.

Single Sign-On Password

Before you exit the Installation Wizard, it asks you to specify a single signon password. This password simplifies the way you subsequently sign on to Certificate Management System by storing the passwords for the internal database and tokens. Each time you log on, you're required to enter just this single password.

- Single signon password_____

Single Sign-On Password

Installation and Configuration

This chapter describes the procedure for installing a Certificate Management System instance. If you are migrating from a previous Certificate Server 1.x installation, first see Appendix A, “Migrating from Certificate Server 1.x.”

Before you use this chapter to guide you through an installation, you should have read Chapters 1 through 3 and filled out the worksheet provided by Chapter 4, “Installation Worksheet.”

This chapter contains the following sections:

- Installation Overview (page 174)
- Stage 1: Running the Installation Script (page 175)
- Stage 2: Using the Installation Wizard (page 182)
- Stage 3: Further Configuration Options (page 229)
- Stage 4: Creating Additional Instances or Certificate Manager Clones (page 229)

Installation Overview

Before you begin installation, make sure your system meets the requirements listed in “System Requirements” on page 78.

The installation process installs the Netscape Administration Server, Netscape Console, and Netscape Directory Server, as well as Netscape Certificate Management System. You typically create two instances of Directory Server: the first is for the configuration directory used by the local Administration Server; the second is used by Certificate Management System itself for its internal database.

You must have an Administration Server in each server root directory. Administration Server can use a local configuration directory or refer to an existing configuration directory installed elsewhere. You must install the Certificate Management System internal database directory locally.

The initial installation script installs Netscape Console and the binaries for the servers, and it creates and starts instances of Administration Server and Directory Server. After running the initial script, you use the Installation Wizard to create and configure instances of Certificate Management System. The wizard helps you through the configuration process of choosing subsystems and creating the necessary keys and certificates.

Installation Stages

Installing Certificate Management System in a single server root directory involves four stages:

- Stage 1: Run the installation script (`setup` on Unix, `setup.exe` on NT) to install Administration Server and Directory Server as necessary and perform the initial phase of Certificate Management System installation. These procedures are described in “Stage 1: Running the Installation Script” on page 175.
- Stage 2: Run the Installation Wizard to set up the initial configuration of the Certificate Management System instance. In this stage you specify which subsystems are to be part of this instance and generate the SSL client and server certificates for each subsystem. These procedures are described in “Stage 2: Using the Installation Wizard” on page 182.

- Stage 3: Use Netscape Console to further configure the new Certificate Management System instance, as needed. See “Stage 3: Further Configuration Options” on page 229. Complete instructions for CMS configuration with Netscape Console can be found in *Netscape Certificate Management System Administrator's Guide*.
- Stage 4 (optional): Use Netscape Console to create additional instances of the Certificate Management System in the same server root directory, and use the Installation Wizard to configure them. For a summary, see “Stage 4: Creating Additional Instances or Certificate Manager Clones” on page 229. Instructions for creating additional instances can be found in Chapter 4 “Installing and Uninstalling CMS Instances” in *Netscape Certificate Management System Administrator's Guide*.

Stage 1: Running the Installation Script

The `setup` program extracts files for the Administration Server, Directory Server, Netscape Console, and Certificate Management System and installs the binaries under the server root directory you have specified. It creates one instance of the Administration Server, one instance of the Directory Server, and one instance of the Certificate Management System, which is not yet configured. The `setup` program also installs Netscape Console and automatically starts the Administration Server and Directory Server.

As you run the initial installation script, the program stores your configuration choices and generates a initialization file, or installation cache. As installation proceeds, the stored initialization file states information about your choices so far. As a result, you can stop the installation process and restart it as necessary. Your choices to the point at which you stopped the installation are automatically restored by the initialization file, and the installation prompts resume at the point in which you left off.

This initialization file applies only to the installation of the Administration Server and Directory Server. If you want to use the file to do additional “silent” installations, see the documentation for these servers.

Running the Installation Script on Unix

To run the installation script on Unix, follow these steps:

1. Log in as `root` to install the servers on a Unix system. This is recommended, but not required. If you are not `root`, you can install only a local version in a directory to which you have write access, using ports higher than 1024, for which you are the administrator for all services.
2. Change to the directory on the distribution CD, and run the `setup` program.
3. Answer the questions that the script asks. You should have previously collected the requested information in the section “Information for Unix Installation Script” of Chapter 4, “Installation Worksheet.” Most questions have a default answer shown in square brackets before the prompt. To accept the default answer, press Enter at the prompt.

Answer the questions for a typical installation as follows:

1. **Would you like to continue with setup? [Yes]:** Press Enter.
2. **Do you agree to the license terms? [No]:** Type `yes` and press Enter.
3. **Select the items you would like to install [1]:** Accept the default to install the Netscape servers.
4. **Install location [/usr/netscape/server4]:** Enter a full pathname to the location where you want to install the servers. The location that you enter must be different from the directory from which you are running the `setup` program. You must have write access to the directory. If the directory that you specify does not exist, the `setup` program creates it for you.
5. **Specify the components you wish to install [All]:** Accept the default value, `All`, to accept the default server product components.
6. **Specify the components you wish to install [1,2,3]:** Enter the numbers corresponding to the server product components you wish to install, or press Enter to accept the default components.

7. **Specify the components you wish to install [1,2]:** Enter the numbers corresponding to the Directory Suite components you wish to install, or press Enter to accept the default components.
8. **Specify the components you wish to install [1,2]:** Enter the numbers corresponding to the Administration Services components you wish to install, or press Enter to accept the default components.
9. **Specify the components you wish to install [1,2]:** Enter the numbers corresponding to the CMS components you wish to install, or press Enter to accept the default components.
10. **Computer name [myhost.mydomain.com]:** Accept the default value to install on the local machine. Do not attempt to install remotely.
11. **System User [nobody]:** Enter the user ID that configuration directory will run as. Where your system supports it, accept the default user `nobody`, creating that user as necessary.
12. **System Group [nobody]:** Enter the group that the configuration directory will run as. Where your system supports it, accept the default group, `nobody`, creating that group as necessary.
13. **Do you want to register this software with an existing Netscape configuration directory server? [No]:** If you accept the default setting, the installation script installs a new instance of Directory Server for use as a configuration directory.

You can also choose to use a previously installed configuration directory. In this case, select "Use existing configuration directory server," then fill in the values that identify and provide access to the previously installed directory.

14. **Do you want to use another directory to store your data? [No]:** If you accept the default setting, the installation script either adds a user/group directory to the newly installed instance of Directory Server (if you accepted the default in step 13) or installs a new instance of Directory Server for use as a user/group directory.

You can also choose to use a previously installed user/group directory. In this case, enter Yes, then fill in the values that identify and provide access to the previously installed directory.

15. **Directory server network port [random #]:** Accept the default, which is either 389 or a randomly generated number, or enter any port number that is not and will not be used for another purpose.
If you are using an existing configuration directory, enter its port number.
16. **Directory server identifier [myhost]:** Enter a unique identifier for the new instance of the configuration directory.
If you are using an existing configuration directory, enter its identifier.
17. **Netscape configuration directory server administrator ID [admin]:**
Enter the name and password of the user who will authenticate to Netscape Console with full privileges. The password must be at least eight characters long.
If you are using an existing configuration directory, enter its administrator ID and password.
18. **Suffix [o=mydomain.com]:** Accept the default value for the suffix, or base DN, to be used for the directory tree.
19. **Directory Manager DN [cn=Directory Manager]:** Enter the distinguished name (DN) and password of the directory manager for the configuration directory. The password must be at least eight characters long.
This DN can be short and does not need to conform to any suffix configured for your directory. It also should not correspond to an actual entry stored in your directory.
20. **Administration Domain [mydomain.com]:** Accept the default value.
This domain name identifies the collection of servers that use the same configuration directory.
21. **Administration port [random #]:** Accept the default port number, which is randomly generated, or enter any port number that is not and will not be used for another purpose.
22. **Run Administration Server as [current login]:** Enter the user ID for the Administration Server process. If you are running as `root`, you can accept the default to run the server as `root`.
23. **Certificate Management System identifier [certificate]:** Enter a unique identifier for the new instance of Certificate Management System.

The script extracts and installs the binaries for all of the servers in the server root directory and creates and starts instances of the Administration Server and Directory Server.

When you have completed the installation script, you can complete the installation and configuration of the CMS instance by running the Installation Wizard. See “Stage 2: Using the Installation Wizard” on page 182.

Running the Installation Script on Windows NT

The `setup.exe` program extracts files for the Administration Server, Directory Server, Netscape Console, and Certificate Management System and installs the binaries under the server root directory you have specified. It creates one instance of Administration Server, one instance of Directory Server, and one instance of Certificate Management System, which is as yet unconfigured. The program installs Netscape Console, and automatically starts the Administration Server and Directory Server.

To run the installation script, follow these steps:

1. Double click `setup.exe` to run the installation program.
2. The installation dialog boxes prompt you to type in answers or make selections.
3. Answer the questions that the script asks. You should have previously collected the requested information in the section “Information for NT Installation Script” of Chapter 4, “Installation Worksheet.”

In the instructions that follow, the name that appears in the title bar of each setup screen is in boldface, followed by a description of the action you should take.

Answer the questions for a typical installation as follows:

1. **Welcome.** Click Next.
2. **Software License Agreement.** If you agree to all the terms of the License Agreement, click Yes.

3. **Select Server or Console Installation.** “Netscape Servers” is selected by default. Click Next to accept the default selection.
4. **Choose Installation Directory.** The default installation directory is `C:\Netscape\Server4`. To specify a server root directory different from the default, click Browse. Enter a full pathname, or navigate to the location where you want to install the servers, then click OK.

The location that you enter must be different from the directory from which you are running the setup program. You must have write access to the directory. If the directory that you specify does not exist, the program can create it for you. Click Next to continue.
5. **Select Products.** Four components are selected by default:
 - Netscape Server Products Core Components.
 - Netscape Directory Suite
 - Administration Services
 - Netscape Certificate Management System

You don’t need to select the fifth component, Netscape Directory Server 4.1 Synch Service, unless you want to set up the Directory Server Synchronization Service.

Click Next to accept the default selection.
6. **Directory Server 4.1.** “This instance will be the configuration directory server” is selected by default. If you accept the default setting, the installation script installs a new instance of Directory Server for use as a configuration directory.

You can also choose to use a previously installed configuration directory. In this case, select “Use existing configuration directory server,” then fill in the values that identify and provide access to the previously installed directory. Click Next to continue.
7. **Directory Server 4.1.** “Store data in this directory server” is selected by default. If you accept the default setting, the installation script either adds a user/group directory to the newly installed instance of Directory Server (if you accepted the default in step 7) or installs a new instance of Directory Server for use as a user/group directory.

You can also choose to use a previously installed user/group directory. In this case, select “Store data in an existing directory server,” then fill in the values that identify and provide access to the previously installed directory. Click Next to continue.

8. **Directory Server 4.1 Server Settings**

- **Server Identifier.** Enter a unique identifier for the new instance of the configuration directory. If you are using an existing configuration directory, enter its identifier.
- **Server Port.** Accept the default, or enter any port number that is not and will not be used for another purpose. The default is 389 if that port is not already used; otherwise, it is a randomly selected port number. If you are using an existing configuration directory, enter its port number.
- **Suffix.** Accept the default value for the suffix, or base DN, to be used for the directory tree.

When all three values are correct, click Next.

9. **Directory Server 4.1 Netscape Configuration Directory Server**

Administrator. Enter the administrator ID and password of the user who will authenticate to the directory console with full privileges. (Think of this as the root or superuser identity for Directory Server.) The password must be at least one character long. If you are using an existing configuration directory, enter its administrator ID and password. Click Next to continue.

10. **Directory Server 4.1 Administration Domain.** Click Next to accept the default value. This name, which should be your organization’s domain name, will be used for the collection of servers that use the same configuration directory.

11. **Directory Server 4.1 Directory Manager Settings.** Enter the distinguished name and password of the directory manager for the configuration directory. The password must be at least eight characters long.

This DN can be short and does not need to conform to any suffix configured for your directory. It also should not correspond to an actual entry stored in your directory. Click Next to continue.

12. **Administration Server Port Selection.** The default is 389 if that port is not already used; otherwise, it is a randomly selected port number. Accept the default port number, or enter any port number that is not and will not be used for another purpose. Click Next to continue.
13. **Netscape Certificate Management System Server Identifier.** Enter a unique identifier for the new instance of Certificate Management System. Click Next to continue.
14. **Configuration Summary.** This screen shows all of the components you are installing and the choices you have made for their configuration. Click Next to continue.
15. **Setup.** At this point, the installation script extracts and installs the binaries for all of the servers in the server root directory, and creates and starts instances of the Administration Server and Directory Server.
16. **Setup Complete.** “Restart my computer now” is selected by default. Click finish to accept the default. After the computer has rebooted, you’ll note that the Netscape Console window is displayed with its associated icons.

When you have completed the installation script, you can complete the installation and configuration of the CMS instance by running the Installation Wizard. See “Stage 2: Using the Installation Wizard” on page 182.

Stage 2: Using the Installation Wizard

After you have finished running the installation script, you use the Installation Wizard to create and configure an instance of Certificate Management System. The Installation Wizard is the same for both Unix and Windows NT.

To bring up Netscape Console and launch the Installation Wizard, follow these steps:

1. Start Netscape Console:
 - On a Windows NT system, click Start, then choose Programs, then Netscape Server Family, then Netscape Console 4.1. Alternatively, click the corresponding shortcut in the Netscape Server Products directory window displayed after setup completes.

- On a Unix system, open a command shell, change to the directory `/usr/netscape/server4`, and execute the file `startconsole`.
2. Log in as the administrator. On Unix systems, you will also need to specify the Administration Server URL that you specified during the installation script.

The main window of Netscape Console appears.

3. In the navigation tree at the left, open your computer, then open Server Group.
4. Select the instance of Certificate Management System that you named while running the installation script.
5. In the Netscape Certificate Management System panel at the right, click Open.

After a few moments, the Introduction screen for the Installation Wizard appears. You use the wizard to get the initial certificates and set the initial configuration for this instance of Certificate Management System.

Your route through the Installation Wizard instructions is determined by the choices you make. The instructions that follow cover a wide variety of standard decisions, but your installation requirements may bring up screens in a slightly different order.

Initial Configuration

In the instructions that follow, the panel title that appears below the title bar for each screen is in boldface, followed by information about the choices you need to make. You should have previously collected the requested information in the section “Initial Configuration” of Chapter 4, “Installation Worksheet.”

1. **Introduction.** Click Next.

If you have not yet installed an internal database for this instance, the Internal Database screen (step 2) appears. If you have previously installed an internal database for this instance, the Recreate Internal Database screen (step 3) appears.

2. **Internal Database.** Specify the LDAP server to use as the Certificate Management System internal database. This database is used to store information (such as certificates or certificate requests) used by all the subsystems you will be installing in this CMS instance. Click Next to continue. The wizard sets up the new internal database, which takes some time.
3. **Recreate Internal Database.** Specify whether you want to remove the existing database in order to create a new internal database, or use the existing internal database. Click Next to continue.
4. **Internal Database password.** A special screen that comes up only if you stop the configuration process partway through and then start over again, in which case the wizard needs to ask for the internal database password again. Click Next to continue.
5. **Administrator.** Enter the user ID, name, and password for the Certificate Management System Administrator. This is the administrator who can access the CMS window and control all CMS settings. Click Next to continue.
6. **Subsystems.** Select the subsystems you want to install or accept the default settings by clicking Next.

You can choose Certificate Manager and Data Recovery Manager together, or Data Recovery and Registration Manager together, or you can choose any individual manager, but you cannot install Certificate Manager and Registration Manager together. The Certificate Manager can be configured to perform all Registration Manager functions, so it's not necessary or possible to install both managers in the same instance.
7. **Remote Certificate Manager.** This screen appears only when you are installing a Registration Manager. Supply the host name and agent SSL port number for the remote Certificate Manager, then click Next to continue.
8. **Remote Data Recovery Manager.** This screen appears only when you are installing a standalone Registration Manager or a standalone Certificate Manager. If you have already installed a remote Data Recovery Manager that you want the new manager to use, click Yes and enter the remote Data Recovery Manager's host name and agent SSL port number. If you don't want to use a remote Data Recovery Manager, click No. Click Next to continue.

9. **Network Configuration.** Enter the port numbers for the ports used by the CMS instance.

The screens that appear next depend on which combination of subsystems you selected in step 6 above. For instructions, see the section that corresponds to your subsystem selection:

- “Certificate Manager Configuration” on page 186
- “Registration Manager Configuration” on page 192
- “Data Recovery Manager Configuration” on page 196
- “Certificate Manager and Data Recovery Manager Configuration” on page 200
- “Registration Manager and Data Recovery Manager Configuration” on page 211

Certificate Manager Configuration

To configure a Certificate Manager, perform the steps described under “Initial Configuration” then follow the steps described here. Some decisions you make determine the content and sequence of the screens that follow.

You should have previously collected the requested information in the section “Certificate Manager Configuration” of Chapter 4, “Installation Worksheet.”

1. **Server Migration from Certificate Server 1.x - Step 1.** Click Yes if you are migrating from Certificate Server 1.x, or No if you do not want to enable data migration. You should not click Yes unless you have performed the procedures described in Appendix A, “Migrating from Certificate Server 1.x.” Click Next to continue.

If you select Yes, the screen “Server Migration from Certificate Server 1.x - Step 2” is displayed. If you select No, you will not see this screen.

Server Migration from Certificate Server 1.x - Step 2. You will see this screen *only* if you selected Yes in the previous screen. You should have previously collected the requested information in the section “Server Migration from Certificate Server 1.x” of Chapter 4, “Installation Worksheet.”

Enter the pathname of the directory where the migration tool output files are located, select the token or tokens in which the Certificate Manager signing certificate and SSL server certificate will reside, and initialize the tokens with passwords.

Click Next to continue. For descriptions of the screens that follow successful server migration, see “Single Signon Configuration” on page 224.

2. **CA’s serial number range.** Specify the lowest serial number the CA should use in the “Starting serial number” field. If you only use one CA server, you can leave the “Ending serial number” blank to indicate no upper limit. If you use cloned CA servers to distribute load, you must specify an upper limit. For cloned CAs, be sure that the range of serial numbers does not overlap with any other CA server.
3. **CA Signing Certificate.** Select the type of CA for which you want to request a signing certificate:

- **Create self-signed CA certificate.** If you select this option, the Installation Wizard generates a self-signed root CA certificate. See “Self-Signed CA Certificate” (the next section) for details.
- **Create subordinate CA certificate request.** If you select this option, the Installation Wizard generates a certificate signing request that must be submitted to another CA. See “Subordinate CA Certificate Request” on page 189 for details.

Click Next to continue.

Self-Signed CA Certificate

You should have previously collected the information requested here in the section “CA Signing Certificate” of Chapter 4, “Installation Worksheet.”

1. **Key-Pair Information for Certificate Manager CA Signing Certificate.** The token you select is used to store the CA signing certificate and key pair. If you have not previously initialized the token’s password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for Certificate Manager CA Signing Certificate.** The values you enter here identify the CA signing certificate. You are not required to enter all the values, but must enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **Validity Period for Certificate Manager CA Signing Certificate.** The validity period for the CA signing certificate determines how soon you will have to renew the certificate, which can be a complex procedure. Enter the validity period, then click Next.
5. **Certificate Extensions for Certificate Manager CA Signing Certificate.** The default settings should work for most deployments. If necessary, you can add an additional extension by pasting its base-64 encoding in the space provided on this screen. For more information about extensions, see “Certificate Extensions.”

You can use tools provided in the CMS SDK samples directory for generating extensions to include in CA and other certificate requests. For details about these tools, check the samples package at this location:

```
<server_root>/cms_sdk/samples/exttools/
```

The certificate extension text field accepts a single extension blob. If you want to add multiple extensions, you should use the `ExtJoiner` program, which is also provided as a sample in the CMS samples directory. See the “Certificate Setup Wizard” section in Chapter 8, “Keys and Certificates,” in *Netscape Certificate Management System Administrator’s Guide* for detailed instructions.

Click Next to continue.

6. **Certificate Manager CA Signing Certificate Creation.** Click Next to generate and install the certificate.
7. **SSL Server Certificate.** Specify whether you want the Certificate Manager’s SSL server certificate to be signed by the Certificate Manager itself or by some other CA:
 - **Sign SSL certificate with my CA signing certificate.** If you select this option, the wizard uses the CA signing certificate you just created to sign the SSL server certificate. See “SSL Server Certificate from the Local CA” on page 220 for details.
 - **Create request for submission to another CA.** If you select this option, the wizard creates a certificate request that you must submit to another CA. See “SSL Server Certificate from a Remote CA” on page 221 for details.

Click Next to continue.

Subordinate CA Certificate Request

The instructions in this section describe the screens displayed if you selected “Create subordinate CA certificate request” on page 149.

1. **Key-Pair Information for Certificate Manager CA Signing Certificate.**
The token you select is used to store the CA signing certificate and key pair. If you have not previously initialized the token’s password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for Certificate Manager CA Signing Certificate.** The values you enter here identify the CA signing certificate. You are not required to enter all the values, but must enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **CA Signing Certificate Request Creation.** This informational screen tells you that the wizard has all the information required to generate the key pair and certificate request. Click Next to generate them.
5. **Submission of Request.** Choose whether you want to submit the request manually or send the request to a remote CMS server automatically.
For automatic enrollment, select “Send the request to a remote CMS now,” enter the hostname and end-entity port number, and select whether this end-entity port uses SSL. Click Next to submit the request, then continue with step .

Follow these steps to submit your certificate request manually. Click Next when you are ready to proceed to the next screen.

- Highlight the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) and copy it to the clipboard.

In addition to the copy on the clipboard, the screen informs you that the certificate request has been saved to a file. You can use either the copy on the clipboard or the copy in the file to transfer your request to the CA that will issue the subordinate CA’s signing certificate.

If you are submitting your request to a third-party CA, follow the instructions provided by that CA. If you are submitting your request to a CMS Certificate Manager, follow these steps:

- Go to the end-entity URL for the Certificate Manager that will issue the subordinate CA's signing certificate. For example, if you assigned the port number 17006 to the non-SSL end-entity port for your root CA, you would go to the URL `http://hostname.17006` to bring up the Certificate Manager page for end entities.
 - Click Manual under the Certificate Manager Enrollment heading in the left-hand frame.
 - Paste the request from the clipboard into the field labeled PKCS #10 Request and fill in any other required information.
 - Click Submit.
 - A Certificate Manager agent can follow the instructions in Step 7 to issue the certificate.
6. **Certificate Request Result.** This screen confirms that the request has been submitted. You can use the request ID provided to retrieve the certificate from the end-entity port once it has been issued. A Certificate Manager agent can follow the instructions in the next step to issue the certificate.
7. **A Certificate Manager agent issues the certificate.** After you submit the request, an agent for the Certificate Manager to which you submitted the request must approve it. For example, if you are the agent, go to the Agent Services page for the Certificate Manager (using the same computer where you got your agent certificate), choose Certificate Manager Agent Services, and follow these steps:
- Select List Requests, and then click Show Pending Requests and click Find. The pending request list is displayed.
 - Locate your request, then click Details to see it. After checking the rest of the certificate request, scroll down to the last section, labeled Privileges.

- Select the checkbox labeled “This certificate is for a Trusted Manager.” (Note that you must be a designated CMS administrator as well as an agent for this option to work correctly.)
- Type a user ID for the new Certificate Manager. This user ID can be the same that you specified in the certificate request, or it can be some other ID that you want to use to identify this manager in the CMS window of Netscape Console, such as CAEng.
- Click Do It.
- After the certificate is generated, click Show Certificate.
- When the certificate is displayed, scroll down to the base-64 encoded version of the certificate, highlight all the text (including -----BEGIN CERTIFICATE ----- and -----END CERTIFICATE -----), and copy it to the clipboard.

You can either paste the encoded certificate into a file or paste it directly into the Installation Wizard, as described in Step 9 below.

8. **Certificate Manager Signing Certificate Installation.** Click Yes to install the certificate now, or click No to install it at another time. The default is No. When you choose the default, you will continue with the configuration, and you’ll be asked at the end of the configuration if you want to install the certificate at that point.

If you have submitted your request to a third-party CA or to a remote Certificate Manager, you may have to wait days or weeks before you receive the certificate. In this case, you should click No, continue as far as you can with the configuration, and resume after you receive the certificate.

You should click Yes only if you have received the base-64 encoded certificate and are ready to install it. Click Next to continue.

9. **Location of Certificate.** Enter the location of the file in which the encoded certificate is located, or paste in a base-64 encoded certificate including header and footer in the text area provided. Click Next to continue.
10. **Certificate Details.** This informational screen displays the certificate so you can inspect its contents. Click Next to continue.
11. **Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. Follow these steps to do so:

- Go to the end-entity URL for the Certificate Manager that issued the subordinate CA signing certificate, select the Retrieval tab, then select Import CA Certificate Chain.
 - Select “Display the CA certificate chain in PKCS#7 for importing into a server,” then click Submit.
 - Copy the certificate to the clipboard.
 - Return to the Installation Wizard.
 - Paste the certificate chain into the text box, then click Next.
12. **SSL Server Certificate.** Specify how you want the Certificate Manager’s SSL server certificate to be issued:
- **Sign SSL Certificate with my CA signing certificate.** If you select this option, see “SSL Server Certificate from the Local CA” on page 220 for details.
 - **Create request for submission to another CA.** If you select this option, see “SSL Server Certificate from a Remote CA” on page 221 for details.

Click Next to continue.

Registration Manager Configuration

To configure a Registration Manager, perform the steps described under “Initial Configuration” then follow the steps described here. Some decisions you make determine the content and sequence of the screens that follow.

You should have previously collected the requested information in the section “Registration Manager Configuration” of Chapter 4, “Installation Worksheet.”

1. **Key-Pair Information for Registration Manager Signing Certificate.**
The token you select is used to store the Registration Manager signing certificate and key pair. If you have not previously initialized the token’s password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for Registration Manager Signing Certificate.** The values you enter here identify the Registration Manager’s signing certificate. You are not required to enter all the values, but must enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **Registration Manager Signing Certificate Request Creation.** This informational screen tells you that the wizard has all the information required to generate the key pair and certificate request. Click Next to generate them.
5. **Submission of Request.** Choose whether you want to submit the request manually or send the request to a remote CMS server automatically.

For automatic enrollment, select “Send the request to a remote CMS now,” enter the hostname and end-entity port number, and select whether this end-entity port uses SSL. Click Next to submit the request, then continue with step .

Follow these steps to submit your certificate request manually. Click Next when you are ready to proceed to the next screen.

- Highlight the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) and copy it to the clipboard.

In addition to the copy on the clipboard, the screen informs you that the certificate request has been saved to a file. You can use either the copy on the clipboard or the copy in the file to transfer your request to the CA that will issue the subordinate CA’s signing certificate.

If you are submitting your request to a third-party CA, follow the instructions provided by that CA. If you are submitting your request to a CMS Certificate Manager, follow these steps:

- Go to the end-entity URL for the Certificate Manager that will issue the subordinate CA's signing certificate. For example, if you assigned the port number 17006 to the non-SSL end-entity port for your root CA, you would go to the URL `http://hostname:17006` to bring up the Certificate Manager page for end entities.
 - Click Manual under the Certificate Manager Enrollment heading in the left-hand frame.
 - Paste the request from the clipboard into the field labeled PKCS #10 Request and fill in any other required information.
 - Click Submit.
 - A Certificate Manager agent can follow the instructions in Step 7 to issue the certificate.
6. **Certificate Request Result.** This screen confirms that the request has been submitted. You can use the request ID provided to retrieve the certificate from the end-entity port once it has been issued. A Certificate Manager agent can follow the instructions in the next step to issue the certificate.
7. **A Certificate Manager agent issues the certificate.** After you submit the request, an agent for the Certificate Manager to which you submitted the request must approve it. For example, if you are the agent, go to the Agent Services page for the Certificate Manager (using the same computer where you got your certificate), choose Certificate Manager Agent Services, and follow these steps:
- Select List Requests, and then click Show Pending Requests and click Find. The pending request list is displayed.
 - Locate your request, then click Details to see it. After checking the rest of the certificate request, scroll down to the last section, labeled Privileges.

- Select the checkbox labeled “This certificate is for a Trusted Manager.” (Note that you must be a designated CMS administrator as well as an agent for this option to work correctly.)
- Type a user ID for the new Registration Manager. This user ID can be the same that you specified in the certificate request, or it can be some other ID that you want to use to identify this manager in the CMS window of Netscape Console, such as RMEng.
- Click Do It.
- After the certificate is generated, click Show Certificate.
- When the certificate is displayed, scroll down to the base-64 encoded version of the certificate, highlight all the text (including -----BEGIN CERTIFICATE ----- and -----END CERTIFICATE -----), and copy it to the clipboard.

You can either paste the encoded certificate into a file or paste it directly into the Installation Wizard, as described in Step 9 below.

8. **Registration Manager Signing Certificate Installation.** Click Yes to install the certificate now, or click No to install it at another time. The default is No. When you choose the default, you will continue with the configuration, and you’ll be asked at the end of the configuration if you want to install the certificate at that point.

If you have submitted your request to a third-party CA or to a remote Certificate Manager, you may have to wait days or weeks before you receive the certificate. In this case, you should click No, continue as far as you can with the configuration, and restart the Installation Wizard after you receive the certificate. For a description of the screens that follow if you choose this option, see “SSL Server Certificate from a Remote CA” on page 221.

You should click Yes only if you have received the base-64 encoded certificate and are ready to install it. Click Next to continue.

9. **Location of Certificate.** Enter the location of the file in which the encoded certificate is located, or paste in a base-64 encoded certificate including header and footer in the text area provided. Click Next to continue.
10. **Certificate Details.** This informational screen displays the certificate so you can inspect its contents. Click Next to continue.

11. **Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. Follow these steps to do so:

- Go to the end-entity URL for the Certificate Manager that issued the Registration Manager's signing certificate, select the Retrieval tab, then select Import CA Certificate Chain.
- Select "Display the CA certificate chain in PKCS#7 for importing into a server," then click Submit.
- Copy the certificate to the clipboard.
- Return to the Installation Wizard.
- Paste the certificate chain into the text box, then click Next.

At this point, the wizard displays a series of screens that you use to request an SSL server certificate for the Registration Manager. See "SSL Server Certificate from a Remote CA" on page 221 for details.

Data Recovery Manager Configuration

To configure a Data Recovery Manager, perform the steps described under "Initial Configuration" then follow the steps described here. Note that some decisions you make will affect the screens you see.

You should have previously collected the requested information in the section "Data Recovery Manager Configuration" of Chapter 4, "Installation Worksheet."

Transport Certificate from a Remote CA

1. **Key-Pair Information for Data Recovery Manager Transport Certificate.** The token you select is used to store the transport certificate and key pair. If you have not previously initialized the token's password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.

3. **Subject Name for Data Recovery Manager Transport Certificate.** The values you enter here identify the CA signing certificate. You are not required to enter all the values, but must enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **Data Recovery Manager Transport Certificate Request Creation.** This informational screen tells you that the wizard has all the information required to generate the key pair and certificate request. Click Next to generate them.
5. **Submission of Request.** Choose whether you want to submit the request manually or send the request to a remote CMS server automatically.
For automatic enrollment, select “Send the request to a remote CMS now,” enter the hostname and end-entity port number, and select whether this end-entity port uses SSL. Click Next to submit the request, then continue with step .

Follow these steps to submit your certificate request manually. Click Next when you are ready to proceed to the next screen.

- Highlight the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) and copy it to the clipboard.

In addition to the copy on the clipboard, the screen informs you that the certificate request has been saved to a file. You can use either the copy on the clipboard or the copy in the file to transfer your request to the CA that will issue the subordinate CA's signing certificate.

If you are submitting your request to a third-party CA, follow the instructions provided by that CA. If you are submitting your request to a CMS Certificate Manager, follow these steps:

- Go to the end-entity URL for the Certificate Manager that will issue the subordinate CA's signing certificate. For example, if you assigned the port number 17006 to the non-SSL end-entity port for your root CA, you would go to the URL `http://hostname.17006` to bring up the Certificate Manager page for end entities.

- Click Manual under the Certificate Manager Enrollment heading in the left-hand frame.
 - Paste the request from the clipboard into the field labeled PKCS #10 Request and fill in any other required information.
 - Click Submit.
 - A Certificate Manager agent can follow the instructions in Step 7 to issue the certificate.
6. **Certificate Request Result.** This screen confirms that the request has been submitted. You can use the request ID provided to retrieve the certificate from the end-entity port once it has been issued. A Certificate Manager agent can follow the instructions in the next step to issue the certificate.
7. **A Certificate Manager agent issues the certificate.** After you submit the request, an agent for the Certificate Manager to which you submitted the request must approve it. For example, if you are the agent, go to the Agent Services page for the Certificate Manager (using the same computer where you got your agent certificate), choose Certificate Manager Agent Services, and follow these steps:
- Select List Requests, and then click Show Pending Requests and click Find. The pending request list is displayed.
 - Locate your request, then click Details to see it. After checking the rest of the certificate request, scroll down to the end of the form.
 - Click Do It.
 - After the certificate is generated, click Show Certificate.
 - When the certificate is displayed, scroll down to the base-64 encoded version of the certificate, highlight all the text (including -----BEGIN CERTIFICATE ----- and -----END CERTIFICATE -----), and copy it to the clipboard.

You can either paste the encoded certificate into a file or paste it directly into the Installation Wizard, as described in Step 9 below.

8. **Data Recovery Manager Transport Certificate Installation.** Click Yes to install the certificate now, or click No to install it at another time. The default is No. When you choose the default, you will continue with the configuration, and you'll be asked at the end of the configuration if you want to install the certificate at that point.

If you have submitted your request to a third-party CA or to a remote Certificate Manager, you may have to wait days or weeks before you receive the certificate. In this case, you should click No, continue as far as you can with the configuration, and resume after you receive the certificate. For a description of the screens that follow if you choose this option, see "Storage Key and Recovery Agent Configuration" on page 200.

You should click Yes only if you have received the base-64 encoded certificate and are ready to install it. Click Next to continue.

9. **Location of Certificate.** Enter the location of the file in which the encoded certificate is located, or paste in a base-64 encoded certificate including header and footer in the text area provided. Click Next to continue.
10. **Certificate Details.** This informational screen displays the certificate so you can inspect its contents. Click Next to continue.
11. **Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. Follow these steps to do so:
 - Go to the end-entity URL for the Certificate Manager that issued the transport certificate, select the Retrieval tab, then select Import CA Certificate Chain.
 - Select "Display the CA certificate chain in PKCS#7 for importing into a server," then click Submit.
 - Copy the certificate to the clipboard.
 - Return to the Installation Wizard.
 - Paste the certificate chain into the text box, then click Next.

Storage Key and Recovery Agent Configuration

The following screens let you configure the storage key and recovery schemes for the Data Recovery Manager.

1. **Storage Key Creation for Data Recovery Manager.** Select the length you have decided on for your storage key, then click Next to continue.
2. **Data Recovery Key Scheme - 1.** Enter the both the required number of recovery agents and the total number of recovery agents, then click Next.
3. **Data Recovery Key Scheme - 2.** The number of table rows correspond to the total number of agents you specified in the previous screen. Enter the user ID and password for each agent in the table, then click Next.

At this point, the wizard displays a series of screens that you use to request an SSL server certificate for the Data Recovery Manager. See “SSL Server Certificate from a Remote CA” on page 221 for details.

Certificate Manager and Data Recovery Manager Configuration

To configure a Certificate Manager and Data Recovery Manager in the same instance, perform the steps described under “Initial Configuration” then follow the steps described here. Some decisions you make determine the content and sequence of the screens that follow.

You should have previously collected the requested information in the sections “Certificate Manager Configuration” and “Data Recovery Manager Configuration” of Chapter 4, “Installation Worksheet.”

Certificate Manager Configuration

To configure the Certificate Manager, follow these steps:

1. **Server Migration from Certificate Server 1.x - Step 1.** Click Yes if you are migrating from Certificate Server 1.x, or No if you do not want to enable datamigration. You should not click Yes unless you have performed the procedures described in Appendix A, “Migrating from Certificate Server 1.x,” on page 93. Click Next to continue.

If you selected Yes, the screen “Server Migration from Certificate Server 1.x - Step 2” is displayed. If you selected No, you will not see this screen.

Server Migration from Certificate Server 1.x - Step 2. You will see this screen *only* if you selected Yes in the previous screen. You should have previously collected the requested information in the section “Server Migration from Certificate Server 1.x” of “Installation Worksheet”

Enter the pathname of the directory where the migration tool output files are located, select the token or tokens in which the Certificate Manager signing certificate and SSL server certificate will reside, and initialize the tokens with passwords.

Click Next to continue. For descriptions of the screens that follow successful server migration, see “Single Signon Configuration” on page 224.

2. **CA’s serial number range.** Specify the lowest serial number the CA should use in the “Starting serial number” field. If you only use one CA server, you can leave the “Ending serial number” blank to indicate no upper limit. If you use cloned CA servers to distribute load, you must specify an upper limit. For cloned CAs, be sure that the range of serial numbers does not overlap with any other CA server.
3. **CA Signing Certificate.** Select the type of CA for which you want to request a signing certificate:
 - **Create self-signed CA certificate.** If you select this option, the Installation Wizard generates a self-signed root CA certificate. See “Self-Signed CA Certificate” (the next section) for details.

- **Create subordinate CA certificate request.** If you select this option, the Installation Wizard generates a certificate signing request that must be submitted to another CA. See “Subordinate CA Certificate Request” on page 203 for details.

Click Next to continue.

Self-Signed CA Certificate

1. **Key-Pair Information for Certificate Manager CA Signing Certificate.** The token you select is used to store the CA signing certificate and key pair. If you have not previously initialized the token’s password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for Certificate Manager CA Signing Certificate.** The values you enter here identify the CA signing certificate. You are not required to enter all the values, but must enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **Validity Period for Certificate Manager CA Signing Certificate.** The validity period for the CA signing certificate determines how soon you will have to renew the certificate, which can be a complex procedure. Enter the validity period, then click Next.
5. **Certificate Extensions for Certificate Manager CA Signing Certificate.** The default settings should work for most deployments. If necessary, you can add an additional extension by pasting its base-64 encoding in the space provided on this screen. For more information about extensions, see Appendix B, “Certificate Extensions.”

You can use tools provided in the CMS SDK samples directory for generating extensions to include in CA and other certificate requests. For details about these tools, check the samples package at this location:

```
<server_root>/cms_sdk/samples/exttools/
```

The certificate extension text field accepts a single extension blob. If you want to add multiple extensions, you should use the `ExtJoiner` program, which is also provided as a sample in the CMS samples directory. See the “Certificate Setup Wizard” section in Chapter 8, “Keys and Certificates,” in *Netscape Certificate Management System Administrator’s Guide* for detailed instructions.

Click Next to continue.

6. **Certificate Manager CA Signing Certificate Creation.** Click Next to generate and install the certificate.

At this point, the Installation wizard begins configuration of the Data Recovery Manager. For details, see “Data Recovery Manager Configuration” on page 206.

Subordinate CA Certificate Request

1. **Key-Pair Information for Certificate Manager CA Signing Certificate.** The token you select is used to store the CA signing certificate and key pair. If you have not previously initialized the token’s password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for Certificate Manager CA Signing Certificate.** The values you enter here identify the CA signing certificate and key pair. You are not required to enter all the values, but must enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **CA Signing Certificate Request Creation.** This informational screen tells you that the wizard has all the information required to generate the key pair and certificate request. Click Next to generate them.
5. **Submission of Request.** Choose whether you want to submit the request manually or send the request to a remote CMS server automatically.
For automatic enrollment, select “Send the request to a remote CMS now,” enter the hostname and end-entity port number, and select whether this end-entity port uses SSL. Click Next to submit the request, then continue with step .

Follow these steps to submit your certificate request manually. Click Next when you are ready to proceed to the next screen.

- Highlight the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) and copy it to the clipboard.

In addition to the copy on the clipboard, the screen informs you that the certificate request has been saved to a file. You can use either the copy on the clipboard or the copy in the file to transfer your request to the CA that will issue the subordinate CA's signing certificate.

If you are submitting your request to a third-party CA, follow the instructions provided by that CA. If you are submitting your request to a CMS Certificate Manager, follow these steps:

- Go to the end-entity URL for the Certificate Manager that will issue the subordinate CA's signing certificate. For example, if you assigned the port number 17006 to the non-SSL end-entity port for your root CA, you would go to the URL `http://hostname.17006` to bring up the Certificate Manager page for end entities.
 - Click Manual under the Certificate Manager Enrollment heading in the left-hand frame.
 - Paste the request from the clipboard into the field labeled PKCS #10 Request and fill in any other required information.
 - Click Submit.
 - A Certificate Manager agent can follow the instructions in Step 7 to issue the certificate.
6. **Certificate Request Result.** This screen confirms that the request has been submitted. You can use the request ID provided to retrieve the certificate from the end-entity port once it has been issued. A Certificate Manager agent can follow the instructions in the next step to issue the certificate.
 7. **A Certificate Manager agent issues the certificate.** After you submit the request, an agent for the Certificate Manager to which you submitted the request must approve it. For example, if you are the agent, go to the Agent

Services page for the Certificate Manager (using the same computer where you got your agent certificate), choose Certificate Manager Agent Services, and follow these steps:

- Select List Requests, and then click Show Pending Requests and click Find. The pending request list is displayed.
- Locate your request, then click Details to see it. After checking the rest of the certificate request, scroll down to the last section, labeled Privileges.
- Select the checkbox labeled “This certificate is for a Trusted Manager.” (Note that you must be a designated CMS administrator as well as an agent for this option to work correctly.)
- Type a user ID for the new Certificate Manager. This user ID can be the same that you specified in the certificate request, or it can be some other ID that you want to use to identify this manager in the CMS window of Netscape Console, such as CAEng.
- Click Do It.
- After the certificate is generated, click Show Certificate.
- When the certificate is displayed, scroll down to the base-64 encoded version of the certificate, highlight all the text (including -----BEGIN CERTIFICATE ----- and -----END CERTIFICATE -----), and copy it to the clipboard.

You can either paste the encoded certificate into a file or paste it directly into the Installation Wizard, as described in Step 9 below.

8. **Certificate Manager Signing Certificate Installation.** Click Yes to install the certificate now, or click No to install it at another time. The default is No. When you choose the default, you will continue with the configuration, and you’ll be asked at the end of the configuration if you want to install the certificate at that point.

If you have submitted your request to a third-party CA or to a remote Certificate Manager, you may have to wait days or weeks before you receive the certificate. In this case, you should click No, continue as far as you can with the configuration, and resume after you receive the certificate.

You should click Yes only if you have received the base-64 encoded certificate and are ready to install it. Click Next to continue.

9. **Location of Certificate.** Enter the location of the file in which the encoded certificate is located, or paste in a base-64 encoded certificate including header and footer in the text area provided. Click Next to continue
10. **Certificate Details.** This informational screen displays the certificate so you can inspect its contents. Click Next to continue.
11. **Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. Follow these steps to do so:
 - Go to the end-entity URL for the Certificate Manager that issued the subordinate CA signing certificate, select the Retrieval tab, then choose Import CA Certificate Chain.
 - Select “Display the CA certificate chain in PKCS#7 for importing into a server,” then click Submit.
 - Copy the certificate to the clipboard.
 - Return to the Installation Wizard.
 - Paste the certificate chain into the text box, then click Next.

At this point, the Installation Wizard begins configuration of the Data Recovery Manager. For details, see the next section.

Data Recovery Manager Configuration

To configure a Data Recovery Manager, follow these steps:

1. **Data Recovery Manager Transport Certificate.** Specify how you want the Data Recovery Manager Transport Certificate to be issued:
 - **Sign Data Recovery Manager Transport Certificate with my CA signing certificate.** If you select this option, see “Transport Certificate from Local CA” (the next section) for details.
 - **Create request for submission to another CA.** If you select this option, see “Transport Certificate from Remote CA” on page 208 for details.

Transport Certificate from Local CA

1. **Key-Pair Information for Data Recovery Manager Transport Certificate.** The token you select is used to store the transport certificate and key pair. If you have not previously initialized the token's password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for Data Recovery Manager Transport Certificate.** The values you enter here identify the CA signing certificate. You are not required to enter all the values, but must enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **Validity Period for Data Recovery Manager Transport Certificate.** The validity period for the transport certificate determines how soon you will have to renew the certificate. Enter the validity period, then click Next.
5. **Certificate Extensions for Data Recovery Manager Transport Certificate.** The default settings should work for most deployments. If necessary, you can add additional extensions by pasting the base-64 encoding for each extension in the space provided on this screen. For more information about extensions, see Appendix C, "Certificate Extensions". Click Next to continue.
6. **Data Recovery Manager Transport Certificate Creation.** Click Next to generate and install the certificate.

To continue configuring the Data Recovery Manager, go to "Storage Key and Recovery Agent Configuration" on page 211.

Transport Certificate from Remote CA

1. **Key-Pair Information for Data Recovery Manager Transport Certificate.** The token you select is used to store the transport certificate and key pair. If you have not previously initialized the token's password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for Data Recovery Manager Transport Certificate.** The values you enter here identifies the transport certificate. You are not required to enter all the values, but must enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **Data Recovery Manager Transport Certificate Request Creation.** This informational screen tells you that the wizard has all the information required to generate the key pair and certificate request. Click Next to generate them.
5. **Submission of Request.** Choose whether you want to submit the request manually or send the request to a remote CMS server automatically.
For automatic enrollment, select "Send the request to a remote CMS now," enter the hostname and end-entity port number, and select whether this end-entity port uses SSL. Click Next to submit the request, then continue with step .

Follow these steps to submit your certificate request manually. Click Next when you are ready to proceed to the next screen.

- Highlight the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) and copy it to the clipboard.

In addition to the copy on the clipboard, the screen informs you that the certificate request has been saved to a file. You can use either the copy on the clipboard or the copy in the file to transfer your request to the CA that will issue the subordinate CA's signing certificate.

If you are submitting your request to a third-party CA, follow the instructions provided by that CA. If you are submitting your request to a CMS Certificate Manager, follow these steps:

- Go to the end-entity URL for the Certificate Manager that will issue the subordinate CA's signing certificate. For example, if you assigned the port number 17006 to the non-SSL end-entity port for your root CA, you would go to the URL `http://hostname.17006` to bring up the Certificate Manager page for end entities.
 - Click Manual under the Certificate Manager Enrollment heading in the left-hand frame.
 - Paste the request from the clipboard into the field labeled PKCS #10 Request and fill in any other required information.
 - Click Submit.
 - A Certificate Manager agent can follow the instructions in Step 7 to issue the certificate.
6. **Certificate Request Result.** This screen confirms that the request has been submitted. You can use the request ID provided to retrieve the certificate from the end-entity port once it has been issued. A Certificate Manager agent can follow the instructions in the next step to issue the certificate.
7. **A Certificate Manager agent issues the certificate.** After you submit the request, an agent for the Certificate Manager to which you submitted the request must approve it. For example, if you are the agent, go to the Agent Services page for the Certificate Manager (using the same computer where you got your agent certificate), choose Certificate Manager Agent Services, and follow these steps:
- Select List Requests, and then click Show Pending Requests and click Find. The pending request list is displayed.
 - Locate your request, then click Details to see it. After checking the rest of the certificate request, scroll down to the end of the form.
 - Click Do It.
 - After the certificate is generated, click Show Certificate.

- When the certificate is displayed, scroll down to the base-64 encoded version of the certificate, highlight all the text (including -----BEGIN CERTIFICATE ----- and -----END CERTIFICATE -----), and copy it to the clipboard.

You can either paste the encoded certificate into a file or paste it directly into the Installation Wizard, as described in Step 9 below.

8. **Data Recovery Manager Transport Certificate Installation.** Click Yes to install the certificate now, or click No to install it at another time. The default is No. When you choose the default, you will continue with the configuration, and you'll be asked at the end of the configuration if you want to install the certificate at that point.

If you have submitted your request to a third-party CA or to a remote Certificate Manager, you may have to wait days or weeks before you receive the certificate. In this case, you should click No, continue as far as you can with the configuration, and resume after you receive the certificate. For a description of the screens that follow if you choose this option, see "Storage Key and Recovery Agent Configuration" on page 211.

You should click Yes only if you have received the base-64 encoded certificate and are ready to install it. Click Next to continue.

9. **Location of Certificate.** Enter the location of the file in which the encoded certificate is located, or paste in a base-64 encoded certificate including header and footer in the text area provided. Click Next to continue.
10. **Certificate Details.** This informational screen displays the certificate so you can inspect its contents. Click Next to continue.
11. **Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. Follow these steps to do so:

- Go to the end-entity URL for the Certificate Manager that issued the transport certificate, select the Retrieval tab, then choose Import CA Certificate Chain.
- Select "Display the CA certificate chain in PKCS #7 for importing into a server," then click Submit.
- Copy the certificate to the clipboard.
- Return to the Installation Wizard.

- Paste the certificate chain into the text box, then click Next.

Storage Key and Recovery Agent Configuration

1. **Storage Key Creation for Data Recovery Manager.** Select the length you have decided on for your storage key, then click Next to continue.
2. **Data Recovery Key Scheme - 1.** Enter the both the required number of recovery agents and the total number of recovery agents, then click Next.
3. **Data Recovery Key Scheme - 2.** The number of table rows correspond to the total number of agents you specified in the previous screen. Enter the user ID and password for each agent in the table, then click Next.
4. **SSL Server Certificate.** Specify how you want the SSL server certificate for this instance of Certificate Management System to be issued:
 - **Sign SSL certificate with my CA signing certificate.** If you select this option, the wizard uses the CA signing certificate you just created to sign the SSL server certificate. See “SSL Server Certificate from the Local CA” on page 220 for details.
 - **Create request for submission to another CA.** If you select this option, the wizard creates a certificate request that you must submit to another CA. See “SSL Server Certificate from a Remote CA” on page 221 for details.

Click Next to continue.

Registration Manager and Data Recovery Manager Configuration

To configure a Registration Manager and Data Recovery Manager in the same instance, perform the steps described under “Initial Configuration,” then follow the steps described here. Some decisions you make determine the content and sequence of the screens that follow.

You should have previously collected the requested information in the sections “Registration Manager Configuration” and “Data Recovery Manager Configuration” of Chapter 4, “Installation Worksheet.”

Registration Manager Configuration

1. **Key-Pair Information for Registration Manager Signing Certificate.**
The token you select is used to store the Registration Manager signing certificate and key pair. If you have not previously initialized the token's password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for Registration Manager Signing Certificate.** The values you enter here identify the CA signing certificate. You are not required to enter all the values, but must enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **Registration Manager Signing Certificate Request Creation.** This informational screen tells you that the wizard has all the information required to generate the key pair and certificate request. Click Next to generate them.
5. **Submission of Request.** Choose whether you want to submit the request manually or send the request to a remote CMS server automatically.
For automatic enrollment, select "Send the request to a remote CMS now," enter the hostname and end-entity port number, and select whether this end-entity port uses SSL. Click Next to submit the request, then continue with step .

Follow these steps to submit your certificate request manually. Click Next when you are ready to proceed to the next screen.

- Highlight the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) and copy it to the clipboard.

In addition to the copy on the clipboard, the screen informs you that the certificate request has been saved to a file. You can use either the copy on the clipboard or the copy in the file to transfer your request to the CA that will issue the subordinate CA's signing certificate.

If you are submitting your request to a third-party CA, follow the instructions provided by that CA. If you are submitting your request to a CMS Certificate Manager, follow these steps:

- Go to the end-entity URL for the Certificate Manager that will issue the subordinate CA's signing certificate. For example, if you assigned the port number 17006 to the non-SSL end-entity port for your root CA, you would go to the URL `http://hostname.17006` to bring up the Certificate Manager page for end entities.
 - Click Manual under the Certificate Manager Enrollment heading in the left-hand frame.
 - Paste the request from the clipboard into the field labeled PKCS #10 Request and fill in any other required information.
 - Click Submit.
 - A Certificate Manager agent can follow the instructions in Step 7 to issue the certificate.
6. **Certificate Request Result.** This screen confirms that the request has been submitted. You can use the request ID provided to retrieve the certificate from the end-entity port once it has been issued. A Certificate Manager agent can follow the instructions in the next step to issue the certificate.
7. **A Certificate Manager agent issues the certificate.** After you submit the request, an agent for the Certificate Manager to which you submitted the request must approve it. For example, if you are the agent, go to the Agent Services page for the Certificate Manager (using the same computer where you got your agent certificate), choose Certificate Manager Agent Services, and follow these steps:
- Select List Requests, and then click Show Pending Requests and click Find. The pending request list is displayed.
 - Locate your request, then click Details to see it. After checking the rest of the certificate request, scroll down to the last section, labeled Privileges.

- Select the checkbox labeled “This certificate is for a Trusted Manager.” (Note that you must be a designated CMS administrator as well as an agent for this option to work correctly.)
- Type a user ID for the new Registration Manager. This user ID can be the same that you specified in the certificate request, or it can be some other ID that you want to use to identify this manager in the CMS window of Netscape Console, such as RMEng.
- Click Do It.
- After the certificate is generated, click Show Certificate.
- When the certificate is displayed, scroll down to the base-64 encoded version of the certificate, highlight all the text (including -----BEGIN CERTIFICATE ----- and -----END CERTIFICATE -----), and copy it to the clipboard.

You can either paste the encoded certificate into a file or paste it directly into the Installation Wizard, as described in Step 9 below.

- 8. Registration Manager Signing Certificate Installation.** Click Yes to install the certificate now, or click No to install it at another time. The default is No. When you choose the default, you will continue with the configuration, and you’ll be asked at the end of the configuration if you want to install the certificate at that point.

If you have submitted your request to a third-party CA or to a remote Certificate Manager, you may have to wait days or weeks before you receive the certificate. In this case, you should click No, continue as far as you can with the configuration, and resume after you receive the certificate. For a description of the screens that follow if you choose this option, see “Data Recovery Manager Configuration” on page 215.

You should click Yes only if you have received the base-64 encoded certificate and are ready to install it. Click Next to continue.
- 9. Location of Certificate.** Enter the location of the file in which the encoded certificate is located, or paste in a base-64 encoded certificate including header and footer in the text area provided. Click Next to continue.
- 10. Certificate Details.** This informational screen displays the certificate so you can inspect its contents. Click Next to continue.

11. **Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. Follow these steps to do so:
 - Go to the end-entity URL for the Certificate Manager that issued the Registration Manager's signing certificate, select the Retrieval tab, then select Import CA Certificate Chain.
 - Select "Display the CA certificate chain in PKCS#7 for importing into a server," then click Submit.
 - Copy the certificate to the clipboard.
 - Return to the Installation Wizard.
 - Paste the certificate chain into the text box, then click Next.

Data Recovery Manager Configuration

Transport Certificate from a Remote CA

1. **Key-Pair Information for Data Recovery Manager Transport Certificate.** The token you select is used to store the transport certificate and key pair. If you have not previously initialized the token's password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for Data Recovery Manager Transport Certificate.** The values you enter here identify the CA signing certificate. You are not required to enter all the values, but must enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **Data Recovery Manager Transport Certificate Request Creation.** This informational screen tells you that the wizard has all the information required to generate the key pair and certificate request. Click Next to generate them.

5. **Submission of Request.** Choose whether you want to submit the request manually or send the request to a remote CMS server automatically.

For automatic enrollment, select “Send the request to a remote CMS now,” enter the hostname and end-entity port number, and select whether this end-entity port uses SSL. Click Next to submit the request, then continue with step .

Follow these steps to submit your certificate request manually. Click Next when you are ready to proceed to the next screen.

- Highlight the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) and copy it to the clipboard.

In addition to the copy on the clipboard, the screen informs you that the certificate request has been saved to a file. You can use either the copy on the clipboard or the copy in the file to transfer your request to the CA that will issue the subordinate CA’s signing certificate.

If you are submitting your request to a third-party CA, follow the instructions provided by that CA. If you are submitting your request to a CMS Certificate Manager, follow these steps:

- Go to the end-entity URL for the Certificate Manager that will issue the subordinate CA’s signing certificate. For example, if you assigned the port number 17006 to the non-SSL end-entity port for your root CA, you would go to the URL `http://hostname.17006` to bring up the Certificate Manager page for end entities.
- Click Manual under the Certificate Manager Enrollment heading in the left-hand frame.
- Paste the request from the clipboard into the field labeled PKCS #10 Request and fill in any other required information.
- Click Submit.
- A Certificate Manager agent can follow the instructions in Step 7 to issue the certificate.

6. **Certificate Request Result.** This screen confirms that the request has been submitted. You can use the request ID provided to retrieve the certificate from the end-entity port once it has been issued. A Certificate Manager agent can follow the instructions in the next step to issue the certificate.
7. **A Certificate Manager agent issues the certificate.** After you submit the request, an agent for the Certificate Manager to which you submitted the request must approve it. For example, if you are the agent, go to the Agent Services page for the Certificate Manager (using the same computer where you got your agent certificate), choose Certificate Manager Agent Services, and follow these steps:
 - Select List Requests, and then click Show Pending Requests and click Find. The pending request list is displayed.
 - Locate your request, then click Details to see it. After checking the rest of the certificate request, scroll down to the end of the form.
 - Click Do It.
 - After the certificate is generated, click Show Certificate.
 - When the certificate is displayed, scroll down to the base-64 encoded version of the certificate, highlight all the text (including -----BEGIN CERTIFICATE ----- and -----END CERTIFICATE -----), and copy it to the clipboard.

You can either paste the encoded certificate into a file or paste it directly into the Installation Wizard, as described in Step 9 below.

8. **Data Recovery Manager Transport Certificate Installation.** Click Yes to install the certificate now, or click No to install it at another time. The default is No. When you choose the default, you will continue with the configuration, and you'll be asked at the end of the configuration if you want to install the certificate at that point.

If you have submitted your request to a third-party CA or to a remote Certificate Manager, you may have to wait days or weeks before you receive the certificate. In this case, you should click No, continue as far as you can with the configuration, and resume after you receive the certificate. For a description of the screens that follow if you choose this option, see "Storage Key and Recovery Agent Configuration" on page 218.

You should click Yes only if you have received the base-64 encoded certificate and are ready to install it. Click Next to continue.

9. **Location of Certificate.** Enter the location of the file in which the encoded certificate is located, or paste in a base-64 encoded certificate including header and footer in the text area provided. Click Next to continue.
10. **Certificate Details.** This informational screen displays the certificate so you can inspect its contents. Click Next to continue.
11. **Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. Follow these steps to do so:
 - Go to the end-entity URL for the Certificate Manager that issued the transport certificate, select the Retrieval tab, then select Import CA Certificate Chain.
 - Select “Display the CA certificate chain in PKCS#7 for importing into a server,” then click Submit.
 - Copy the certificate to the clipboard.
 - Return to the Installation Wizard.
12. Paste the certificate chain into the text box, then click Next.

Storage Key and Recovery Agent Configuration

1. **Storage Key Creation for Data Recovery Manager.** Select the length you have decided on for your storage key, then click Next to continue.
2. **Data Recovery Key Scheme - 1.** Enter the both the required number of recovery agents and the total number of recovery agents, then click Next.
3. **Data Recovery Key Scheme - 2.** The number of table rows correspond to the total number of agents you specified in the previous screen. Enter the user ID and password for each agent in the table, then click Next.

To continue configuring the Data Recovery Manager, go to “SSL Server Certificate from a Remote CA” on page 221.

Cloned Certificate Manager Configuration

To configure a cloned Certificate Manager, install the software or create a new CMS instance and copy the `key3.db` and `cert7.db` files from the `config` directory of the original server to the `config` directory of the clone. You must copy the database files before you perform the steps described under “Initial Configuration.” Finally, follow the steps described here. Some decisions you make determine the content and sequence of the screens that follow.

You should have previously collected the requested information in the section “Cloned Certificate Manager Configuration” of Chapter 4, “Installation Worksheet.”

1. **Server Migration from Certificate Server 1.x - Step 1.** Click No. If you are migrating from Certificate Server 1.x, you migrate the data to the original CMS Certificate Manager instance. The clones should not use migrated data.
2. **CA’s serial number range.** Specify the lowest serial number the CA should use in the “Starting serial number” field. If you only use one CA server, you can leave the “Ending serial number” blank to indicate no upper limit. If you use cloned CA servers to distribute load, you must specify an upper limit. For cloned CAs, be sure that the range of serial numbers does not overlap with any other CA server.
3. **Clone key and certificate materials - Step 1.** Click Yes to reuse the certificate and key material in the database files you copied from the original server. In the “Instance Name” field enter the instance ID of the original Certificate Manager instance. Select the token name where the keys and certificate are stored, and enter the token’s password.
4. **Clone key and certificate materials - Step 2.** On this screen you choose whether to reuse the SSL server certificate stored in the original Certificate Manager instance database or create a new one. You can only reuse the SSL server certificate if the clone uses the same hostname as the original server instance. To reuse the SSL server certificate, select Yes, enter the instance ID of the original server instance, select a token, and enter the token password.

If you do not or cannot reuse the SSL server certificate, the screens that follow will be those described in “SSL Certificate Configuration.”

If you do reuse the SSL server certificate, the next screen will be the one described in “Single Signon Configuration.”

SSL Certificate Configuration

SSL Server Certificate from the Local CA

The following screens allow you to generate an SSL Server Certificate signed with the local CA signing certificate.

1. **Key-Pair Information for SSL Server Certificate.** The token you select is used to store the SSL server certificate and key pair. If you have not previously initialized the token's password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for SSL Server Certificate.** The values you enter here identify the SSL server certificate. The CN must be the host name of the machine on which the server is running. You must also enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **Validity Period for SSL Server Certificate.** The validity period for the SSL server certificate determines how soon you will have to renew the certificate. Enter the validity period, then click Next.
5. **Certificate Extensions for SSL Server Certificate.** The default settings should work for most deployments. If necessary, you can add additional extensions by pasting the base-64 encoding for each extension in the space provided on this screen. For more information about extensions, see Appendix C, “Certificate Extensions.” Click Next to continue.

6. **SSL Server Certificate Creation.** This information screen tells you that the configuration wizard has all the required information to generate a key pair and its corresponding certificate. Click Next to generate the certificate.

To complete your configuration of this CMS instance, see “Single Signon Configuration” on page 224.

SSL Server Certificate from a Remote CA

The following screens allow you to configure an SSL Server Certificate issued by the subordinate or remote CA.

1. **Key-Pair Information for SSL Server Certificate.** The token you select is used to store the SSL server certificate and key pair. If you have not previously initialized the token’s password, you must do so in this screen. Specify the key type and length, then click Next to continue.
2. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature: SHA-1, MD2, or MD5. Click Next to continue.
3. **Subject Name for SSL Server Certificate.** The values you enter here identify the CA signing certificate. The CN must be the host name of the machine on which the server is running. You must also enter the Organization, such as the name of your company. The Organization is required because its absence causes Netscape Communicator 4.x to crash. Enter the values for the subject DN components here, then click Next.
4. **SSL Server Certificate Request Creation.** This informational screen tells you that the wizard has all the information required to generate the key pair and certificate request. Click Next to generate the request.
5. **Submission of Request:** Choose whether you want to submit the request manually or send the request to a remote CMS server automatically.
For automatic enrollment, select “Send the request to a remote CMS now,” enter the hostname and end-entity port number, and select whether this end-entity port uses SSL. Click Next to submit the request, then continue with step .

Follow these steps to submit your certificate request manually. Click Next when you are ready to proceed to the next screen.

Stage 2: Using the Installation Wizard

- Highlight the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) and copy it to the clipboard.

In addition to the copy on the clipboard, the screen informs you that the certificate request has been saved to a file. You can use either the copy on the clipboard or the copy in the file to transfer your request to the CA that will issue the subordinate CA's signing certificate.

If you are submitting your request to a third-party CA, follow the instructions provided by that CA. If you are submitting your request to a CMS Certificate Manager, follow these steps:

- Go to the end-entity URL for the Certificate Manager that will issue the subordinate CA's signing certificate. For example, if you assigned the port number 17006 to the non-SSL end-entity port for your root CA, you would go to the URL `http://hostname.17006` to bring up the Certificate Manager page for end entities.
 - Click Manual under the Certificate Manager Enrollment heading in the left-hand frame.
 - Paste the request from the clipboard into the field labeled PKCS #10 Request and fill in any other required information.
 - Click Submit.
 - A Certificate Manager agent can follow the instructions in Step 7 to issue the certificate.
6. **Certificate Request Result.** This screen confirms that the request has been submitted. You can use the request ID provided to retrieve the certificate from the end-entity port once it has been issued. A Certificate Manager agent can follow the instructions in the next step to issue the certificate.
 7. **A Certificate Manager agent issues the certificate.** After you submit the request, the agent for the Certificate Manager to which you submitted the request must approve it. For example, if you are the agent, go to the Agent Services page for the Certificate Manager (using the same computer where you got your agent certificate), choose Certificate Manager Agent Services, and follow these steps:

- Select List Requests, then click Show Pending Requests and click Find. The pending request list is displayed.
- Locate your request, click Details to see it, then scroll down to the bottom of the form and click Do It.
- After the certificate is generated, click Show Certificate.
- When the certificate is displayed, scroll down to the base-64 encoded version of the certificate, highlight all the text (including -----BEGIN CERTIFICATE ----- and -----END CERTIFICATE-----), and copy it to the clipboard.

You can either paste the encoded certificate into a file or paste it directly into the Installation Wizard, as described in Step 9 below.

8. **SSL Server Certificate Installation.** Click Yes to install the certificate now, or click No to install it at another time. The default is No. When you choose the default, you will continue with the configuration, and you'll be asked at the end of the configuration if you want to install the certificate at that point.

If you have submitted your request to a third-party CA or to a remote Certificate Manager, you may have to wait days or weeks before you receive the certificate. In this case, you should click No, continue as far as you can with the configuration, and resume after you receive the certificate. For a description of the screens that follow if you choose this option, see “Single Signon Configuration” on page 224.

You should click Yes only if you have received the base-64 encoded certificate and are ready to install it. Click Next to continue.

9. **Location of Certificate.** Enter the location of the file in which the encoded certificate is located, or paste in a base-64 encoded certificate including header and footer in the text area provided. Click Next to continue.
10. **Certificate Details.** This informational screen displays the certificate so you can inspect its contents. Click Next to continue.
11. **Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. Follow these steps to do so:

- Go to the end-entity URL for the Certificate Manager that issued the SSL server certificate, select the Retrieval tab, then choose Import CA Certificate Chain.
- Select “Display the CA certificate chain in PKCS#7 for importing into a server,” then click Submit.
- Copy the certificate to the clipboard.
- Return to the Installation Wizard.

12. Paste the certificate chain into the text box, then click Next.

To complete your configuration of this CMS instance, see “Single Signon Configuration” (the next section).

Single Signon Configuration

The following are the final screens of the Installation Wizard.

1. **Create Single Signon Password.** The single signon password simplifies the way you subsequently sign on to CMS by storing the passwords for the internal database, tokens, and LDAP publishing. Each time you log on, you’re only required to enter this single password.
Enter the single signon password, then click Next to continue.
2. **Configuration Status.** This screen should indicate that your configuration has been successful. Click Done to exit the wizard.

You have now completed your configuration of this CMS instance. For information on creating additional instances in the same server root directory, see “Stage 4: Creating Additional Instances or Certificate Manager Clones” on page 229.

For information on creating the first agent for the managers in this instance, see “Administrator/Agent Certificate Enrollment” on page 226.

Additional Steps

Each of the following screens may be displayed at different points in the Installation Wizard, depending upon your actions taken thus far and your path through the installation.

- **Get Certificates to be Installed.** If you have to wait for one or more certificates from a remote CA that you need to complete configuration, this screen is displayed when you complete the Installation Wizard. Click Done to end this session with the wizard. After you receive your certificates, restart the wizard and it will guide you through the steps required to install them.
- **Certificate Manager Token Password.** This screen is displayed when you re-enter the Installation Wizard when configuring the Certificate Manager. This token password is required for identification until you complete the installation, and will thus be stored in the single signon cache.
- **Registration Manager Token Password.** This screen is displayed when you re-enter the Installation Wizard when configuring the Registration Manager. This token password is required for identification until you complete the installation, and will then be stored in the single signon cache.
- **SSL Server Certificate Password.** This screen is displayed when you re-enter the Installation Wizard when configuring the SSL Server Certificate. This token password is required for identification until you complete the installation, and will then be stored in the single signon cache.
- **Data Recovery Manager Token Password.** This screen is displayed when you re-enter the Installation Wizard when configuring the Data Recovery Manager. This token password is required for identification until you complete the installation, and will then be stored in the single signon cache.
- **Internal Token Password.** This screen is displayed when you re-enter the Installation Wizard after configuring the internal token. This token password is required for identification until you complete the installation, and will then be stored in the single signon cache.

Administrator/Agent Certificate Enrollment

Immediately after installing any Certificate Management System instance, the administrator must enroll for the initial administrator/agent certificate. This is the first user certificate that Certificate Management System issues.

The initial user is both an administrator and an agent. This person can create additional agents with the appropriate user privileges and issue them certificates. Since there is no agent yet to approve the request, a special enrollment form allows you to get this first certificate automatically.

After you submit this initial Administrator/Agent Certificate Enrollment form, it is automatically disabled, so that no one else can acquire a certificate without agent approval or some form of automated authentication. The system automatically adds the initial user to the list of agents.

To enroll for the first agent certificate, you should be working at the computer you intend to use as the agent, so that the new certificate will be installed in the browser you will be using to access the Agent Services pages. Follow these steps:

1. Open a web browser window.
2. Go to the URL for the SSL agent port.

By default, this is a URL of the following form:

```
https://<hostname>:<agent_port_number>
```

- For <hostname>, provide the fully qualified domain name of the machine on which Certificate Management System is installed; for example, `mymachine.mydomain.com`.
- The <agent_port_number> is the TCP port specified during installation for agent communications over SSL.

The first time you access this port, the system opens the Administrator/Agent Certificate Enrollment form.

Because you have accessed an SSL port, Certificate Management System presents its server SSL certificate to your browser for authentication. This is the server SSL certificate that you created during installation. Because you just created it, it is not on your browser's list of trusted certificates. Before

you see the Administrator/Agent Certificate Enrollment form, a series of dialog boxes appears that lets you add the CMS server certificate to your list of trusted certificates.

3. Complete the dialog boxes as instructed (the exact procedure depends on the browser you are using).
4. In the Administrator/Agent Certificate Enrollment form, enroll for a client SSL certificate as the system's first privileged user by entering the following information:

Authentication Information

User ID: The ID you entered for the CMS administrator during installation.

Password: The password you specified for the CMS administrator during installation.

Subject Name

The subject name is the distinguished name (DN) that identifies the certified owner of the certificate.

Full name: Name of administrator/agent

Login name: User ID of administrator/agent

Email address: Email address of administrator/agent

Organization unit: Name of the organization unit to which the administrator/agent belongs

Organization: Name of the company or organization the administrator/agent works for.

Country: Two-letter code for the administrator/agent's country.

User's Key Length Information

Key Length: The length of the private key that will be generated by your browser. This key corresponds to the public key that is part of the administrator/agent certificate.

Note that the validity period of this initial agent certificate is hard-coded as one year.

5. Click Submit.
6. Follow the instructions your browser presents as it generates a key pair.
7. If authentication is successful, the new certificate will be imported into your browser, and you will be given an opportunity to make a backup copy.

Now you have a client authentication certificate in the name you specified. This special user, who was named as the initial administrator for Certificate Management System during installation, has been automatically designated as the first agent. This certificate allows you to access the Agent Services pages. As an agent, you can approve enrollment requests and start issuing new certificates. To access the CMS windows in Netscape Console, you use User ID that you specified for the certificate and the corresponding password—both of which must correspond to the values you specified for the CMS administrator during installation.

Important After you submit the initial Administrative Enrollment form, it is no longer available from the agent port. If something goes wrong and you are unable to obtain the administrator/agent certificate, you must reset a parameter in the configuration file to make the initial administrative enrollment form available again. Follow these steps:

1. In the left frame of Netscape Console, open the CMS instance for which you want to display the Administrator/Agent Certificate Enrollment form.
The server requests the password for the CMS administrator.
2. Click the icon labeled Stop the Server.
3. Go to the directory `<server_root>/<instance_ID>/config`, open the file `CMS.cfg` in a text editor, and find the following line:
`agentGateway.enableAdminEnroll=false`
4. Change `false` to `true`, and save the file.
5. Start the server from the CMS window where you stopped it. (Alternatively, right-click on the name of the instance in the left frame and choose Start Server.) At this point, the server asks you for the single signon password you specified during installation.
6. The next time you access the SSL agent port, the Administrator/Agent Certificate Enrollment form will be available again.

Stage 3: Further Configuration Options

When you have completed the initial configuration and installation of a CMS instance, you use the CMS window for that instance within Netscape Console to further configure the system as necessary. For example, you may want to configure LDAP publishing, authentication modules, and policy modules, and customize end-entity forms and other aspects of the system's operation.

For detailed information about the many CMS configuration options available, see *Netscape Certificate Management System Administrator's Guide*.

Stage 4: Creating Additional Instances or Certificate Manager Clones

After the initial installation, you can use the Administration Server Console to create additional instances of Certificate Management System in the same server root directory. Use the Certificate Management System Console and the Installation Wizard to configure any new instances.

To create an additional instance of Certificate Management System:

1. Start Netscape Console and log in with the administrator password.
The main window of Netscape Console appears.
2. In the navigation tree at the left, open your computer, then open Server Group.
3. Right-click Server Group. Choose "instance of" from the menu, then choose Certificate Management System from the submenu.
4. In the resulting dialog box, specify the unique identifier for the new instance.
5. If the new instance is a Certificate Manager clone, copy the `key3.db` and `cert7.db` files from the original server's `config` directory to the new instance's `config` directory.

The config directory is

```
<server_root>/<instance_ID>/config
```

6. Configure the new instance using the Installation Wizard and Console, as you did for the first instance.

For more information about installing multiple CMS instances, see “Chapter 4, Installing and Uninstalling CMS Instances,” in *Netscape Certificate Management System Administrator’s Guide*.

First Agent for an Additional CMS Instance

When you have finished setting up an additional CMS instance, you need to create at least one agent for that instance. If the new instance includes a Certificate Manager, you can create the administrator/agent as described in “Administrator/Agent Certificate Enrollment” on page 226 as you did for the first instance in the server root. If the new instance does not include a Certificate Manager (that is, it contains a Registration Manager, a Data Recovery Manager, or a Registration Manager and Data Recovery Manager), you need to create a new agent as described in this section.

To create the first agent for an additional CMS instance that doesn’t include a Certificate Manager, you use the CMS window for the new instance in Netscape Console to add a new user with agent privileges. You can then either associate the original administrator/agent certificate with the new user (if the original administrator and the new agent are the same person), or you can issue a new agent certificate for each new manager agent and associate the new certificate with the new user.

In either case, you create the new user and associate a certificate with that user as follows. These instructions assume that you have already copied the base-64 encoded certificate (whether the original administrator/agent certificate or a new agent certificate) to the clipboard.

1. Start Netscape Console and log in with the administrator password.

The main window of Netscape Console appears.

2. In the navigation tree at the left, open your computer, then open Server Group.

3. Select the name of the new instance.
4. In the Netscape Certificate Management System panel at the right, click Open.
5. Log in as the CMS administrator.
6. Select the Configuration tab and select “Users and Groups” in the navigation tree.
7. On the Users page, click Add, and in the dialog box that appears, provide the following information:

User ID: ID for the new user, for example, RegMgr

Full name: Name of the new user, for example, Registration Manager Agent

Password: Password for the new user

E-Mail: Email address for the new user

Group: Select the appropriate agent group.

8. Click OK.
The new user appears in the list of users.
9. Select the new user’s entry in the list of users.
10. Click Certificates.
11. In the Manage User Certificates dialog, click Import.
12. In the Import Certificate dialog box, click Paste from Clipboard and click OK.
13. In the Manage User Certificate Dialog box, click Done.

You have now designated an agent for the specified manager. You can now present the certificate you installed for that agent to access the Agent Services pages for that manager in the new instance.

For more information about setting up and managing agents, see “Chapter 7, Managing Privileged Users and Groups,” in *Netscape Certificate Management System Administrator’s Guide*.

Stage 4: Creating Additional Instances or Certificate Manager Clones

A

Migrating from Certificate Server 1.x

This appendix explains how to use the Migration Tool that comes with Netscape Certificate Management System. This executable command-line script extracts database contents (as stored in the Informix database) and certificate/key data (as stored in flat-file DBM databases) from Certificate Server 1.x and places the data in three platform-independent files that can be transferred by diskette, tape, or FTP to a Certificate Management System 4.x installation area for importing into the new system.

This appendix has the following sections:

- Using the Migration Tool (page 233)
- Importing Data to New Databases (page 244)
- Hardware, Operating System, and Version Support (page 245)

Using the Migration Tool

The Migration Tool is installed when you install CMS. You run the tool after installation and before you run the Installation Wizard to configure your instance (the Installation Wizard will ask for the files generated by the Migration Tool).

Use the `migrate` command appropriate for the platform on which your Certificate Server 1.x is running. You will probably need to copy the `migrate` executable from the machine where you installed Certificate Management

Using the Migration Tool

System 4.x to the Certificate Server machine using FTP, a diskette, or a shared network disk. The `migrate` executables are located in platform-specific directories:

```
<server_root>/bin/cert/tools/migrate/<platform>
```

To begin the migration process, you enter the `migrate` command and its arguments in a command shell, provide database information and passwords, and monitor the output in the shell window.

Before running the Migration Tool, make sure that the following conditions have been met:

- You have processed all pending database requests.
- You have stopped the instance of Certificate Server 1.x that you will migrate.
- The global variable `INFORMIXSQLHOSTS` is undefined.
- You are able to run the Migration Tool as a user with permission to read `magnus.conf`, `CASigningKey.db`, `ServerCert.db`, and `ServerKey.db` in the `config/` directory of the Certificate Server 1.x instance.
- The user running the Migration Tool has read and write permissions in the output directory.
- On AIX, the directory containing Informix is `/usr/informix` (the default). If Informix has not been installed in `/usr/informix`, you can make a symbolic link from there to the actual installation directory.
- On Windows NT, the location of the Informix DLLs is in the `PATH` environment variable. If not, the Migration Tool will fail because it cannot find `ISQLT07C.DLL`. For example, if the Informix installation directory is `c:\informix`, `PATH` must include `c:\informix\bin`.

To terminate the execution of the tool at any time, type Control-C on Unix or Control-Break on Windows NT.

Command-Line Syntax

Use the following command in a Unix or DOS command shell:

```
migrate certsrvroot=<directory> outputdir=<directory>
      dbrootdir=<directory> servername=<name> help
```

Warning The `certsrvroot` directory must be the directory that contains the `config` directory for the server instance. For example, if the Certificate Server 1.x instance `config` directory is `/opt/ns-home/cert-server/config`, use `/opt/ns-home/cert-server` as the `certsrvroot` parameter.

Arguments

The `migrate` command takes the following arguments:

<code>certsrvroot</code>	The <i>server root/server identifier</i> , as defined in the Certificate Server 1.x installation documentation. This directory must point to the root of the server instance that will be migrated.
<code>outputdir</code>	A directory that does not currently exist but that will be created and populated with the results of the <code>migrate</code> command.
<code>dbrootdir</code>	The directory in which the Informix database resides—by default, <code>/usr/informix</code> on Unix platforms and <code>c:\informix</code> on Windows NT.
<code>servername</code>	The name of the Informix database server from which to extract data. By default, the database server name is <code>ifmx_online</code> .
<code>help</code>	Optional. Prints a usage string for the command to the command shell.

The Migration Process

There are two stages to the migration process. The Migration Tool first connects to the Informix database, extracts database records, and writes them out to an LDIF file in the specified output directory. It then opens the flat files containing the key and certificate information, and transfers that information to ASCII files in the output directory.

When you issue the `migrate` command, the tool prompts you to enter login information and passwords for the Certificate Server 1.x Informix database, and performs the first phase of the migration. After migrating the database records,

the tool prompts for a transport password to protect the exported data, the password for the existing CA Signing key, and the password for the existing SSL Server key. Using these passwords, the tool migrates the flat file information.

Entering Informix Database Login Information

The tool prompts for the Informix database name and login:

```
STARTING Informix Data Base migration...
  ENTER the existing "Certificate Server 1.x" Informix Data Base
Information:
  Informix Data Base name [cmsdb]: <data base name>
  Informix Data Base login name [cmsdbusr]: <db administrative login>
  Informix Data Base password: *****
```

The database name is the name of the Informix database in which the Certificate Server tables `t_cert_record`, `t_seq_num_gen`, and `t_iss_auth_prop` reside.

The default database name and user name are those used for a standard installation of Certificate Server 1.x. Unless you changed the names during the installation, you can accept the defaults by pressing Enter at the prompt.

When you enter the login information, the Migration Tool connects to the database, extracts records, and writes the extracted records to files in the directory specified by the `outputdir` argument. Text such as the following appears:

```
CONNECTED to Informix Data Base...
MIGRATING certificate records...
  Extracted 1 cert record(s)
  Extracted 2 cert record(s)
  ...
MIGRATING last serial number...
MIGRATING certificate revocation list...
SUCCESSFULLY completed Informix Data Base migration;
GENERATED "database_add.ldif" and "database_mod.ldif" files!
```

If the migration of data from the Informix database fails, the tool describes the Informix error and asks whether to continue or exit. If you choose to continue, certificate and key migration proceeds. See Informix manuals to find the source of the problem, fix it, then run the tool again to extract the Informix data.

For example, if you run the Migration Tool on the wrong computer, and it cannot find the Informix database, you might see the following message referring to the Informix error 461 (file not found):

```
Starting Database migration...
CMS -- UNLOGGABLE FAILURE: [VENDORLIB] Vendor Library Error:
Cannot open file 'sql.iem'; Cannot open file 'os.iem' (-461)(4)

migrate: error: Could not connect to database!
Continue migration [y|n]: y
```

In this example, the user chooses to continue with the second phase of the migration, then go back and run the tool on the right machine to migrate the Informix database information to CMS.

Entering Key and Certificate Database Passwords

After the Migration Tool has migrated the records from the Informix database, it converts and exports the SSL server and CA signing certificate chains.

Next, the tool asks you to create a transport password. The transport password is used to encrypt keys as they are extracted from the key database, before they are written out to the `keyscerts.dat` file.

```
STARTING Certs and Keys Data Base migration...
SUCCESSFULLY dumped Server Certificate Chain!
SUCCESSFULLY dumped CA Signing Certificate Chain!
ENTER a new "CMS 4.x" Transport password
used to protect the private key material:
Transport password: *****
Verify Transport password: *****
```

If the transport password and the verify transport password entries are not the same, the following message appears, and the tool exits:

```
Transport and verify transport passwords are not the same
```

If the transport password does not conform to the two minimum quality rules, one of the following messages appears:

```
The transport password must be a minimum of 8 characters
The transport password must contain both alphabetic and numeric
characters
```

After this message, the password prompt appears so you can enter a new password.

Next, the tool will ask for the passwords needed to access the existing SSL server certificate and CA signing certificate:

```
ENTER the existing "Certificate Server 1.x" password
used to protect SSL Server Key data:
  SSL Server Key password: *****
SUCCESSFULLY dumped SSL Server Key!
ENTER the existing "Certificate Server 1.x" password
used to protect CA Signing Key data:
  CA Signing Key password: *****
```

If either password is incorrect, an error message appears and the tool exits. See “Exit Codes and Error Messages” on page 238.

When you enter the passwords, the Migration Tool opens and extracts information from the flat certificate and key database files, writing the information to an ASCII file in the directory specified by the `outputdir` argument. If the migration is successful, text such as the following appears:

```
SUCCESSFULLY dumped CA Signing Key!
SUCCESSFULLY completed Certs and Keys Data Base migration;
GENERATED "keycerts.dat" file!Starting Certs and Keys migration...
```

Exit Codes and Error Messages

When the Migration Tool exits, it provides an exit code and displays an explanatory message. If the data migration process is successful, the Migration Tool returns the code 0 and prints the success message. If an error occurs, the tool returns one of the error codes and prints an error message. In some cases, you will be asked if you want to continue when an error occurs. If you answer ‘no’, the tool exits with the status listed. Table A.1 describes the success and error exit codes and messages.

Table A.1 Exit codes and conditions

Exit codes	Exit message	Reason
0	Successfully created <i>outputdir</i> with Certificate Server 1.x data and the three output files	Indicates successful export of both the key material and database contents.
1	Invalid password; unable to export	One of the three passwords required to access the signing key, server key, or database was incorrect. Check your passwords and try running the Migration Tool again.
2	Could not connect to the Informix Data Base!	A problem occurred opening a connection to the DBMS. Check to make sure the DBMS is running. Use <code>onmonitor</code> or another Informix command to start the DBMS. In most cases this error occurs if a user name or password is incorrect.
3	<code>mkdir</code> of output directory failed -- Either parent directory does not exist OR Not enough disk space or inodes	The tool could not create the output directory. Make sure that the parent directory exists and that the user running the tool has write permission in that directory.
4	Transport and verify transport passwords are not the same	The strings entered for the password and then for verification were not the same.
5	ERROR: encrypting the SSL Server Key Material according to the new transport password. ERROR: can not encrypt the CA Signing Key material according to the new transport password.	The named key could not be encrypted using the transport password provided. The tool may have been unable to allocate memory during encryption or there may be a problem accessing the Certificate Server 1.x key database file.
6	Could not open [<i>outputdir</i> / <i>database_add.ldif</i> <i>outputdir</i> / <i>database_mod.ldif</i>] for write	The tool failed to create <code>ldif</code> files in the output directory. Possible permissions problem.
7	<i>failure_location</i> : Out of memory	The tool was unable to allocate more memory at the indicated location.
8	Base 64 Encoding of data failed: Unable to encode	Possible corrupt binary data was found. The tool could not proceed with base-64 encoding of binaries obtained from the database or from the flat database files.

Using the Migration Tool

Table A.1 Exit codes and conditions (Continued)

Exit codes	Exit message	Reason
9	could not open <i>full_pathname/magnus.conf</i>	The configuration file needed to find the locations of Certificate Server 1.x flat database files does not exist, cannot be read, or cannot be found.
15	ERROR: Writing key data	The tool could not write a key, certificate, or certificate chain to the <i>keyscerts.dat</i> file. Make sure that the user running the tool has write permission on <i>keyscerts.dat</i> and the output directory where it is stored.
16	Failed to open the Key Data Base: <i>filename</i> .	The tool could not open the named SSL Server key database file for reading. The user running the tool must have read access to the database file (<i>config/ServerKey.db</i> by default).
17	can not open CA Signing Key Data Base: <i>filename</i>	The tool could not open the CA Signing key database file for reading. The user running the tool must have read access to the database file (<i>config/CASigningKey.db</i> by default).
20	lost connection to Informix Data Base while migrating certificate revocation list.	The connection to the Informix database became invalid while the tool was attempting to migrate records. Check to make sure that the database server is still running.
21	Data Base table "t_iss_auth_prop" was not found. CRL migration skipped.	If the <i>t_iss_auth_prop</i> table is not found in the database, CRL migration cannot be completed.
22	Data Base table "t_cert_record" was not found, certificate records migration aborted.	The <i>t_cert_record</i> table is required for migration. If the database you have specified does not include this table, the migration cannot succeed.

Table A.1 Exit codes and conditions (Continued)

Exit codes	Exit message	Reason
23	Data Base table "t_seq_num_gen" was not found, CRL migration aborted.	If the t_seq_num_gen table is not found in the database, the tool cannot determine the new starting serial number for the 4.x installation.
24	No binary found in "t_iss_auth_prop", CRL migration skipped. or No entries were found in "t_iss_auth_prop", CRL migration aborted.	The t_iss_auth_prop table exists, but it either contains no data or the records it contains do not have data in the blob_val field that can be converted to a base-64 encoding.

Generated Files

When you run the Migration Tool, the files `database_add.ldif`, `database_mod.ldif`, and `keyscerts.dat` are generated and placed in the output directory (specified by the `outputdir` argument):

- `database_add.ldif`
Contents of the 1.x Informix database `t_certificate_record` table. The entries are in standard `ldif` format for use with the `ldapmodify -a` command.
- `database_mod.ldif`
Contents of the 1.x Informix database, from the `CRL` and `last_serial_number` tables. The entries are formatted for use with the `ldapmodify` command.
- `keyscerts.dat`
An ASCII file containing the private keys (RSA only), public-key certificates (X.509 v3 format) for the signing key and certificate, and the SSL key and certificate. The file contains the full certificate chains of these keys and certificates.

Temporary data structures (such as request queues or transactions in process) are not converted. Therefore, before running the Migration Tool, the database administrator should have processed all pending requests.

Using Output Files on Another Platform

This section only applies if you are migrating from any UNIX platform to Windows NT or from Windows NT to any UNIX platform.

The output files from the Migration Tool are all plain ASCII files. UNIX and Windows NT use different line breaks in ASCII files, so when you transfer the output from Windows NT to UNIX (or vice versa) you need to modify the files. ASCII files transferred from Windows NT to UNIX will have an extra Ctrl-M at the end of each line that needs to be removed before the Installation Wizard can import the data in the files. ASCII files transferred from UNIX to Windows NT may not have the Ctrl-M (the whole file will appear as one line in a text editor); you need to insert carriage returns manually to break the file into separate lines.

Most FTP clients will convert the line breaks in the ASCII file to the appropriate native control character, so consider using FTP to transfer the files to the Certificate Management System 4.x machine if it is available.

If you are unsure whether your files need to be converted, or if you have problems during the import phase of the Installation Wizard, convert your files according to the steps in Converting Output Files.

A simple way to convert your files is to use the stream editor `sed`. Any conversion involves one UNIX machine, and all UNIX machines should have `sed` installed. Therefore, the instructions should be followed when the files are on the UNIX machine (whether you are transferring to or from UNIX).

Converting Output Files

1. Move the files to new filenames

The output of the conversion will be overwrite any files with the original names:

```
# mv database_add.ldif database_add.ldif.orig
# mv database_mod.ldif database_mod.ldif.orig
# mv keyscerts.dat keyscerts.dat.orig
```

2. Use sed to convert the files to the appropriate format.

1. To convert from Windows NT to UNIX, remove the Ctrl-M line breaks:

```
# sed -e 's/^M//g' database_add.ldif.orig > database_add.ldif
# sed -e 's/^M//g' database_mod.ldif.orig > database_mod.ldif
# sed -e 's/^M//g' keyscerts.dat.orig > keyscerts.dat
```

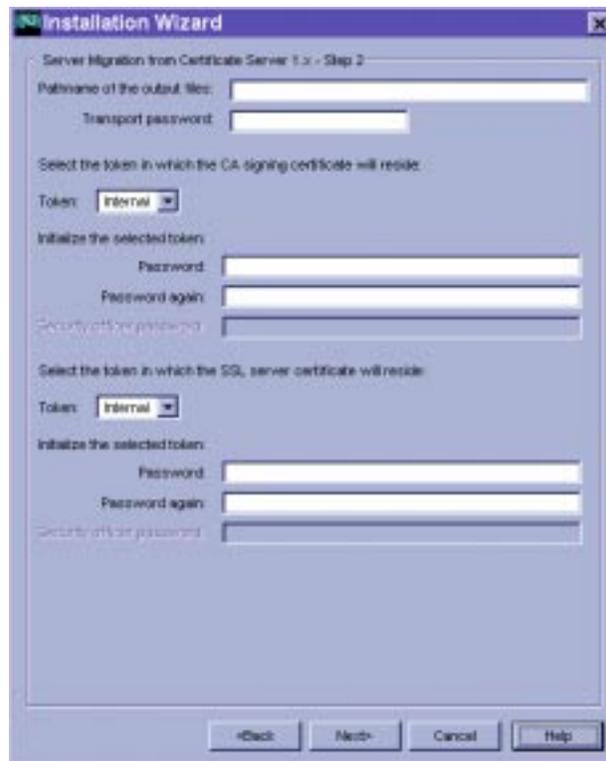
2. To convert from UNIX to Windows NT, add a Ctrl-M to each line break:

```
# sed -e 's/$/^M/g' database_add.ldif.orig > database_add.ldif
# sed -e 's/$/^M/g' database_mod.ldif.orig > database_mod.ldif
# sed -e 's/$/^M/g' keyscerts.dat.orig > keyscerts.dat
```

To get the ^M to show up in the shell commands, press Ctrl-V then Ctrl-M. If this does not work in your shell, run the command in the Bourne shell (/bin/sh).

Importing Data to New Databases

When you are installing Certificate Management System, the Installation Wizard offers you the choice of importing existing certificates and keys. If you select this option, the following screen appears.



Use this screen to enter the following information:

- The path to the three generated files. This is the directory you specified in the `outdir` argument to the `migrate` command.
- The transport password. This is the password you specified and confirmed during the second phase of the migration process.
- The required token passwords. Both of these are for the Certificate Management System installation. Depending on what else you have specified during the configuration, you may not need to provide all of the passwords here; the ones you do not need are inactive.

When you click Next, the Installation Wizard imports the needed certificates and keys that you have exported from Certificate Server 1.x. Depending on how you have configured the installation, the Wizard imports the server SSL certificate, the CA signing certificate, or both.

Once all of the data has been imported, your Certificate Management System 4.0 server will use the original CA Signing certificate and SSL Server certificate. If the new server is running on a different machine than the Certificate Server 1.x instance, you probably will want to issue a new SSL Server Certificate. See Chapter 8 “Keys and Certificates” in *Netscape Certificate Management System Administrator’s Guide* for instructions for replacing the SSL Server certificate.

The administrator’s certificate that was used to manage Certificate Server 1.x will no longer work as the administrator certificate. Once the Installation Wizard has finished, continue with “Administrator/Agent Certificate Enrollment” on page 226 to issue a new certificate for the administrator.

Hardware, Operating System, and Version Support

Netscape has tested the Migration Tool on the following Certificate Server 1.x platforms and operating systems:

- Hewlett-Packard PA, HP-UX 10.10
- IBM RS/6000, AIX 4.1, 4.2
- Sun Sparc, Solaris 2.4, 2.5.1
- Intel Pentium, Windows NT 3.51, 4.0

Hardware, Operating System, and Version Support

Certificate Extensions

This appendix explains both the standard certificate extensions defined by X.509 v3 and the extensions defined by Netscape that were used in versions of products released before X.509 v3 was finalized. It also provides recommendations for extensions to use with specific kinds of certificates, including both PKIX Part 1 recommendations and Netscape extensions that must be supported for compatibility with early versions of Netscape products.

This appendix contains the following sections:

- Introduction to Certificate Extensions (page 247)
- Recommendations for Extension Use (page 249)
- Standard X.509 v3 Certificate Extensions (page 253)
- Standard X.509 v3 CRL Extensions (page 275)
- Netscape-Defined Certificate Extensions (page 280)
- Adding Extensions in Certificate Management System (page 282)
- CA Certificates and Extension Interactions (page 282)

Introduction to Certificate Extensions

An X.509 v3 certificate contains an extensions field that permits any number of additional fields to be added to the certificate. Certificate extensions provide a way of adding information such as alternative subject names and usage

restrictions to certificates. Older versions of Netscape browsers and servers that were developed before PKIX part 1 standards were defined require Netscape-specific extensions.

The X.509 v1 certificate specification was originally designed to bind public keys to names in an X.500 directory. As certificates began to be used on the Internet and extranets, and directory lookups could not always be performed, problem areas such as the following emerged that were not foreseen in the original specification:

- **Trust.** The X.500 specification establishes trust by means of a strict directory hierarchy. By contrast, Internet and extranet deployments frequently involve distributed trust models that do not conform to the hierarchical X.500 approach.
- **Certificate use.** Some organizations may wish to restrict the use of certificates for policy reasons. For example, some certificates may be restricted to client authentication only.
- **Multiple certificates.** It's not uncommon for certificate users to possess multiple certificates with identical subject names but different key material. In this case, it's necessary to identify which key and certificate should be used for what purpose.
- **Alternate names.** For some purposes, it is useful to have alternative subject names that are also bound to the public key in the certificate.
- **Additional attributes.** Some organizations may find it convenient to store additional information in certificates, for example for situations in which it's not possible to look up information in a directory.
- **Relationship with CA.** When certificate chaining involves intermediate CAs, it is useful to have information about the relationships among CAs embedded in their certificates.
- **CRL checking.** Since it's not always possible to check a certificate's revocation status against a directory or with the original certificate authority, it is useful for certificates to include information about where to check CRLs.

Eventually, the X.509 v3 specification addressed many of these issues by defining a general format for certificate extensions and specifying a number of standard extensions. The X.509 v3 certificate format also allows communities to define private extensions to carry information unique to those communities.

Before the X.509 v3 standard was finalized, Netscape and other companies had to address some of the most pressing issues listed above with their own extension definitions. Therefore, to maintain compatibility with older versions of browsers that were released before the X.509 v3 specification was finalized, certain kinds of certificates should include some of the Netscape extensions. For details, see Recommendations for Extension Use.

The X.500 and X.509 specifications are controlled by the International Telecommunication Union (ITU), an international organization that primarily serves large telecom companies, government organizations, and other entities concerned with the international telecommunications network. The Internet Engineering Task Force (IETF), which controls many of the standards that underlie the Internet, is currently developing public-key infrastructure X.509 (PKIX) standards. These proposed standards further refine the X.509 v3 approach to extensions for use on the Internet. The recommendations for certificates and CRLs have reached proposed standard status, and can be retrieved from <http://www.ietf.org/rfc/rfc2459.txt>. This document is often referred to as PKIX Part 1.

Recommendations for Extension Use

Most deployments will use some or all of these extensions:

authorityKeyIdentifier. Identifies the public key corresponding to the private key used to sign a certificate.

basicConstraints. Identifies CA certificates and optionally specifies a maximum certificate chain path length.

cRLDistributionPoints. Defines how CRL information for the certificate is to be obtained.

extKeyUsage. Indicates purpose or purposes for which the certificate may be used, either in addition to or instead of the purposes indicated by the keyUsage extension.

keyUsage. Indicates the purpose or purposes for which the public key certified by the certificate may be used.

netscape-cert-type. Indicates the purpose or purposes for which the certificate may be used. Required only for compatibility with some Netscape products that were released before X.509 v3 was finalized.

subjectAltName. Specifies one or more alternative names for the identity bound by the CA to the certified public key.

subjectKeyIdentifier. Identifies the public key certified by the certificate.

These extensions, plus others, are described in detail in later sections of this appendix. Additional extensions may be useful for a variety of purposes. However, the extensions listed above are either required or recommended for various kinds of certificates issued by Certificate Management System.

Table B.1 summarizes guidelines for using these extensions. The table provides a summary only. Each extension is explained in detail later in the Appendix. Keep the following in mind as you use the table:

- Using certificate extensions incorrectly can lead to severe deployment problems. Make sure you have thoroughly analyzed your deployment needs and completely understand the purpose of each extension you want to use before adding them to certificates.
- Unless otherwise noted in Table B.1, the extensions indicated should be included with certificates of each type to ensure compatibility with both PKIX Part 1 and with future Netscape or iPlanet products.
- Extensions marked “required” must be supported for some existing Netscape or Microsoft products or for other reasons explained in the extension descriptions that follow.

Table B.1 Recommendations for Use of Certificate Extensions with CMS

Certificate type	CA root	Intermediate CA	Issued certificate
SSL client certificate	<p>authorityKeyIdentifier</p> <p>basicConstraints: true (required)</p> <p>extKeyUsage: client auth</p> <p>keyUsage: keyCertSign, cRLSign</p> <p>netscape-cert-type: SSL CA (if extension exists, bit must be set)</p> <p>subjectKeyIdentifier</p>	<p>authorityKeyIdentifier</p> <p>basicConstraints: true (required)</p> <p>cRLDistributionPoints</p> <p>extKeyUsage: client auth</p> <p>keyUsage: keyCertSign, cRLSign</p> <p>netscape-cert-type: SSL CA (required for client authentication with some Netscape servers)</p> <p>subjectKeyIdentifier</p>	<p>authorityKeyIdentifier</p> <p>cRLDistributionPoints</p> <p>extKeyUsage: client auth</p> <p>keyUsage: digitalSignature</p> <p>netscape-cert-type: SSL client (if extension exists, bit must be set; otherwise, not required)</p> <p>subjectKeyIdentifier</p>
S/MIME client certificate (single key pair)	<p>authorityKeyIdentifier</p> <p>extKeyUsage: Email</p> <p>keyUsage: keyCertSign, cRLSign</p> <p>netscape-cert-type: S/MIME CA (if extension exists, bit must be set)</p> <p>subjectKeyIdentifier</p>	<p>authorityKeyIdentifier</p> <p>cRLDistributionPoints</p> <p>extKeyUsage: Email</p> <p>keyUsage: keyCertSign, cRLSign</p> <p>netscape-cert-type: S/MIME CA (if extension exists, bit must be set)</p> <p>subjectKeyIdentifier</p>	<p>authorityKeyIdentifier</p> <p>cRLDistributionPoints</p> <p>extKeyUsage: Email</p> <p>keyUsage: digitalSignature</p> <p>netscape-cert-type: S/MIME (if extension exists, bit must be set)</p> <p>subjectAltName</p> <p>subjectKeyIdentifier</p>

Recommendations for Extension Use

Table B.1 Recommendations for Use of Certificate Extensions with CMS (Continued)

Certificate type	CA root	Intermediate CA	Issued certificate
S/MIME client certificate (dual key pair)	<p>authorityKeyIdentifier</p> <p>extKeyUsage: Email</p> <p>keyUsage: keyCertSign, cRLSign</p> <p>subjectKeyIdentifier</p>	<p>authorityKeyIdentifier</p> <p>cRLDistributionPoints</p> <p>extKeyUsage: Email</p> <p>keyUsage: keyCertSign, cRLSign</p> <p>subjectKeyIdentifier</p>	<p>authorityKeyIdentifier</p> <p>cRLDistributionPoints</p> <p>extKeyUsage: Email</p> <p>keyUsage, signing certificate: digitalSignature (required)</p> <p>keyUsage, encryption certificate: keyEncipherment (required)</p> <p>subjectAltName</p> <p>subjectKeyIdentifier</p>
SSL server certificate	<p>authorityKeyIdentifier</p> <p>extKeyUsage: Server Auth (recommended), Microsoft SGC and Netscape SGC (required for step-up)</p> <p>keyUsage: keyCertSign, cRLSign</p> <p>netscape-cert-type: SSL CA (if extension exists, bit must be set)</p> <p>subjectKeyIdentifier</p>	<p>authorityKeyIdentifier</p> <p>cRLDistributionPoints</p> <p>extKeyUsage: Server Auth (recommended), Microsoft SGC and Netscape SGC (required for step-up)</p> <p>keyUsage: keyCertSign, cRLSign</p> <p>netscape-cert-type: SSL CA (if extension exists, bit must be set)</p> <p>subjectKeyIdentifier</p>	<p>authorityKeyIdentifier</p> <p>cRLDistributionPoints</p> <p>extKeyUsage: Server Auth (recommended), Microsoft SGC and Netscape SGC (required for step-up)</p> <p>keyUsage: keyEncipherment</p> <p>netscape-cert-type: SSL Client, SSL Server (required for some Netscape servers)</p> <p>subjectAltName</p> <p>subjectKeyIdentifier</p>

Table B.1 Recommendations for Use of Certificate Extensions with CMS (Continued)

Certificate type	CA root	Intermediate CA	Issued certificate
Object signing/ Authenticode certificate	authorityKeyIdentifier extKeyUsage: Code Signing (required for Authenticode) keyUsage: keyCertSign, cRLSign netscape-cert-type: Object-signing CA (required for Object Signing) subjectKeyIdentifier	authorityKeyIdentifier cRLDistributionPoints extKeyUsage: Code Signing (required for Authenticode) keyUsage: keyCertSign, cRLSign netscape-cert-type: Object-signing CA (required for Object Signing) subjectKeyIdentifier	authorityKeyIdentifier cRLDistributionPoints extKeyUsage: Code Signing (required for Authenticode) keyUsage: digitalSignature netscape-cert-type: Object-signing (required for Object Signing) subjectAltName subjectKeyIdentifier

Standard X.509 v3 Certificate Extensions

This section summarizes the extension types that are defined as part of the Internet X.509 Version 3 standard, as of September 1998, and indicates which types are recommended by the PKIX working group.

This section summarizes important information about each certificate. For complete details, see both the X.509 v3 standard (available from the ITU) and the Internet X.509 Public Key Infrastructure - Certificate and CRL Profile (RFC 2459), available at <http://www.ietf.org/rfc/rfc2459.txt>. The descriptions of extensions reference the RFC and section number of the standard draft that discusses the extension; the object identifier (OID) for each extension is also provided.

Each extension in a certificate can be designated as critical or noncritical. A certificate-using system, such as browser software, must reject the certificate if it encounters a critical extension it does not recognize; however, a noncritical extension can be ignored if it is not recognized.

The descriptions below contain recommendations for use of the extension from Netscape and Microsoft. The Microsoft recommendations were taken from "Structuring X.509 Certificates for Use with Microsoft Products" at <http://www.microsoft.com/security/tech/certificates/structuring.asp>, dated December 4, 1997.

Certificate Management System (CMS) version support is listed for each extension. "Supported" means that the indicated version of CMS ships with built-in support for the extension via a policy plug-in. "Not supported" means that the indicated version of CMS does not ship a policy plug-in for the extension (although the extension can be used if a custom plug-in is written).

These are the standard X.509 v3 extensions described in the sections that follow:

- authorityInfoAccess (page 254)
- authorityKeyIdentifier (page 256)
- basicConstraints (page 257)
- certificatePolicies (page 258)
- cRLDistributionPoints (page 259)
- extKeyUsage (page 261)
- issuerAltName (page 263)
- keyUsage (page 264)
- nameConstraints (page 267)
- OCSPNocheck (page 268)
- policyConstraints (page 269)
- policyMappings (page 270)
- privateKeyUsagePeriod (page 271)
- subjectAltName (page 256)
- subjectDirectoryAttributes (page 273)
- subjectKeyIdentifier (page 274)

authorityInfoAccess

OID

1.3.6.1.5.5.7.1.1

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.2.1

Criticality

This extension must be noncritical.

Discussion

The Authority Information Access extension indicates how and where to access information about the issuer of the certificate. The extension contains an `accessMethod` and an `accessLocation` field. The `accessMethod` specifies (by an OID) the type and format of information about the issuer found at the `accessLocation`.

PKIX Part 1 defines one `accessMethod` (`id-ad-caIssuers`) to get a list of CAs that have issued certificates higher in the CA chain than the issuer of the certificate using the extension. The `accessLocation` field then typically contains a URL indicating the location and protocol (LDAP, HTTP, FTP) used to retrieve the list.

The Online Certificate Status Protocol (RFC 2560), available at <http://www.ietf.org/rfc/rfc2560.txt>, defines an `accessMethod` (`id-ad-ocsp`) for using OCSP to verify certificates. The `accessLocation` field then contains a URL indicating the location and protocol used to access an OCSP responder that can validate the certificate.

CMS Version Support

Refer to “Authority Information Access Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator's Guide*.

- **CMS 4.1:** Not supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape recommends that you add this extension with `id-ad-ocsp` and the URL for an OCSP responder to every certificate that can be verified using OCSP.

OCSP signing certificates and CA signing certificates should only use the `authorityInfoAccess` extension to point to an OCSP responder if that responder has been configured to verify them. For example, if there is a hierarchy of responders, a subordinate responder may point to its parent for verification. If a CA signing certificate points to an OCSP responder, that responder's signing certificate should be signed by a different CA (for example, the CA that issued the CA signing certificate in question).

Microsoft Recommendation

Microsoft products do not currently use on-line revocation checking.

authorityKeyIdentifier

OID

2.5.29.35

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.1

Criticality

This extension is always noncritical and is always evaluated.

Discussion

The Authority Key Identifier extension identifies the public key corresponding to the private key used to sign a certificate. This extension is useful when an issuer has multiple signing keys (for example, due to CA certificate renewal).

The extension consists of either or both of the following:

- an explicit key identifier (`keyIdentifier` field)
- an issuer (`authorityCertIssuer` field) and serial number (`authorityCertSerialNumber` field) identifying a certificate

If the `keyIdentifier` field exists, then it is used to select the certificate with a matching `subjectKeyIdentifier` extension. If the `authorityCertIssuer` and `authorityCertSerialNumber` fields are present, then they are used to identify the correct certificate by issuer and serial number.

If this extension is not present, then the issuer name alone is used to identify the issuer certificate.

PKIX Part 1 requires this extension for all certificates except self-signed root CA certificates. Where a key identifier has not been previously established, PKIX recommends that the `authorityCertIssuer` and `authorityCertSerialNumber` fields be specified. These fields permit construction of a complete certificate chain by matching the `SubjectName` and `CertificateSerialNumber` fields in the issuer's certificate against the `authorityCertIssuer` and `authorityCertSerialNumber` in the `AuthorityKeyIdentifier` extension of the subject certificate.

CMS Version Support

Refer to "Authority Key Identifier Extension Policy" in Chapter 18 of *Netscape Certificate Management System Administrator's Guide*.

- **CMS 4.1:** Supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape recommends that this extension be present in all certificates and that the `authorityCertIssuer` and `authorityCertSerialNumber` fields be specified. This extension is not supported by Navigator 3.x, but its presence in a certificate won't interfere with Navigator 3.x.

Microsoft Recommendation

Microsoft recommends that this extension be present in all certificates and that the `authorityCertIssuer` and `authorityCertSerialNumber` fields be specified.

basicConstraints

OID

2.5.29.19

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.10

Criticality

PKIX Part 1 requires that this extension be marked critical. This extension is evaluated regardless of its criticality.

Discussion

This extension is used during the certificate chain verification process to identify CA certificates and to apply certificate chain path length constraints. The `cA` component should be set to true for all CA certificates. PKIX recommends that this extension should not appear in end-entity certificates.

If the `pathLenConstraint` component is present, its value must be greater than the number of CA certificates that have been processed so far (starting with the end-entity certificate and moving up the chain). If `pathLenConstraint` is omitted, then all of the higher level CA certificates in the chain must not include this component when the extension is present.

See “CA Certificates and Extension Interactions” on page 282 regarding the interaction of the this extension with the Netscape Certificate Type extension.

CMS Version Support

Refer to “Basic Constraints Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator’s Guide*.

- **CMS 4.1:** Supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape requires this extension for all CA certificates.

Microsoft Recommendation

Microsoft recommends this extension for all certificates.

certificatePolicies

OID

2.5.29.32

References

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.5

Criticality

This extension may be critical or noncritical.

Discussion

The Certificate Policies extension defines one or more policies, each of which consists of an OID and optional qualifiers. The extension can include a URI to the issuer's Certificate Practice Statement or can embed issuer policy information, such as a user notice in text form. This information can be used by certificate-enabled applications.

If this extension is present, PKIX Part 1 recommends that policies be identified with an OID only, or if necessary only certain recommended qualifiers.

CMS Version Support

Refer to "Certificate Policies Extension Policy" in Chapter 18 of *Netscape Certificate Management System Administrator's Guide*.

- **CMS 4.1:** Not supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape recommends that this extension be included at the discretion of the certificate issuer.

Microsoft Recommendation

Microsoft recommends that this extension be included in all certificates.

cRLDistributionPoints

OID

2.5.29.31

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.14

Criticality

PKIX recommends that this extension be marked noncritical and that it be supported for all certificates.

Discussion

This extension defines how CRL information for this certificate is to be obtained. It should be used if the system is configured to use CRL issuing points.

If the extension contains a `DistributionPointName` of type `URI`, the URI is assumed to be a pointer to the current CRL for the associated reasons and will be issued by the associated `cRLIssuer`. The expected values for the URI are those defined for the `subjectAltName` extension. If the `distributionPoint` omits reasons, the CRL must include revocations for all reasons. If the `distributionPoint` omits `cRLIssuer`, the CRL must be issued by the CA that issued the certificate.

PKIX recommends that this extension be supported by CAs and applications.

CMS Version Support

Refer to “CRL Distribution Points Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator's Guide*.

- **CMS 4.1:** Supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape recommends that this extension be supported for all certificates, with the exception of self-signed root CA certificates.

Microsoft Recommendation

Microsoft recommends that this extension be supported.

extKeyUsage

OID

2.5.29.37

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.13

Criticality

If this extension is marked critical, the certificate must be used for one of the indicated purposes only. If it is not marked critical, it is treated as an advisory field that may be used to identify keys but does not restrict the use of the certificate to the indicated purposes.

Discussion

The Extended Key Usage extension indicates one or more purposes for which the certified public key may be used. These purposes may be in addition to or in place of the basic purposes indicated in the key usage extension.

The Extended Key Usage extension must include OCSP Signing in an OCSP responder's certificate (unless the CA signing key that signed the certificates validated by the responder is also the OCSP signing key). The OCSP responder's certificate must be issued directly by the CA that signs certificates the responder will validate.

The Key Usage, Extended Key Usage, and Basic Constraints extensions act together to define the purposes for which the certificate is intended to be used. Applications can use these extensions to disallow the use of a certificate in inappropriate contexts.

Table B.2 lists the uses defined by PKIX for this extension, and Table B.3 lists uses privately defined by Microsoft or Netscape.

Table B.2 PKIX Extended Key Usage Extension Uses

Use	OID
Server authentication	1.3.6.1.5.5.7.3.1
Client authentication	1.3.6.1.5.5.7.3.2
Code signing	1.3.6.1.5.5.7.3.3
Email	1.3.6.1.5.5.7.3.4
Timestamping	1.3.6.1.5.5.7.3.8
OCSP Signing	1.3.6.1.5.5.7.3.9*

* OCSP Signing is not defined in PKIX Part 1, but in RFC 2560, “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP.”

Table B.3 Private Extended Key Usage Extension Uses

Use	OID
Certificate trust list signing	1.3.6.1.4.1.311.10.3.1
Microsoft Server Gated Crypto (SGC)	1.3.6.1.4.1.311.10.3.3
Microsoft Encrypted File System	1.3.6.1.4.1.311.10.3.4
Netscape SGC	2.16.840.1.113730.4.1

CMS Version Support

Refer to “Extended Key Usage Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator’s Guide*.

- **CMS 4.1:** Not supported
- **CMS 4.2:** Supported

Netscape Recommendations

Netscape recommends that this extension be supported for all certificates, and requires it for all certificates that support step-up, or Server Gated Crypto (SGC). OCSP Signing should be included in all certificates issued to OCSP responders.

Microsoft Recommendations

Microsoft products interpret this extension as follows. If the extension is not present, the certificate is considered to be valid for any usage (to support backward compatibility with certificates that did not use this extension). Otherwise, interpretation depends on usage, as follows:

- Authenticode requires that Code Signing be the unique usage specified.
- SGC operation requires that the SGC usage be specified.
- Timestamping requires that timestamping usage be specified.

Microsoft allows users to control certificate properties that correspond to Extended Key Usage specifications. For example, from the Internet Explorer 4.0 user interface, the user may deselect a CA certificate in a list of CA certificates otherwise trusted for a given usage. Note that the user may only restrict uses, and not add uses that are not supported by the certificate itself. These user settings affect only the interpretation of the certificate on the computer where they are set. They do not affect the certificate itself.

A given certificate is valid only for the intersection of key usages of all the certificates in the chain to its root (as determined by both the Extended Key Usage extension for each certificate and the corresponding user settings). To be valid for a particular usage, the end-entity certificate and all certificates in the chain must all be valid for that usage.

issuerAltName

OID

2.5.29.18

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.8

Criticality

PKIX Part 1 recommends that this extension be marked noncritical.

Discussion

The `IssuerAlternativeName` extension is used to associate Internet-style identities with the certificate issuer. Names must use the forms defined for `subjectAltName`.

CMS Version Support

Refer to “Issuer Alternative Name Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator’s Guide*.

- **CMS 4.1:** Not supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape products do not examine this extension.

Microsoft Recommendation

Microsoft products do not examine this extension. Microsoft recommends that `authorityKeyIdentifier` be used rather than `issuerAltName` or the certificate’s issuer name for the purposes of building certificate chains.

keyUsage

OID

2.5.29.15

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.3

Criticality

This extension may be critical or noncritical. PKIX Part 1 recommends that it should be marked critical if it is used.

Discussion

The Key Usage extension defines the purpose of the key contained in the certificate. The Key Usage, Extended Key Usage, Basic Constraints, and Netscape Certificate Type extensions act together to specify the purposes for which a certificate can be used. For more information on interactions between these extensions in CA certificates, see “CA Certificates and Extension Interactions” on page 282.

If this extension is included at all, set the bits as follows:

- `digitalSignature` (0) for SSL client certificates, S/MIME signing certificates, and object-signing certificates.
- `nonRepudiation` (1) for some S/MIME signing certificates and object-signing certificates. Note, however, that the use of this bit is controversial. You should carefully consider the legal consequences of its use before setting it for any certificate.
- `keyEncipherment` (2) for SSL server certificates and S/MIME encryption certificates.
- `dataEncipherment` (3) when the subject's public key is used to encipher user data (as opposed to key material).
- `keyAgreement` (4) whenever the subject's public key is used for key agreement.
- `keyCertSign` (5) for all CA signing certificates
- `cRLSign` (6) for CA signing certificates that are used to sign CRLs
- `encipherOnly` (7) if the public key is to be used only for enciphering data. If this bit is set, `keyAgreement` should also be set.
- `decipherOnly` (8) if the public key is to be used only for deciphering data. If this bit is set, `keyAgreement` should also be set.

Table B.4 summarizes the above guidelines for typical certificate uses.

Table B.4 Certificate uses and corresponding Key Usage bits

Purpose of certificate	Required Key Usage bit
CA Signing	keyCertSign cRLSign
SSL Client	digitalSignature
SSL Server	keyEncipherment
S/MIME Signing	digitalSignature
S/MIME Encryption	keyEncipherment
Certificate Signing	keyCertSign
Object Signing	digitalSignature

If the `keyUsage` extension is present and is marked critical, then it will be used to enforce the usage of the certificate and key. The extension is used to limit the usage of a key; if the extension is not present or not critical, all types of usage are allowed.

If the `keyUsage` extension is present (critical or not), it is used to select from multiple certificates for a given operation. For example, it is used to distinguish separate signing and encryption certificates for users who have separate certificates and key pairs for these operations.

CMS Version Support

Refer to “Key Usage Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator’s Guide*.

- **CMS 4.1:** Supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape recommends this extension for all certificates if their intended purpose or purposes are known. Netscape requires this extension for all dual-key signing certificates.

Microsoft Recommendation

Microsoft recommends this extension for all certificates if their intended purpose or purposes are known. If the extension is absent, Microsoft products will assume the certificate is valid for all usages. If the extension is present, Microsoft products will interpret the extension in the same way whether marked critical or not. If the extension is present, the actual usage must conform to the specified usage.

The only Microsoft application that currently enforces this extension is Microsoft Outlook.

nameConstraints

OID

2.5.29.30

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.11

Criticality

PKIX Part 1 requires that this extension be marked critical.

Discussion

This extension, which can be used in CA certificates only, defines a name space within which all subject names in subsequent certificates in a certification path must be located.

CMS Version Support

Refer to “Name Constraints Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator’s Guide*.

- **CMS 4.1:** Not supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape products do not currently examine this extension.

Microsoft Recommendation

Microsoft products do not currently examine this extension.

OCSPNocheck

OID

1.3.6.1.5.5.7.48.4

Reference

<http://www.ietf.org/rfc/rfc2560.txt> 4.2.2.2.1

Criticality

This extension should be noncritical.

Discussion

The extension is meant to be included in an OCSP responder's signing certificate. The extension tells an OCSP client that the signing certificate can be trusted without querying the OCSP responder (since the reply would again be signed by the OCSP responder, and the client would again request the validity status of the signing certificate). This extension is null-valued: its meaning is determined by its presence or absence.

Since the presence of this extension in a certificate will cause OCSP clients to trust responses signed with that certificate, use of this extension should be managed carefully. If the OCSP signing key is compromised, the entire process of validating certificates in the PKI will be compromised for the duration of the validity period of the certificate. Therefore, certificates using `OCSPNocheck` should be issued with short lifetimes and be renewed frequently.

CMS Version Support

Refer to "OCSPNocheck Extension Policy" in Chapter 18 of *Netscape Certificate Management System Administrator's Guide*.

- **CMS 4.1:** Not supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape recommends using this extension in OCSF responder signing certificates. The validity period should be short enough to minimize the potential impact of a compromised OCSF responder signing key to your organization.

Microsoft Recommendation

Microsoft products do not currently use online status checking.

policyConstraints

OID

2.5.29.36

References

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.12

Criticality

This extension may be critical or noncritical.

Discussion

This extension, which is for CA certificates only, constrains path validation in two ways. It can be used to prohibit policy mapping or to require that each certificate in a path contain an acceptable policy identifier.

PKIX requires that, if present, this extension must never consist of a null sequence. At least one of the two available fields must be present.

CMS Version Support

Refer to “Policy Constraints Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator's Guide*.

- **CMS 4.1:** Not supported
- **CMS 4.2:** Supported

Netscape Recommendations

Netscape products do not currently examine this extension.

Microsoft Recommendations

Microsoft products do not currently examine this extension.

policyMappings

OID

2.5.29.33

References

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.6

Criticality

This extension must be noncritical.

Discussion

The Policy Mappings extension is used in CA certificates only. It lists one or more pairs of OIDs used to indicate that the corresponding policies of one CA are equivalent to policies of another CA. It may be useful in the context of cross-certification.

This extension may be supported by CAs and/or applications.

CMS Version Support

Refer to “Policy Mappings Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator's Guide*.

- **CMS 4.1:** Not supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape products do not currently examine this extension.

Microsoft Recommendations

Microsoft products do not currently examine this extension.

privateKeyUsagePeriod

OID

2.5.29.16

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.4

Discussion

The Private Key Usage Period extension allows the certificate issuer to specify a different validity period for the private key than for the certificate itself. This extension is intended for use with digital signature keys.

PKIX Part 1 recommends against the use of this extension. CAs conforming to PKIX Part 1 *must not* generate certificates with this extension.

CMS Version Support

Refer to “Private Key Usage Period Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator’s Guide*.

- **CMS 4.1:** Not supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape recommends against the use of this extension.

Microsoft Recommendation

Microsoft recommends against the use of this extension.

subjectAltName

OID

2.5.29.17

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.7

Criticality

If the certificate's subject field is empty, this extension must be marked critical.

Discussion

The Subject Alternative Name extension includes one or more alternative (non-X.500) names for the identity bound by the CA to the certified public key. It may be used in addition to the certificate's subject name or as a replacement for it. Defined name forms include Internet electronic mail address (SMTP, as defined in RFC-822), DNS name, IP address, and uniform resource identifier (URI).

PKIX requires this extension for entities that are identified by name forms other than the X.500 distinguished name (DN) used in the subject field. PKIX Part 1 describes additional rules for the relationship between this extension and the subject field.

Email addresses may be provided either in the Subject Alternative Name extension, the certificate subject name field, or both. If the email address is provided as part of the subject name, it must be in the form of the `EmailAddress` attribute defined by PKCS-9. Software that supports S/MIME must be able to read an email address from either the Subject Alternative Name extension or from the subject name field.

CMS Version Support

Refer to "Subject Alternative Name Extension Policy" in Chapter 18 of *Netscape Certificate Management System Administrator's Guide*.

- **CMS 4.1:** Supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape recommends the use of this extension with all certificates issued by a CA (except for SSL client certificates).

Netscape products read only the first alternative name in this extension, and ignore the rest. For S/MIME certificates, Netscape software first checks the first alternative name in this extension (if the extension is present) for the `EmailAddress` attribute. If the first alternative name is not an `EmailAddress` attribute, Netscape software looks for the `e=` attribute of the DN. If the `e=` attribute is not present, Netscape software looks for the `mail=` attribute of the DN.

Microsoft Recommendation

Microsoft recommends the use of this extension whenever X.500 guidelines are insufficient for naming purposes. Currently, no Microsoft products require the use of Subject Alternative Name. All Microsoft products that support S/MIME are capable of reading email names from this extension or from the subject name. Future versions of Microsoft Exchange Server will issue certificates with X.500 names that do not contain the Email Address attribute, and will place the SMTP address in the Subject Alternative Name extension.

subjectDirectoryAttributes

OID

2.5.29.9

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.9

Criticality

PKIX Part 1 requires that this extension be marked noncritical.

Discussion

The Subject Directory Attributes extension conveys any desired directory attribute values for the subject of the certificate. It is not recommended as an essential part of the proposed PKIX standard, but may be used in local environments.

CMS Version Support

Refer to “Subject Directory Attributes Extension Policy” in Chapter 18 of *Netscape Certificate Management System Administrator’s Guide*.

- **CMS 4.1:** Not supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape products do not examine this extension.

Microsoft Recommendation

Microsoft products do not examine this extension.

subjectKeyIdentifier

OID

2.5.29.14

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 4.2.1.2

Criticality

This extension is always noncritical.

Discussion

The Subject Key Identifier extension identifies the public key certified by this certificate. This extension provides a way of distinguishing public keys if more than one is available for a given subject name, for example after the certificate has been renewed with a new key.

The value of this extension should be calculated by performing a SHA-1 hash of the certificate's DER-encoded `subjectPublicKeyInfo`, as recommended by PKIX. This extension is used with the form of the `authorityKeyIdentifier` extension in which the issuer's public key is specified by a hash. In this case the verifier does not need to compute the hash, since it's only necessary to compare the issuer's Subject Key Identifier with the subject's Authority Key Identifier.

PKIX Part 1 requires this extension for all CA certificates and recommends it for all other certificates.

CMS Version Support

Refer to "Subject Key Identifier Extension Policy" in Chapter 18 of *Netscape Certificate Management System Administrator's Guide*.

- **CMS 4.1:** Supported
- **CMS 4.2:** Supported

Netscape Recommendation

Netscape recommends this extension for all certificates.

Microsoft Recommendation

Microsoft recommends this extension for all certificates.

Standard X.509 v3 CRL Extensions

In addition to certificate extensions, the X.509 v3 proposed standard defines extensions to CRLs, which provide methods for associating additional attributes with Internet CRLs. These are of two kinds: extensions to the CRL itself, and extensions to individual certificate entries in the CRL.

- Extensions for CRLs (page 276)
- CRL Entry Extensions (page 279)

Extensions for CRLs

The sections that follow describe the CRL extension types that are defined as part of the Internet X.509 v3 Public Key Infrastructure proposed standard, as of September 1998.

These are the CRL extensions described in the sections that follow:

- authorityKeyIdentifier (page 276)
- CRLNumber (page 276)
- deltaCRLIndicator (page 277)
- issuerAltName (page 278)
- issuingDistributionPoint (page 278)

authorityKeyIdentifier

OID

2.5.29.35

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 5.2.1

Discussion

The Authority Key Identifier extension for a CRL identifies the public key corresponding to the private key used to sign the CRL. For details, see the discussion under certificate extensions at authorityKeyIdentifier.

CRLNumber

OID

2.5.29.20

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 5.2.3

Criticality

This extension must not be critical.

Discussion

The CRL Number extension specifies a sequential number for each CRL issued by a CA. It allows users to easily determine when a particular CRL supersedes another CRL.

PKIX requires that all CRLs have this extension.

deltaCRLIndicator

OID

2.5.29.27

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 5.2.4

Criticality

PKIX requires that this extension be critical if it exists.

Discussion

The Delta CRL Indicator extension identifies a delta-CRL. The use of delta-CRLs allows changes to be added to the local database while ignoring unchanged information that is already in the local database. This can significantly improve processing time for applications that store revocation information in a format other than the CRL structure.

This extension is used only with delta-CRLs, which are not supported by Certificate Management System.

issuerAltName

OID

2.5.29.18

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 5.2.2

Discussion

The Issuer Alternative Name extension allows additional identities to be associated with the issuer of the CRL. For details, see the discussion under certificate extensions at `issuerAltName`.

issuingDistributionPoint

OID

2.5.29.28

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 5.2.5

Criticality

PKIX requires that this extension be critical if it exists.

Discussion

The Issuing Distribution Point CRL extension identifies the CRL distribution point for a particular CRL and indicates what kinds of revocation it covers.

PKIX Part I does not require this extension.

CRL Entry Extensions

The sections that follow lists the CRL entry extension types that are defined as part of the Internet X.509 v3 Public Key Infrastructure proposed standard, as of September 1998. All of these extensions are noncritical.

These are the CRL entry extensions described in the sections that follow:

- `certificateIssuer` (page 279)
- `holdInstructionCode` (page 279)
- `invalidityDate` (page 280)
- `reasonCode` (page 280)

certificateIssuer

OID

2.5.29.29

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 5.3.4

Discussion

The Certificate Issuer extension identifies the certificate issuer associated with an entry in an indirect CRL.

This extension is used only with indirect CRLs, which are not supported by Certificate Management System.

holdInstructionCode

OID

2.5.29.23

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 5.3.2

Discussion

The Hold Instruction Code extension indicates the action to be taken after encountering a certificate that has been placed on hold.

invalidityDate

OID

2.5.29.24

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 5.3.3

Discussion

The Invalidity Date extension provides the date on which the private key was compromised or that the certificate otherwise became invalid.

reasonCode

OID

2.5.29.21

Reference

<http://www.ietf.org/rfc/rfc2459.txt> 5.3.1

Discussion

The Reason Code extension identifies the reason for certificate revocation.

Netscape-Defined Certificate Extensions

Netscape has defined certain certificate extensions for use with Navigator and Communicator. Some of the extensions that have been defined are now obsolete, and others can be superseded by the extensions defined in the X.509

proposed standard. All Netscape extensions should be tagged as noncritical, so that their presence in a certificate does not make that certificate incompatible with other clients.

The specifications for all Netscape-defined extensions are defined at <http://home.netscape.com/eng/security/comm4-cert-exts.html>. For most CMS deployments, only `netscape-cert-type` and `netscape-comment` need to be supported to maintain compatibility with Navigator 3.x. Therefore, only these two Netscape certificate extensions are described here.

netscape-cert-type

OID

2.16.840.1.113730.1

Discussion

The Netscape Certificate Type extension can be used to limit the purposes for which a certificate can be used. It has been replaced by the X.509 v3 extensions **extKeyUsage** and **basicConstraints**, but must still be supported in deployments that include Navigator 3.x clients.

If the extension exists in a certificate, it limits the certificate to the uses specified in it. If the extension is not present, the certificate can be used for all applications except object signing.

The value is a bit-string, where the individual bit positions, when set, certify the certificate for particular uses as follows:

- bit 0: SSL Client certificate
- bit 1: SSL Server certificate
- bit 2: S/MIME certificate
- bit 3: Object-signing certificate
- bit 4: Reserved for future use
- bit 5: SSL CA certificate
- bit 6: S/MIME CA certificate
- bit 7: Object-signing CA certificate

netscape-comment

OID

2.16.840.1.113730.13

Discussion

The value of this extension is an IA5String. It is a comment that can be displayed to the user when the certificate is viewed.

Adding Extensions in Certificate Management System

When Certificate Management System creates a certificate in response to a certificate request, it can add extensions according to the policies defined in extension-specific policy modules. Policy modules are available with the distribution that can be used to add certificate extensions. For more information about policies and extensions, see Chapter 16, “Introduction to Policy,” in *Netscape Certificate Management System Administrator's Guide*.

By default, only noncritical extensions are added to certificates. This ensures that the resulting certificates can be used with all clients. If you add a critical extension, the resulting certificate can only be used by clients that support that extension.

You can write a policy module to add any extension that Certificate Management System supports. For a summary of the certificate extensions supported by CMS policy modules, see “Policy Modules” on page 56.

CA Certificates and Extension Interactions

Netscape recommends that all CA certificates contain the `basicConstraints` extension, as this is the standard way to identify a CA certificate. In addition, to ensure support for Navigator 3.x, CAs should also use `netscape-cert-type`. These two extensions can interact with each other. The following table describes what different combinations of the two extensions mean.

Extensions Present	Description
Only <code>basicConstraints</code>	The certificate is a CA certificate if the <code>cA</code> component is true. Path length processing is done as described above.
Only <code>netscape-cert-type</code>	The certificate is a CA if at least one of the CA bits is set: SSL CA (5), S/MIME CA (6), or object-signing CA (7). The certificates issued by this CA are limited to the particular applications specified. Path length processing is done as though the <code>pathLenConstraint</code> is unlimited.
Neither extension	The certificate is not a CA.
Both extensions	The certificate is a CA certificate if the <code>cA</code> component of <code>basicConstraints</code> is true. If one or more of the SSL CA (5), S/MIME CA (6), or object-signing CA (7) bits are set in the <code>netscape-cert-type</code> extension, then the CA will be limited to issuing certificates for the specified application areas; otherwise, the CA can issue certificates for any application.

A certificate chain generally consists of an entity certificate, zero or more intermediate CA certificates, and a root CA certificate. Typically the root CA certificate is self-signed and is loaded into Communicator's certificate database as a trusted CA.

An exchange of certificates takes place when performing an SSL handshake, when sending an S/MIME message, or when sending a signed object. As part of the handshake, the sender is expected to send the subject certificate and any intermediate CA certificates needed to link the subject certificate to the trusted root. For certificate chaining to work properly the certificates should have the following properties:

- CA certificates must have either the **`basicConstraints`** extension, the `netscape-cert-type` extension with one or more CA bits set, or both, as described above.
- If CAs issue multiple certificates for the same identity, for example for separate signing and encryption keys, they must include the `keyUsage` extension in the subject certificates.
- If CAs ever intend to generate new keys for their CA, they must add the `authorityKeyIdentifier` extension to all subject certificates. If the `key ID` is anything other than the SHA-1 hash of the CA certificates

CA Certificates and Extension Interactions

`subjectPublicKeyInfo` field, then the CA certificate should contain the `subjectKeyIdentifier` extension. This will allow for a smooth transition when the new issuing certificate becomes active.

Certificate Download Specification

This appendix describes the data formats used by Netscape Communicator 4.x for installing certificates. It also describes how certificates are imported into different environments.

- Data Formats (page 285)
- Importing Certificate Chains (page 286)
- Importing Certificates into Netscape Communicator (page 287)
- Importing Certificates into Netscape Servers (page 288)
- Object Identifiers (page 288)

Data Formats

Netscape products can accept certificates in several formats. Although the format can vary, the certificates themselves are X.509 version 1, 2, or 3.

Binary Formats

The Netscape certificate loader recognizes several binary formats, as follows.

- **DER-encoded certificate**

This is a single binary DER-encoded certificate.

- **PKCS #7 certificate chain**

This is a PKCS #7 *SignedData* object. The only significant field in the *SignedData* object is the certificates. In particular, the signature and the contents are ignored. In future versions of the software, the CRLs will also be used. The PKCS #7 format allows multiple certificates to be downloaded at once. See Importing Certificate Chains (page 286) for more information about handling multiple certificates.

- **Netscape Certificate Sequence**

This is a simpler format for downloading certificate chains. It consists of a PKCS #7 *ContentInfo* structure, wrapping a sequence of certificates. The value of the *contentType* field should be *netscape-cert-sequence* (see Object Identifiers on page 288), while the *content* field has the following structure:

```
CertificateSequence ::= SEQUENCE OF Certificate
```

This format allows multiple certificates to be downloaded at once. See Importing Certificate Chains (page 286) for more information about handling multiple certificates.

Text Formats

Any of the above binary formats can also be imported in text form. The text form begins with the following line:

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

Following this line is the certificate data, which can be in any of the binary formats just described. This data should be base 64 encoded as described by RFC 1113. The data is followed by this line:

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

Importing Certificate Chains

Several of the supported formats can contain multiple certificates. When the Netscape certificate decoder encounters a collection of certificates, it handles them as follows:

- The first certificate is processed in a context-specific manner, which varies according to how it is being imported. For Communicator, this handling depends upon the MIME content type that is used on the object being downloaded. For Netscape servers, it depends upon the options selected in the server administration interface.
- Subsequent certificates are all treated the same. If the certificates contain the SSL-CA bit in the netscape-cert-type certificate extension and do not already exist in the local certificate database, they are added as untrusted CAs. In this way they can be used for certificate chain validation as long as there is a trusted CA somewhere along the chain.

Importing Certificates into Netscape Communicator

Communicator imports certificates via HTTP. There are several MIME content types that are used to indicate to Communicator what type of certificate is being imported. These MIME types are as follows:

- `application/x-x509-user-cert`

The certificate being downloaded is a user certificate belonging to the user operating Communicator. If the private key associated with the certificate does not exist in the user's local key database, then Communicator generates an error dialog and the certificate is not imported. If a certificate chain is being imported, then the first certificate in the chain must be the user certificate, and any subsequent certificates will be added as untrusted CA certificates to the local database.

- `application/x-x509-ca-cert`

The certificate being downloaded represents a certificate authority. When it is downloaded, a sequence of dialogs guides the user through the process of accepting the Certificate Authority and deciding whether to trust sites certified by the CA.

If a certificate chain is being imported, the first certificate in the chain must be the CA certificate, and Communicator adds any subsequent certificates in the chain to the local database as untrusted CA certificates.

- `application/x-x509-email-cert`

The certificate being downloaded is a user certificate belonging to another user for use with S/MIME. If a certificate chain is being imported, the first certificate in the chain must be the user certificate, and Communicator adds any subsequent certificates to the local database as untrusted CA certificates. This process allows people or CAs to post their email certificates on web pages for download by other users who want to send them encrypted mail.

Note Communicator checks that the size of the object being downloaded matches the size of the encoded certificates. Therefore it is important to ensure that no extra characters, such as `NULL` or `NewLine`, are added at the end of the object.

Importing Certificates into Netscape Servers

Server certificates are imported via the server administration interface. Certificates are pasted into a text input field in an HTML form, and then the form is submitted to the administration server. Since the certificates are pasted into text fields, only the text formats described above are supported for servers.

The type of certificate being imported is specified by the server administrator by selections made on the administration pages. If a certificate chain is being imported, then the first certificate in the chain must be the server or CA certificate, and the server adds any subsequent certificates to the local database as untrusted CA certificates.

For detailed information about importing certificates into Netscape Enterprise Server and configuring it to support certificate-based client authentication, see Appendix D, "Using SSL with iPlanet Web Server, Enterprise Edition 4.x" (page 291).

Object Identifiers

The base of all Netscape object IDs is

```
netscape OBJECT IDENTIFIER ::= { 2 16 840 1 113730 }
```

The hexadecimal byte value of this OID, when DER-encoded, is

```
0x60, 0x86, 0x48, 0x01, 0x86, 0xf8, 0x42
```

The following OIDs are mentioned in this document:

```
netscape-data-type OBJECT IDENTIFIER ::= { netscape 2 }
```

```
netscape-cert-sequence OBJECT IDENTIFIER ::= { netscape-data-type 5 }
```

Object Identifiers

Using SSL with iPlanet Web Server, Enterprise Edition 4.x

This appendix explains how to get client certificate authentication working with iPlanet Web Server, Enterprise Edition 4.x. When you have finished following these steps, you will have a web server that requires a user to present a valid client SSL certificate (issued by Certificate Management System) in order to access the restricted areas on the server. The certificate that the user presents must match the certificate that was published to the LDAP directory when it was issued.

To use SSL with iPlanet Web Server, you must either have an existing instance of iPlanet Web Server 4.x that you want to be an SSL server or create a new instance to be an SSL server. To create a new instance, see “Creating a New Server” on page 292.

To enable SSL for a particular server instance, you must obtain a server SSL certificate for the server, then configure the server to require client authentication and to check users’ client certificates against certificate information that Certificate Management System has published to the LDAP directory.

This appendix has the following sections:

- Creating a New Server (page 292)
- Obtaining a Server Certificate (page 293)
- Enabling SSL on the Server (page 299)
- Testing Client Authentication (page 308)

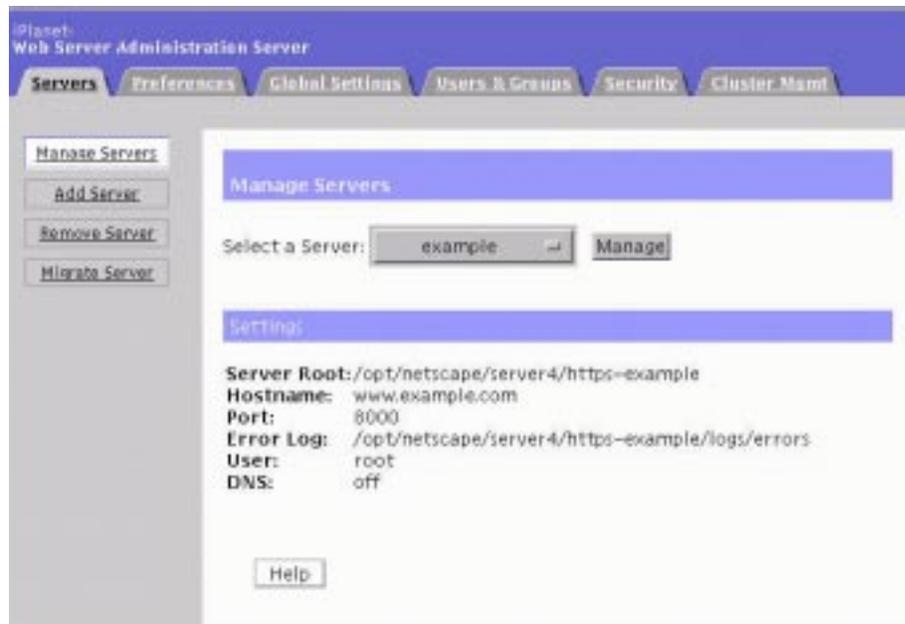
Creating a New Server

If you have an existing instance of iPlanet Web Server that you want to simply convert to be an SSL server, you can skip this step and go to “Obtaining a Server Certificate” on page 293. Otherwise, create a new instance of iPlanet Web Server and follow the remaining procedures to configure the new instance for SSL and client authentication.

To create a new instance of the server, follow these steps:

1. Log into iPlanet Web Server Administration Server using your administrator's ID and password.

A Manage Servers window appears. In this figure, there is already one server running called example, on port 8000.



2. Click Add Server. In the screen that appears, most of the fields have default values.
3. Verify and update any settings as necessary. Sample server settings are:
 - **Server Name:** myhost.example.com

- **Server Port:** (typically 443, but can be any unused port)
- **Server Identifier:** `myhost-ssl`
- **Server User:** nobody

For details about these fields, click Help to see the iPlanet Web Server documentation.

4. Submit the form.
A notification for a new server is created.
5. When you are ready to configure the new server to enable SSL, click “Configure your new server.”
See “Enabling SSL on the Server” on page 299.

Obtaining a Server Certificate

You must obtain a server SSL certificate and import it into iPlanet Web Server before you can configure the server to use SSL. To obtain the server SSL certificate for an existing instance of iPlanet Web Server, follow the steps in the following sections:

- Creating a Trust Database (page 293)
- Submitting a Certificate Signing Request (page 294)
- Importing the Certificate (page 296)

Creating a Trust Database

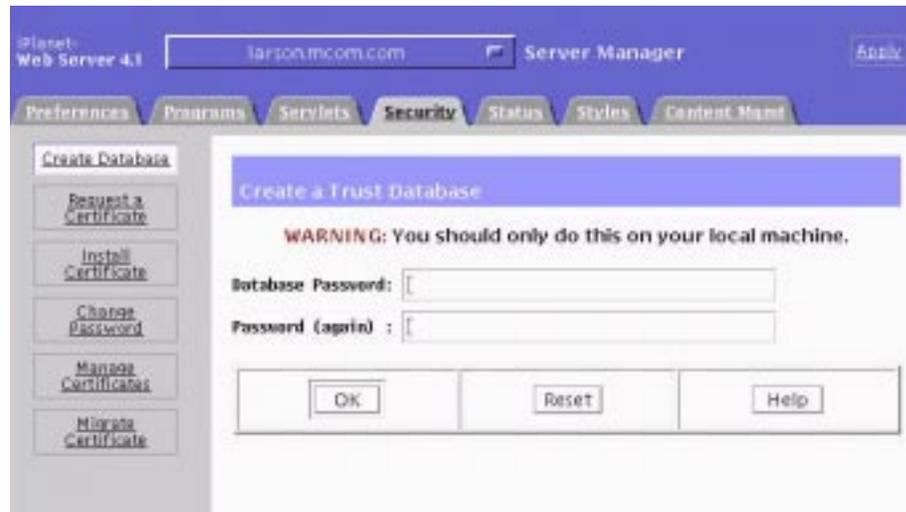
To support SSL, you first create a Trust Database that will contain all of the keys and certificates used by the server (including many pre-installed root certificates from public Certificate Authorities).

To create a trust database:

1. Open the Server Manager page for the server you want to configure.
2. Click the Security tab.

The default page on the Security Tab is the Create a Trust Database page; an example is shown in the following figure.

Obtaining a Server Certificate



3. Type in a password to protect the Trust Database in the password fields.
This password will protect the certificates the server uses, including its SSL server certificate. The password must contain at least 8 characters and have at least one non-alphabetic character.
Whenever you start an SSL-enabled HTTP server, you will be asked for this password to access the certificate database.
4. Click OK to submit the form.

Submitting a Certificate Signing Request

Once you have a Trust Database, you can create a PKCS #10 certificate request and submit it to Certificate Management System to obtain your server SSL certificate.

To generate the PKCS #10 certificate request, follow these steps:

1. Open the Server Manager page for the server you want to configure.
2. Click the Security tab.
3. Click Request a Certificate in the Security tab menu.

This figure shows an example of the Request a Server Certificate page that appears.

iPlanet Web Server 4.1 | larson.mcom.com | Server Manager | [Help](#)

Preferences Programs Servlets **Security** Status Styles Content Mgmt

[Create Database](#)

[Request a Certificate](#)

[Install Certificate](#)

[Change Password](#)

[Manage Certificates](#)

[Migrate Certificate](#)

Request a Server Certificate

New certificate.
 Certificate renewal.

You can also see a [list of available certificate authorities](#).

Submit to Certificate Authority via:

CA Email Address:

CA URL:

Select the module to use with this certificate.

Cryptographic Module:

Key Pair File Password:

Before requesting a certificate, you should read the [overview](#) of the certificate process, and then go through the [detailed steps](#) on creating a correct distinguished name which you should enter below. You will also generate the proper authorization letter that you will use to obtain your certificate from a certification authority.

Requestor name:

Telephone number:

Common name:

Email address:

Organization:

Organizational Unit:

4. Select the New certificate checkbox.
5. In the “Submit to Certificate Authority via” area, select the CA URL checkbox.

6. In the CA URL text field, enter the URL for the end-entity enrollment interface of a CMS Certificate or Registration Manager.
Simply append `/enrollment` to the URL for the end-entity gateway. For example, `https://myca.example.com:443/enrollment`.
7. Type the Trust Database password in the Key Pair File Password field.
8. The remaining fields request identifying information about the server. Use the fully qualified domain name of the server for Common Name.
9. Click OK to submit the form.
A confirmation window appears, showing the information you entered and the PKCS#10 request. Back up the PKCS#10 data by copying it with the browser's copy command and pasting it into a file using a text editor.
10. Double check the information about the server, then click OK to submit the form.
A message from the CMS server appears to tell you that the request is pending. Note the request ID number; it can be used to retrieve the certificate from the CMS end-entity gateway when the certificate is issued.

Importing the Certificate

A CMS agent will process your certificate request. When the certificate is issued, you will receive an email containing the certificate or a URL where the certificate can be retrieved. Once you have been issued a server certificate, you must import it into your server. (This is different from importing a personal certificate into your browser.)

To import the server certificate into the server, follow these steps:

1. In your browser or mail reader, go to the page or message containing the certificate.
2. Scroll down to the part of the page that contains the base-64 encoded certificate. It looks like this:

```
-----BEGIN CERTIFICATE-----  
MIICeTCCAeKgAwIBAgICHfQwDQYJKoZIhvcNAQEEBQAwdzELMAkGA1  
UEBhMCVVMxLDAqBgNVBAoTI05ldHNjYXB1IENvbW11bm1jYXRpb25z
```

```
IENvcnBvcYXB1IENvbW11bm1jYXRpb25zIENvcnAuMSAwHgYDVQQDF  
Bdtb2cuKG5ldHNjYXB1fG1jb20pLmNvbTCBnTANBgkqhkiG9w0B  
N0nZmUaB3adv7D1TPA==  
-----END CERTIFICATE-----
```

3. Select and copy the base-64 encoded certificate, using Copy from the Edit menu in the browser or mail reader.
4. Go back to your iPlanet Web Server's Administration page, and open the Server Manager page for the server where you are installing this certificate.
5. Click the Security tab.
6. Click Install Certificate in the Security tab menu.

Obtaining a Server Certificate

This page appears.

The screenshot shows the 'Install a Server Certificate' dialog box in the Netscape Server Manager. The window title is 'Server Manager' and the address bar shows 'larsen.mcom.com'. The 'Security' tab is selected. On the left, there is a sidebar with buttons: 'Create Database', 'Request a Certificate', 'Install Certificate', 'Change Password', 'Manage Certificates', and 'Migrate Certificate'. The main area contains the following fields and options:

- Certificate for:**
 - This Server
 - Server Certificate Chain
 - Trusted Certificate Authority (CA)
- Select the module to use with this certificate.**
- Cryptographic Module:** internal (software)
- Key Pair File Password:** [password field]
- Enter Certificate Name ONLY if certificate is not for 'This Server'.**
- Certificate Name:** [text field]
- Message is in this file: [text field]
- Message text (with headers):
 - BEGIN CERTIFICATE-----
 - MIICCTCCAhugAeTBAAIBCSAMBkgkHfC9wDBAQFADBI1M0swCQYDV0GFWzVUzET
 - NEEGAIUECBWQZFsxKZvcaTpyTEVMBQGAUEBwMTY9IbnRhaWQvvlIidEMWwG
 - ATUECvMDQ0ITM2bWV0V0QLEwREb2NzMRwGyVW00EwNBZlT0aM2aV2F0Z5BN
 - YNSz2VvB4xDYAeMlYmZESM]IyOfokDTAeMlYm]ESM]IyOfewZELMAKGA1UE
 - BHNCBgnSbACBgVWAgTE2xsbGosbGosCzATBnVBAcTAcswMQswCQYDV0QEWzVU
 - bDELMAKGA1UECvM0bG9iCzATBnVBAcTAcswMQswCQYDV0QEWzVUzETNEEGAI
 - ADCS1QBgQCykIejH5OUvpaBkC70mVWk/rTWhovFuWZZrM0AcTKT0hF3DsoeF+r
 - eghWkKjPfrndkdG3n19d2S0+ys9vUtxEobRnBkIkusMDPKExaqqcaacWMPYTG
 - yYvyvWz2RzqICNLFUJ77Ik3+k006U2Pw07sCOE7Tzo31tFbuT1w10AqA8u3Qw

At the bottom, there are 'OK', 'Reset', and 'Help' buttons.

7. Verify that the certificate is for “This Server.”
8. Type the Trust Database password in the Key Pair File Password field.
9. Select “Message text (with headers).”
10. Paste the encoded certificate information into the text box.
11. Click OK.

A confirmation page appears, showing the contents of the certificate to be added.

12. Click Add Server Certificate.

A dialog box tells you to restart the iPlanet Web Server for the changes to take effect.

13. Complete the instructions in “Enabling SSL on the Server” on page 299 before you restart the server.

Enabling SSL on the Server

To enable SSL on the server, you must follow the steps in “Enabling Encryption on the Server” on page 300. This procedure is all that is required if you will not be using SSL client authentication.

If you plan to use SSL client authentication, you can enable it either for all connections or only for requests for resources protected by an access control list (ACL).

In both cases, you must import and trust the certificate of the CA that signs the client certificates you will trust. The instructions for this step are in “Trusting the Root CA Certificate” on page 300.

To enable SSL client authentication on all requests, perform only the task in “Enabling Client Authentication for All Requests” on page 302.

To enable SSL client authentication for ACL-protected resources, you must use an LDAP directory where end-entity’s certificates are published, tell the web server how to map certificates to directory entries, and configure ACLs to look up users in the directory. To accomplish these tasks, perform the procedures in the following sections:

- Specifying the Authentication Directory (page 302)
- Modifying the Configuration File (page 304)
- Modifying the Access Control Lists (page 306)

Enabling Encryption on the Server

This procedure explains how to turn SSL on for the server.

To enable the general use of SSL for server communications, follow these steps.

1. Open the Server Manager page for the server that you want to use SSL
2. Click the Preferences tab if it is not showing.
3. In the left frame, click Encryption On/Off.
The Encryption On/Off page appears.
4. For the Encryption checkbox, select On.
5. In the Port Number box, enter the port number you want to use for the SSL service.
(The default for HTTPS is 443.)
6. Verify that the correct alias is selected.
7. Click OK.
8. Follow the directions to Save and Apply the changes.

You can Save to save the changes to the configuration file, or Save and Apply to save the changes and restart the server. If you plan to continue configuring SSL, you can just Save now and restart the server later.

Trusting the Root CA Certificate

For the server to accept SSL client certificates issued by your root CA, you must import the certificate chain from your root CA into the server and establish it as a trusted CA.

Use the secure end-entity pages to import the certificate chain, as follows:

1. Go to the URL for the secure end-entity port of the Certificate Manager that is to act as your root CA, using HTTPS. For example:

```
https://myCA.example.com:443/
```

2. Select the Retrieval tab.
3. Click Import CA Certificate Chain.
4. In the Import CA Certificate Chain form, select “Display the CA certificate chain in PKCS#7 for importing into a server.”
5. Click Submit.

The certificate chain appears in your browser window in an encoded format.

6. Copy the encoded certificate chain, using Copy from the browser's Edit menu.
7. Open the iPlanet Web Server Server Manager page for the server where you want to import the CA certificate chain.
8. Click the Security tab.
9. Select Install Certificate from the Security tab menu on the left.
10. Select “Trusted Certificate Authority.”
11. Type the Trust Database password into the Key Pair File Password field.
12. Select “Message text (with headers),” and paste the encoded certificate chain into the text box.
13. Submit the form.
14. In the confirmation page, click Add Server Certificate.

After you have made the remaining configuration changes described next, restart the server for the changes to take effect.

Enabling Client Authentication for All Requests

To require SSL client authentication for all requests on a server:

1. Open the Server Manager page for the server that uses SSL.
2. Click the Preferences tab if it is not showing.
3. In the left frame, click Encryption Preferences.
4. For the “Require client certificate (regardless of access control)” checkbox, select Yes.
5. Click OK to submit the form.
6. Choose Save if you want to configure more about the server or Save and Apply to save changes and restart the server.

Specifying the Authentication Directory

You must specify a particular LDAP directory for iPlanet Web Server to use for authentication. This must be the same directory to which CMS publishes certificate information.

Note Certificate Management System must be configured to publish certificate information to a directory in order for the web server to verify the client certificate.

To specify an authentication directory, follow these steps:

1. Open the Web Server Administration Server page, and select the Global Settings tab.
2. Select Configure Directory Service.

This screen appears.

The screenshot shows the 'Configure Directory Service' window in the iPlanet Web Server Administration Server. The window title is 'iPlanet Web Server Administration Server'. The navigation tabs include 'Servers', 'Preferences', 'Global Settings', 'Users & Groups', 'Security', and 'Cluster Mgmt'. The 'Global Settings' tab is active, and the 'Configure Directory Service' button is selected. The main content area is titled 'Configure Directory Service' and contains the 'LDAP Directory Server Configuration' section. The fields are filled with the following values: Host Name: ldap.example.com, Port: 389, Base DN: o=example.com, Bind DN: uid=admin,ou=Administrators, ou=TopologyManagement, and Bind Password: *****. The 'Use Secure Sockets Layer (SSL) for connections?' checkbox is checked. At the bottom, there are buttons for 'Save Changes', 'Reset', and 'Help'.

3. Supply the host name, port number, and base DN for the LDAP directory to be used for authentication.
4. If you want, click Yes to specify an SSL connection for authentication communications between iPlanet Web Server and Directory Server. (You must also enable the SSL connection in Directory Server.)
5. Specify the Base DN to use for searching for user entries.
6. Specify the distinguished name (DN) to use to bind to the directory for searching.
The Bind DN can be the DN of a directory administrator or any DN that has permission to search the directory.
7. Specify the password used with the Bind DN to bind to the directory.
8. When you have finished filling out the form, save the changes to the iPlanet Web Server configuration.

Note for CGI Programmers

When you have set up your iPlanet Web Server to use the LDAP server as shown, you also get access to the following environment variables from within CGI scripts:

- `REMOTE_USER` is set to the UID of the user, such as `jsmith`. This is the most useful variable. For example, you can use it to check the LDAP directory for the user's manager, phone number, and so on, or you can customize the information presented to different users.
- `CLIENT_CERT` contains an encoded copy of the user's certificate.
- `AUTH_TYPE` is set to `ssl` when appropriate.
- `HTTPS` is set to `on` when appropriate.
- `HTTPS_KEYSIZE` is the number of bits in the encryption key, for example, 128.
- `HTTPS_SECRETKEYSIZE` is the number of bits in the secret key, usually 40 for export and 128 for the US.

Modifying the Configuration File

The iPlanet Web Server does not automatically check each certificate against the certificate revocation list (CRL), and so cannot detect a revoked certificate. However, if Certificate Management System is configured to remove revoked certificates from the LDAP directory, you can tell iPlanet Web Server to verify each client certificate against the LDAP directory, thus protecting against the presentation of revoked certificates.

For more information on the `certmap.conf` file, see "Editing the `certmap.conf` file" in Chapter 5, "Using SSL," in *Managing Servers with Netscape Console*.

The `certmap.conf` file tells iPlanet Web Server how to map a client certificate to the LDAP server to make a valid LDAP query. The file is located in `<web_server_root>/userdb/certmap.conf`.

Note The formatting of this file is extremely important. Extra spaces or linefeeds, for example, can cause certificate authorization to fail.

In this example of a `certmap.conf` file, we have issued certificates that have a UID field and then specified that field as the key field for the LDAP search.

```
certmap example CN=Certificate Manager, OU=Information Systems,
O=Example, C=US
example:DNComps O, C
example:FilterComps UID
example:verifycert on
```

- `certmap <token> <issuerDN>`

The `certmap` line establishes a token to identify rules corresponding to certificates whose issuer DN matches the DN provided. Subsequent lines in the `certmap.conf` file that begin with the token specify rules to map the SSL client certificate to an entry in the LDAP directory.

- `DNComps`

The `DNComps` line tells the server to glean the given attributes from the user's certificate to figure out where to start looking for the user in the LDAP tree. The example uses `O` and `C`: if a user's certificate has attributes "O=Netscape Communications Corp." and "C=US," the web server uses that DN when it looks for the user in LDAP. You can include the entry but leave the value blank; in this case, the server searches the entire LDAP tree for entries matching the filter.

- `FilterComps`

The `FilterComps` line tells the server to search based on the UID field in the certificate. If you configure all certificates issued by your root CA to have a UID field, this kind of search will always succeed.

- `verifycert`

The fourth line tells the server to verify that the certificate which the user has presented is in fact the certificate currently in the `usercertificate` attribute on the LDAP server. If you do not include this line, the server will check that the user is a legal user (that is, has access privileges to some particular part of the document root), but it will not check whether the user is using the right certificate.

If the user tries to present a revoked certificate to iPlanet Web Server, the server returns a 404 error. This error also occurs if the user does not have a certificate in the LDAP directory for any other reason, for example, if the certificate was issued at a time when the directory was unavailable for update.

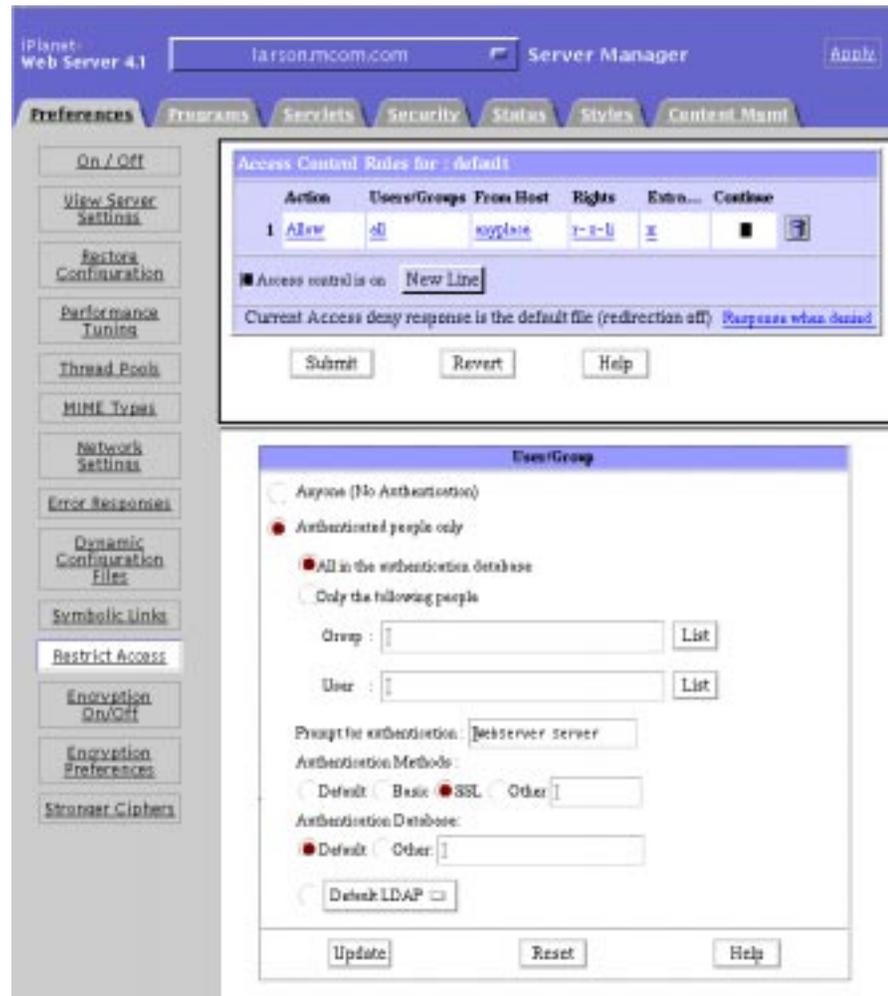
Modifying the Access Control Lists

You can configure the access control lists (ACLs) on iPlanet Web Server to allow only those who hold a valid certificate issued by your root CA to access the parts of the site that you designate as private.

To require client authentication for access to all or part of your site, follow these steps:

1. Open the Server Manager page for the server that you want to use SSL.
2. Click the Preferences tab if it is not showing.
3. In the left frame, click Restrict Access.
4. In the right panel, select Entire Server, or a subdirectory to which you want to restrict access.
5. Click Edit Access Control.

This page appears.



6. In the top pane under Users/Groups, select All.
7. In the bottom pane, select the following:
 - Authenticated people only

- Select either “All in the authentication database” or “Only the following people.” If you restrict access, select authorized users from the lists of specific users and groups
 - Under Authentication Method, select SSL
 - Under Authentication Database, select Default LDAP
8. Click Update.
 9. In the top pane, click Submit.

If you choose to require SSL authentication for particular users or groups, those users must obtain a client SSL certificate from your root CA and present it when they try to access the parts of the site you have chosen to protect.

Note There is a default setting for the entire iPlanet Web Server. iPlanet Web Server 4.x ships with defaults that allow anyone to read and publish anything on the server. You should consider your ACL needs and change the default setting accordingly. For detailed instructions on modifying users and groups and access privileges, refer to the documentation for iPlanet Web Server.

Testing Client Authentication

To test the configuration, you must start the server for which you have enabled SSL and attempt to access a page that you have protected.

To test the configuration, follow these steps:

1. Start the server, either from the iPlanet Web Server Administration Server or from the command line.
 - To start the server from the iPlanet Web Server Administration Server, go to the server’s Server Administration page, select the Preferences tab, enter the Trust Database password, and click On. Note that if you have *not* enabled SSL on Administration Server (as you just did for iPlanet Web Server), your password will go across the network unencrypted.
 - To start the server from the command line, open a command shell window, go to the installation directory, and run the `start` script for the new server instance. You must supply the key database password to

unlock the certificate and start up the new SSL server. Note that if you do not have a secure connection, your password will go across the network unencrypted. The script interaction looks like the following:

```
> pwd
/opt/netscape/suitespot/https-example-ssl
> ls
agents-db  conf_bk    db          restart    start
catalog    config     logs        rotate     stop
> ./start
Please enter password for Internal (software) token: <password>
iPlanet-WebServer-Enterprise/4.1 BB1
startup: listening to https://www.example.com, port 443 as nobody
```

2. Use your browser to access a page on the server that is part of a subdirectory to which you have restricted access. (See “Modifying the Access Control Lists” on page 306.)
3. If you are on the list of restricted users and if SSL has been successfully enabled, you will be asked to present your client SSL certificate from your root CA.

If you have problems, look at the error log files for Administration Server and iPlanet Web Server to determine what the problem might be.

Testing Client Authentication

Export Control Information

This appendix describes the cryptographic operations, key lengths, and cipher suites that have received US government approval for the export-controlled version of Certificate Management System. It does not describe the global version of Certificate Management System.

In most cases, the full-strength encryption version (or global version) of CMS is exportable outside of the United States of America. CMS has received “retail” status from the United States Department of Commerce Bureau of Export Administration; under new regulations, retail status makes it possible to export CMS with the same encryption and cryptographic features available in the US and Canada.

The global version of CMS is still not exportable to the following persons:

- end-users in nine prohibited destinations: Afghanistan (Taliban-controlled areas), Cuba, Iran, Iraq, Libya, North Korea, Serbia (except Kosovo), Sudan, and Syria
- persons prohibited by US law from receiving exports (including Denied Parties, Denied Entities, and Specially Designated Nationals)

Other conditions may apply which require that only the export-controlled version of CMS be made available to certain persons. For example, local laws may prohibit importing strong encryption, US law may change in the future, or CMS may come as part of a larger software bundle that does not receive retail status from the US government.

This appendix has the following sections:

- Approved Export Operations and Key Sizes (page 312)
- SSL Cipher Suite Profiles for Export (page 315)

Approved Export Operations and Key Sizes

Table E.1 lists all cryptographic operations available in the export-controlled version of Certificate Management System, and the key strength or algorithm strength allowed for each operation. The term *export-strength* is defined in “SSL Cipher Suite Profiles for Export” on page 315.

Table E.1 Approved export operations and key lengths

Description of cryptographic operation	Key length or algorithm strength
SSL connections: from end entity to Registration Manager [HTML forms]	export-strength SSL
SSL connections: from end entity to Registration Manager [CSR processors]	export-strength SSL
SSL connections: from Registration Manager to Certificate Manager	export-strength SSL
SSL connections: from Registration Manager to Data Recovery Manager	export-strength SSL
SSL connections: from Registration Manager to Directory	export-strength SSL
SSL connections: from Certificate Manager to Directory	export-strength SSL
SSL connections: from Netscape Console to Registration Manager, Certificate Manager, and Data Recovery Manager subsystems	export-strength SSL
Generation, verification, and storage of PQG parameters along with DSA certificates	P,G <= 4096 and Q=160 bits
Generation, signing (encryption), verifying (decryption), and storage of RSA keys for the purpose of signing/verifying X.509 digital certificates	key <= 4096 bits

Table E.1 Approved export operations and key lengths (Continued)

Description of cryptographic operation	Key length or algorithm strength
Generation, signing (encryption), verifying (decryption), and storage of DSA keys for the purpose of signing/verifying X.509 digital certificates	key <= 4096 bits
Generation, signing, verifying, and storage of RSA keys for the purpose of client authentication from Registration Manager to Certificate Manager subsystems	key <= 4096 bits
Generation, signing, verifying, and storage of RSA keys for the purpose of client authentication from Registration Manager to Data Recovery Manager subsystems	key <= 4096 bits
Generation, signing, verifying, and storage of RSA keys for the purpose of client authentication from Registration Manager subsystems to Directory	key <= 4096 bits
Generation, signing, verifying, and storage of DSA keys for the purpose of client authentication from Registration Manager to Certificate Manager subsystems	key <= 4096 bits
Generation, signing, verifying, and storage of DSA keys for the purpose of client authentication from Registration Manager to Data Recovery Manager subsystems	key <= 4096 bits
Generation, signing, verifying, and storage of DSA keys for the purpose of client authentication from Registration Manager subsystems to Directory	key <= 4096 bits
Generation, signing, verifying, and storage of RSA keys for the purpose of client authentication between Registration Manager, Certificate Manager, and Data Recovery Manager subsystems	key <= 4096 bits
Generation, signing, verifying, and storage of DSA keys for the purpose of client authentication between Registration Manager, Certificate Manager, and Data Recovery Manager subsystems	key <= 4096 bits
Generation, signing, verifying, and storage of RSA keys for the purpose of SSL server authentication of the Registration Manager	authentication key <= 4096 bits key exchange key <= 1024 bits

Approved Export Operations and Key Sizes

Table E.1 Approved export operations and key lengths (Continued)

Description of cryptographic operation	Key length or algorithm strength
Generation, signing, verifying, and storage of RSA keys for the purpose of SSL server authentication of the Certificate Manager	authentication key <= 4096 bits key exchange key <= 1024 bits
Generation, signing, verifying, and storage of RSA keys for the purpose of SSL server authentication of the Data Recovery Manager	authentication key <= 4096 bits key exchange key <= 1024 bits
Generation, signing, verifying, and storage of DSA keys for the purpose of SSL server authentication of the Registration Manager	authentication key <= 4096 bits key exchange key <= 1024 bits
Generation, signing, verifying, and storage of DSA keys for the purpose of SSL server authentication of the Certificate Manager	authentication key <= 4096 bits key exchange key <= 1024 bits
Generation, signing, verifying, and storage of DSA keys for the purpose of SSL server authentication of the Data Recovery Manager	authentication key <= 4096 bits key exchange key <= 1024 bits
Signature and verification of CMMF/CRMF messages by Certificate Manager, Registration Manager, and Data Recovery Manager using RSA algorithm	key <= 4096 bits
Signature and verification of CMMF/CRMF messages by Certificate Manager, Registration Manager, and Data Recovery Manager using DSA algorithm	key <= 4096 bits
Transport key for Data Recovery Manager: generation, storage, and verification of RSA key for the purpose of transport of end-entity private keys to the Data Recovery Manager (unwrap of keys)	key <= 4096 bits
Long-term storage key for Data Recovery Manager: generation, storage, encryption, and decryption using RSA key for the purpose of long term storage of end-entity private keys (wrap and unwrap of keys for storage and recovery)	key <= 4096 bits

Table E.1 Approved export operations and key lengths (Continued)

Description of cryptographic operation	Key length or algorithm strength
Bulk ciphers for use in encrypting key material for long term storage within Data Recovery Manager	DES-EDE3, RC2-128, RC2-40, DES
Bulk ciphers for use in encrypting key material for transport between Registration Manager and Data Recovery Manager	DES-EDE3, RC2-128, RC2-40, DES

SSL Cipher Suite Profiles for Export

Table E.2 summarizes the cipher suite profiles approved by the US government for use in the export-controlled version of Certificate Management System.

Table E.2 SSL 3.0 export-approved cipher suite profiles for Export

SSL Protocol Version	Cipher-key length (mode) and hash algorithm
SSL2	RC4-128-EXPORT40-WITH-MD5
	RC2-128-CBC-EXPORT40-WITH-MD5
SSL3	RSA-WITH-RC4-40-MD5
	RSA-EXPORT56-WITH-RC4-MD5
	RSA-WITH-RC2-CBC-40-MD5
	RSA-EXPORT56-WITH-RC2-CBC-MD5
	RSA-EXPORT-WITH-DES40-CBC-SHA
	RSA-EXPORT56-WITH-DES-CBC-SHA
	RSA-WITH-NULL-MD5
	RSA-WITH-NULL-SHA

SSL Cipher Suite Profiles for Export

Glossary

access control	The process of controlling who is allowed to do what. For example, access control to servers is typically based on an identity, established by a password or a certificate, and on rules regarding what that entity can do. See also access control list (ACL).
access control entry (ACE)	An access rule that specifies either (1) how subjects requesting access are to be identified or (2) what rights are allowed or denied for a particular subject or subjects. See access control list (ACL).
access control list (ACL)	A collection of access control entries that define a hierarchy of access rules to be evaluated when a server receives a request for access to a particular resource. See access control entry (ACE).
administrator	The person who installs and configures one or more CMS managers and sets up privileged users, or agents, for them. See also agent.
agent	A user who belongs to a group authorized to manage agent services for a CMS manager. See also Certificate Manager agent, Registration Manager agent, Data Recovery Manager agent.
agent services	1. Services that can be administered by a CMS agent via HTML pages served by the CMS manager for which the agent has been assigned the necessary privileges. 2. The HTML pages for administering such services.
attribute value assertion (AVA)	An assertion of the form <i>attribute = value</i> , where <i>attribute</i> consists of a tag, such as <code>o</code> (organization) or <code>uid</code> (user ID), and <i>value</i> consists of a value, such as "Netscape Communications Corp." or a login name. AVAs are used to form the distinguished name (DN) that identifies the subject of a certificate (called the subject name of the certificate).
authentication	Confident identification; that is, assurance that a party to some computerized transaction is not an impostor. Authentication typically involves the use of a password, certificate, PIN, or other information that can be used to validate identity over a computer network. See also password-based authentication, certificate-based authentication, client authentication, server authentication.

authentication module	A set of rules (implemented as a Java class) for authenticating an end entity, agent, administrator, or any other entity that needs to interact with a CMS manager. In the case of typical end-user enrollment, after the user has supplied the information requested by the enrollment form, the enrollment servlet uses an authentication module associated with that form to validate the information and authenticate the user's identity. See servlet.
authorization	Permission to access a resource controlled by a server. Authorization typically takes place after the ACLs associated with a resource have been evaluated by a server. See access control list (ACL).
automatic authentication	A way of configuring a CMS manager that allows automatic authentication for the purposes of end-entity enrollment, without human intervention. With this form of authentication, a certificate request that completes authentication module processing successfully is automatically approved for policy processing and certificate issuance.
bind DN	A user ID, in the form of a distinguished name (DN), used with a password to authenticate to Netscape Directory Server.
CA certificate	A certificate that identifies a certificate authority. See also certificate authority (CA), subordinate CA, root CA.
CA hierarchy	A hierarchy of CAs in which a root CA delegates the authority to issue certificates to subordinate CAs. Subordinate CAs can also expand the hierarchy by delegating issuing status to other CAs. See also certificate authority (CA), subordinate CA, root CA.
CA server key	The SSL server key of the server providing a CA service.
CA signing key	The private key that corresponds to the public key in the CA certificate. A CA uses its signing key to sign certificates and CRLs.
certificate	Digital data, formatted according to the X.509 standard, that specifies the name of an individual, company, or other entity (the subject name of the certificate) and certifies that a public key, which is also included in the certificate, belongs to that entity. A certificate is issued and digitally signed by a certificate authority (CA). A certificate's validity can be verified by checking the CA's digital signature using the techniques of public-key cryptography. To be trusted within a public-key infrastructure (PKI), a certificate must be issued and signed by a CA that is trusted by other entities enrolled in the PKI.

certificate authority (CA)	A trusted entity that issues a certificate after verifying the identity of the person or entity the certificate is intended to identify. A CA also renews and revokes certificates and generates CRLs. The entity named in the issuer field of a certificate is always a CA. Certificate authorities can be independent third parties (such as the CAs listed at https://certs.netscape.com/client.html) or a person or organization using certificate-issuing server software (such as Netscape Certificate Management System). Certificate Management System makes it possible to divide the role of a CA among one or more Registration Managers, which handle most or all interactions with certificate owners, and a Certificate Manager, which issues certificates.
certificate-based authentication	Authentication based on certificates and public-key cryptography. See also password-based authentication.
certificate chain	A hierarchical series of certificates signed by successive certificate authorities. A CA certificate identifies a certificate authority (CA) and is used to sign certificates issued by that authority. A CA certificate can in turn be signed by the CA certificate of a parent CA, and so on up to a root CA. Certificate Management System allows any end entity to retrieve all the certificates in a certificate chain.
Certificate Enrollment Protocol (CEP)	A certificate management protocol jointly developed by Cisco Systems and VeriSign, Inc. CEP is an early implementation of Certificate Management Messages over Cryptographic Message Syntax (CMC). CEP specifies how a device communicates with a CA, including how to retrieve the CA's public key, how to enroll a device with the CA, and how to retrieve a CRL. CEP uses PKCS #7 and PKCS #10. For more information about CEP, see http://www.cisco.com/warp/public/778/security/821_pp.htm .
certificate extensions	An X.509 v3 certificate contains an extensions field that permits any number of additional fields to be added to the certificate. Certificate extensions provide a way of adding information such as alternative subject names and usage restrictions to certificates. A number of standard extensions have been defined by the PKIX working group. Older versions of Netscape browsers and servers support Netscape-specific extensions that were required (mainly to indicate certificate usage) before standard extensions were defined.
certificate fingerprint	A one-way hash associated with a certificate. The number is not part of the certificate itself, but is produced by applying a hash function to the contents of the certificate. If the contents of the certificate changes, even by a single character, the same function produces a different number. Certificate fingerprints can therefore be used to verify that certificates have not been tampered with.

Certificate Management Messages over Cryptographic Message Syntax (CMC)	Message format used to convey a request for a certificate to a Registration Manager or Certificate Manager. A proposed standard from the Internet Engineering Task Force (IETF) PKIX working group. For detailed information, see http://www.ietf.org/internet-drafts/draft-ietf-pkix-cmc-02.txt .
Certificate Management Message Formats (CMMF)	Message formats used to convey certificate requests and revocation requests from end entities to a Registration Manager or Certificate Manager and to send a variety of information to end entities. A proposed standard from the Internet Engineering Task Force (IETF) PKIX working group. CMMF is subsumed by another proposed standard, Certificate Management Messages over Cryptographic Message Syntax (CMC). For detailed information, see http://www.ietf.org/internet-drafts/draft-ietf-pkix-cmmf-02.txt .
Certificate Manager	An independent CMS subsystem capable of acting as a stand-alone certificate authority. A Certificate Manager instance issues, renews, and revokes certificates, which it can publish along with CRLs to an LDAP directory. It can be configured to accept requests from end entities, Registration Managers, or both. When set up to work with a separate Registration Manager, the Certificate Manager processes requests and returns the signed certificates to the Registration Manager. See certificate authority (CA).
Certificate Manager agent	A user who belongs to a group authorized to manage agent services for a Certificate Manager. These services include the ability to access and modify (approve and reject) certificate requests and issue certificates.
Certificate Request Message Format (CRMF)	Format used for messages related to life-cycle management of X.509 certificates. This format is a subset of CMMF. See also Certificate Management Message Formats (CMMF). For detailed information, see ftp://ftp.isi.edu/in-notes/rfc2511.txt .
certificate revocation list (CRL)	As defined by the X.509 standard, a list of revoked certificates by serial number, generated and signed by a certificate authority (CA).
chain of trust	See certificate chain.
chained CA	See linked CA.
cipher	See cryptographic algorithm.
client authentication	The process of identifying a client to a server, for example, with a name and password or with a certificate and some digitally signed data. See certificate-based authentication, password-based authentication, server authentication.

client SSL certificate	A certificate used to identify a client to a server using the SSL protocol. See Secure Sockets Layer (SSL).
CMC	See Certificate Management Messages over Cryptographic Message Syntax (CMC).
CMMF	See Certificate Management Message Formats (CMMF).
CMS	See Netscape Certificate Management System (CMS), Cryptographic Message Syntax (CMS).
CMS instance	An instance of a CMS subsystem, comprising both code and data and treated as a discrete entity.
CMS subsystem	One of the three CMS Managers: Certificate Manager, Registration Manager, or Data Recovery Manager.
CMS window	A window that can be opened for any single CMS instance from within Netscape Console. A CMS window allows the CMS administrator to control configuration settings for the corresponding CMS instance.
configuration directory	A Directory Server instance that contains the configuration entries used by Netscape Console to track the servers in a server group.
CRL	See certificate revocation list (CRL).
CRMF	See Certificate Request Message Format (CRMF).
cross-certification	The exchange of certificates by two CAs in different certification hierarchies, or chains. Cross-certification extends the chain of trust so that it encompasses both hierarchies. See also certificate authority (CA).
cryptographic algorithm	A set of rules or directions used to perform cryptographic operations such as encryption and decryption.
Cryptographic Message Syntax (CMS)	The syntax used to digitally sign, digest, authenticate, or encrypt arbitrary messages, such as CMMF.
cryptographic module	See PKCS #11 module.
cryptographic service provider (CSP)	A cryptographic module that performs cryptographic services, such as key generation, key storage, and encryption, on behalf of software that uses a standard interface such as that defined by PKCS #11 to request such services.
CSP	See cryptographic service provider (CSP).

Data Recovery Manager	An optional, independent CMS subsystem that manages the long-term archival and recovery of RSA encryption keys for end entities. A Certificate Manager or Registration Manager can be configured to archive end entities' encryption keys with a Data Recovery Manager before issuing new certificates. The Data Recovery Manager is useful only if end entities are encrypting data (such as sensitive email) that the organization may need to recover someday. It can be used only with end entities that support dual key pairs—that is, two separate key pairs, one for encryption and one for digital signatures.
Data Recovery Manager agent	A user who belongs to a group authorized to manage agent services for a Data Recovery Manager, including managing the request queue and authorizing recovery operation using HTML-based administration pages.
Data Recovery Manager recovery agent	One of the m of n people who own portions of the storage key for the Data Recovery Manager.
Data Recovery Manager storage key	Special key used by the Data Recovery Manager to encrypt the end entity's encryption key (after it has been decrypted with the Data Recovery Manager's private transport key). The storage key never leaves the Data Recovery Manager.
Data Recovery Manager transport certificate	Certifies the public key used by an end entity to encrypt the entity's encryption key for transport to the Data Recovery Manager. The Data Recovery Manager uses the private key corresponding to the certified public key to decrypt the end entity's key before encrypting it with the Data Recovery Manager storage key. The Data Recovery Manager also uses the same private key to sign the proof of archival token it sends to the Registration Manager after storing an end entity's encryption key.
decryption	The unscrambling of data that has been encrypted. See encryption.
Data Encryption Standard (DES)	A FIPS-approved cryptographic algorithm required by FIPS 140-1 and specified by FIPS PUBS 46-2. DES, which uses 56-bit keys, is a standard encryption and decryption algorithm that has been used successfully throughout the world for more than 20 years. See also FIPS PUBS 140-1. For detailed information, see http://www.itl.nist.gov/div897/pubs/fip46-2.htm .
digital ID	See certificate.
digital signature	To create a digital signature, the signing software first creates a one-way hash from the data to be signed (such as a newly issued certificate). The one-way hash is then encrypted with the private key of the signer. The resulting digital signature is unique for each piece of data signed. Even a single comma added

to a message changes the digital signature for that message. Successful decryption of the digital signature with the signer's public key and comparison with another hash of the same data provides tamper detection. Verification of the certificate chain for the certificate containing the public key provides authentication of the signer. See also nonrepudiation, encryption.

Digital Signature Algorithm (DSA)	A FIPS-approved cryptographic algorithm specified by the Digital Signature Standard (DSS), FIPS PUBS 186. DSA is a standard algorithm used to create digital signatures. For detailed information, see http://www.itl.nist.gov/div897/pubs/fip186.htm .
distinguished name (DN)	A series of AVAs that identify the subject of a certificate. See attribute value assertion (AVA).
DSA	See Digital Signature Algorithm (DSA).
dual key pair	Two public-private key pairs--four keys altogether--corresponding to two separate certificates. The private key of one pair is used for signing operations, and the public and private keys of the other pair are used for encryption and decryption operations. Each pair corresponds to a separate certificate. See also encryption key, public-key cryptography, signing key.
eavesdropping	Surreptitious interception of information sent over a network by an entity for which the information is not intended.
encryption	The process of scrambling information in a way that disguises its meaning. See decryption.
encryption key	A private key used for encryption only. An encryption key and its equivalent public key, plus a signing key and its equivalent public key, constitute a dual key pair.
enrollment	The process of requesting and receiving an X.509 certificate for use in a public-key infrastructure (PKI). Also known as <i>registration</i> .
end entity	In a public-key infrastructure (PKI), a person, router, server, or other entity that uses a certificate to identify itself.
extensions field	See certificate extensions.
fingerprint	See certificate fingerprint.
FIPS PUBS 140-1	Federal Information Standards Publications (FIPS PUBS) 140-1 is a US government standard for implementations of cryptographic modules--that is, hardware or software that encrypts and decrypts data or performs other

cryptographic operations (such as creating or verifying digital signatures). Many products sold to the US government must comply with one or more of the FIPS standards. For detailed information, see <http://www.itl.nist.gov/div897/pubs/fip140-1.htm>.

firewall	A system or combination of systems that enforces a boundary between two or more networks.
impersonation	The act of posing as the intended recipient of information sent over a network. Impersonation can take two forms: spoofing and misrepresentation.
intermediate CA	A CA whose certificate is located between the root CA and the issued certificate in a certificate chain.
IP spoofing	The forgery of client IP addresses.
JAR file	A digital envelope for a compressed collection of files organized according to the Java archive (JAR) format.
Java archive (JAR) format	A set of conventions for associating digital signatures, installer scripts, and other information with files in a directory.
Java Cryptography Architecture (JCA)	The API specification and reference developed by Sun Microsystems for cryptographic services. For detailed information, see http://java.sun.com/products/jdk/1.2/docs/guide/security/CryptoSpec.html#Introduction
Java Development Kit (JDK)	Software development kit provided by Sun Microsystems for developing applications and applets using the Java programming language.
Java Native Interface (JNI)	A standard programming interface that provides binary compatibility across different implementations of the Java Virtual Machine (JVM) on a given platform, allowing existing code written in a language such as C or C++ for a single platform to bind to Java. For detailed information, see http://java.sun.com/products/jdk/1.2/docs/guide/jni/index.html .
Java Security Services (JSS)	A Java interface for controlling security operations performed by Netscape Security Services (NSS).
KEA	See Key Exchange Algorithm (KEA).
key	A large number used by a cryptographic algorithm to encrypt or decrypt data. A person's public key, for example, allows other people to encrypt messages intended for that person. The messages must then be decrypted by using the corresponding private key.

key exchange	A procedure followed by a client and server to determine the symmetric keys they will both use during an SSL session.
Key Exchange Algorithm (KEA)	An algorithm used for key exchange by the US Government.
Lightweight Directory Access Protocol (LDAP)	A directory service protocol designed to run over TCP/IP and across multiple platforms. LDAP is a simplified version of Directory Access Protocol (DAP), used to access X.500 directories. LDAP is under IETF change control and has evolved to meet Internet requirements.
linked CA	An internally deployed certificate authority (CA) whose certificate is signed by a public, third-party CA. The internal CA acts as the root CA for certificates it issues, and the third-party CA acts as the root CA for certificates issued by other CAs that are linked to the same third-party root CA. Also known as “chained CA” and by other terms used by different public CAs.
manual authentication	A way of configuring a CMS manager that requires human approval of each certificate request. With this form of authentication, a servlet forwards a certificate request to a request queue after successful authentication module processing. An agent with appropriate privileges must then approve each request individually before policy processing and certificate issuance can proceed.
MD5	A message digest algorithm that was developed by Ronald Rivest. See also one-way hash.
message digest	See one-way hash.
misrepresentation	The presentation of an entity as a person or organization that it is not. For example, a web site might pretend to be a furniture store when it is really just a site that takes credit-card payments but never sends any goods. Misrepresentation is one form of impersonation. See also spoofing.
Netscape Certificate Management System (CMS)	A highly configurable set of software components and tools for creating, deploying, and managing certificates. CMS comprises three major subsystems that can be installed in different CMS instances in different physical locations: Certificate Manager, Registration Manager, and Data Recovery Manager.
Netscape Console	The Java application used to set up and manage Netscape servers.
Netscape Security Services (NSS)	A set of libraries designed to support cross-platform development of security-enabled communications applications. Applications built using the NSS libraries support the Secure Sockets Layer (SSL) protocol for authentication, tamper detection, and encryption, and the PKCS #11 protocol for cryptographic token

interfaces. Netscape uses NSS to support these features in a wide range of products, including Certificate Management System. NSS is also available separately as a software development kit.

nonrepudiation	The inability by the sender of a message to deny having sent the message. A digital signature provides one form of nonrepudiation.
object signing	A technology that allows software developers to sign Java code, JavaScript scripts, or any kind of file and allows users to identify the signers and control access by signed code to local system resources.
object-signing certificate	A certificate whose associated private key is used to sign objects using the technology known as object signing.
one-way hash	A number of fixed length generated from data of arbitrary length with the aid of a hashing algorithm. The number (also called a message digest) has two characteristics: (1) It is unique to the hashed data. Any change in the data, even deleting or altering a single character, results in a different value. (2) The content of the hashed data cannot, for all practical purposes, be deduced from the hash.
password-based authentication	Confident identification by means of a name and password. See also authentication, certificate-based authentication.
PKCS #7	The public-key cryptography standard that governs signing and encryption.
PKCS #10	The public-key cryptography standard that governs certificate requests.
PKCS #11	The public-key cryptography standard that governs cryptographic tokens such as smart cards.
PKCS #11 module	A driver for a cryptographic device that provides cryptographic services, such as encryption and decryption, via the PKCS #11 interface. A PKCS #11 module (also called a <i>cryptographic module</i> or <i>cryptographic service provider</i>) can be implemented in either hardware or software. A PKCS #11 module always has one or more slots, which may be implemented as physical hardware slots in some form of physical reader (for example, for smart cards) or as conceptual slots in software. Each slot for a PKCS #11 module can in turn contain a token, which is the hardware or software device that actually provides cryptographic services and optionally stores certificates and keys. Netscape provides a built-in PKCS #11 module with Certificate Management System.
PKCS #12	The public-key cryptography standard that governs key portability.

policy module	A rule (implemented as a Java class) that validates the contents of a certificate request for that rule and formulates the contents of the certificate to be issued.
private key	One of a pair of keys used in public-key cryptography. The private key is kept secret and is used to decrypt data encrypted with the corresponding public key.
proof-of-Archival (POA)	Data signed with the private Data Recovery Manager transport key that contains information about an archived end-entity key, including key serial number, name of the Data Recovery Manager, subject name of the corresponding certificate, and date of archival. The signed proof-of-archival data is the response returned by the Data Recovery Manager to the Registration Manager or Certificate Manager after a successful key archival operation. See also Data Recovery Manager transport certificate.
public key	One of a pair of keys used in public-key cryptography. The public key is distributed freely and published as part of a certificate. It is typically used to encrypt data sent to the public key's owner, who then decrypts the data with the corresponding private key.
public-key cryptography	A set of well-established techniques and standards that allow an entity to verify its identity electronically or to sign and encrypt electronic data. Two keys are involved: a public key and a private key. A public key is published as part of a certificate, which associates that key with a particular identity. The corresponding private key is kept secret. Data encrypted with the public key can be decrypted only with the private key.
public-key infrastructure (PKI)	The standards and services that facilitate the use of public-key cryptography and X.509 v3 certificates in a networked environment.
RC2, RC4	Cryptographic algorithms developed for RSA Data Security by Rivest. See also cryptographic algorithm.
registration	See enrollment.
Registration Manager	An optional, independent CMS subsystem that performs tasks involving end entities, such as enrollment or renewal, on behalf of a Certificate Manager. The Registration Manager can be configured to process requests and approve them either manually (that is, with the aid of a human being) or automatically (based entirely on customizable policies and procedures). After the Registration Manager approves requests, it typically forwards them to the Certificate Manager, which processes them and returns the issued certificates to the Registration Manager. The Registration Manager then distributes the certificates to the end entities and (typically) publishes them to the appropriate directory.

Registration Manager agent	A user who belongs to a group authorized to manage agent services for a Registration Manager, including the ability to access and modify (approve and reject) certificate requests.
root CA	The certificate authority (CA) with a self-signed certificate at the top of a certificate chain. See also CA certificate, subordinate CA.
RSA algorithm	Short for Rivest-Shamir-Adleman, a public-key algorithm for both encryption and authentication. It was developed by Ronald Rivest, Adi Shamir, and Leonard Adleman and introduced in 1978.
RSA key exchange	A key-exchange algorithm for SSL based on the RSA algorithm.
sandbox	A Java term for the carefully defined limits within which Java code must operate.
Secure Sockets Layer (SSL)	A protocol that allows mutual authentication between a client and server and the establishment of an authenticated and encrypted connection. SSL runs above TCP/IP and below HTTP, LDAP, IMAP, NNTP, and other high-level network protocols.
server authentication	The process of identifying a server to a client. See also client authentication.
server group	The servers in a server root directory managed by a single instance of Netscape Administration Server.
server root	The directory used to store CMS and other Netscape Server binaries that make up a server group.
server SSL certificate	A certificate used to identify a server to a client using the Secure Sockets Layer (SSL) protocol.
servlet	Java code that handles a particular kind of interaction with end entities on behalf of a CMS manager. For example, certificate enrollment, renewal, revocation, and key recovery requests are each handled by separate servlets.
SHA-1	Secure Hash Algorithm, a hash function used by the US Government.
signature algorithm	A cryptographic algorithm used to create digital signatures. Certificate Management System supports the MD5 and SHA-1 signing algorithms. See also cryptographic algorithm, digital signature.

signing certificate	A certificate whose public key corresponds to a private key used to create digital signatures. For example, Certificate Manager must have a signing certificate whose public key corresponds to the private key it uses to sign the certificates it issues. A Registration Manager must have a signing certificate whose public key corresponds to the private key it uses to sign the certificate requests it sends to the Certificate Manager.
signing key	A private key used for signing only. A signing key and its equivalent public key, plus an encryption key and its equivalent public key, constitute a dual key pair.
single sign-on	<ol style="list-style-type: none"> 1. In CMS, a password that simplifies the way you sign on to Certificate Management System by storing the passwords for the internal database and tokens. Each time you log on, you're required to enter just this single password. 2. The ability for a user to log in once to a single computer and be authenticated automatically by a variety of servers within a network. Partial single sign-on solutions can take many forms, including mechanisms for automatically tracking passwords used with different servers. Certificates support single sign-on within a public-key infrastructure (PKI). A user can log in once to a local client's private-key database and thereafter, as long as the client software is running, rely on certificate-based authentication to access each server within an organization that the user is allowed to access.
slot	The portion of a PKCS #11 module (implemented in either hardware or software) that contains a token.
smart card	A small device, typically about the size of a credit card, that contains a microprocessor and is capable of storing cryptographic information (such as keys and certificates) and performing cryptographic operations. Smart cards implement some or all of the PKCS #11 interface.
spoofing	The act of pretending to be someone else. For example, a person can pretend to have the email address <code>jdoe@netscape.com</code> , or a computer can identify itself as a site called <code>www.netscape.com</code> when it is not. Spoofing is one form of impersonation. See also misrepresentation, impersonation.
SSL	See Secure Sockets Layer (SSL).
subject	The entity identified by a certificate. In particular, the subject field of a certificate contains a subject name that uniquely describes the certified entity.
subject name	A distinguished name (DN) that uniquely describes the subject of a certificate.

subordinate CA	A certificate authority whose certificate is signed by another subordinate CA or by the root CA. See CA certificate, root CA.
symmetric encryption	An encryption method that uses the same cryptographic key to encrypt and decrypt a given message.
tamper detection	A mechanism ensuring that data received in electronic form has not been tampered with; that is, that the data received entirely corresponds with the original version of the same data.
token	A hardware or software device that is associated with a slot in a PKCS #11 module. It provides cryptographic services and optionally stores certificates and keys.
tree hierarchy	The hierarchical structure of an LDAP directory.
trust	Confident reliance on a person or other entity. In a public-key infrastructure (PKI), trust refers to the relationship between the user of a certificate and the certificate authority (CA) that issued the certificate. If you trust a CA, you can generally trust valid certificates issued by that CA.
virtual private network (VPN)	A way of connecting geographically distant divisions of an enterprise. The VPN allows the divisions to communicate over an encrypted channel, allowing authenticated, confidential transactions that would normally be restricted to a private network.

Index

A

- Administration Server
 - and demo 80
 - NT setup 148
 - Unix setup 145
- administrator/agent, initial enrollment 93–95, 226–228
- agent enrollment 230–231
- authentication
 - client, with Enterprise Server 3.x 291–309
 - decisions for deployment 137
- authentication modules 29–30, 30–43, 55–56, 74
- authorityKeyIdentifier 256, 276, 283

B

- basicConstraints 257, 283

C

- CA decisions, for deployment 128–132
 - CA renewal 131–132
 - distinguished name 128
 - extensions 130–131
 - root versus subordinate 130
 - signing certificate 129
 - signing key 129
- CA signing certificate 129
 - configuration of 153–156, 166–??
- CEP 44–45, 46, 50, 74
- certificateIssuer 279
- certificate life-cycle management 33, 48–53, 62
- Certificate Management System (CMS)
 - access to subsystems 50

- architecture 70–74
 - command-line utilities 68–70
 - identifier 145, 148
 - overview of 22–29
 - servlets 29
 - standards supported by 74–76
- Certificate Manager
 - configuration of 151–157
 - Data Recovery Manager and 123–128
 - Data Recovery Manager and Registration Manager and 125–128
 - demo and 81
 - features of 62
 - installed by itself 119–120
 - introduced 24
 - Registration Manager and 120–122
- certificatePolicies 258
- certificates
 - Certificate Manager 135
 - Data Recovery Manager 136
 - extensions for 247–284
 - for subsystems, summarized 134–136
 - installing 285–289
 - life-cycle management 48–53
 - management formats and protocols 74–75
 - Registration Manager 136
 - SSL server, for CMS subsystems 135
 - X.509 specification 76
- cipher suites for export 315
- client authentication, with Enterprise Server 3.x 291–309
- CMC 75
- CMMF 75
- CMS. *See* Certificate Management System, Cryptographic Message Syntax
- CMS instances

- ports and 138–140
- server groups and 118, 138–140
- command-line utilities 68–70
- configuration directory
 - demo and 80
 - NT setup 146, 147–148
 - Unix setup 144
- configuration directory server
 - Unix setup 142
- conventions used in this book 15
- cRLDistributionPoints 259
- CRLNumber 276
- CRLs
 - Certificate Manager support for 63
 - extensions for 275–280
- CRMF 74
- Cryptographic Message Syntax (CMS) 75

D

- database, internal CMS 81
- Data Recovery Manager
 - Certificate Manager and 123–128
 - Certificate Manager and Registration Manager and 125–128
 - configuration of 159–164
 - features of 64
 - introduced 24
 - recovery agents for 163–164
 - transport certificate 160–163
- deltaCRLIndicator 277
- demo 77–113
 - first user certificate for 93–95
 - installation of 77–113
 - Installation Wizard and 89–93
 - overview of 80–84
 - passwords for 83–84
 - port numbers for 82
 - software installed for 82
 - using 95–113
 - using an LDAP directory with 103–113
 - verifying installation 96–101
- deployment planning 117–140
 - authentication decisions 137
 - CA decisions 128–132
 - CA renewalCA renewal 131–132
 - distinguished name 128
 - extensions 130–131
 - root versus subordinate 130
 - signing certificate 129
 - signing key 129
 - certificate decisions
 - Certificate Manager 135
 - Data Recovery Manager 136
 - Registration Manager 136
 - enrollment scenarios 33–47
 - firewall considerations 34
 - hardware token decisions 132–??
 - LDAP publishing decisions 133–134
 - policy decisions 137–138
 - port assignments 138–140
 - SSL server certificate decisions 135
 - storage key 136
 - subsystem certificate decisions 134–136
 - topology decisions 118–128
- distinguished name (DN)
 - for CA 128, 129
 - for CA signing certificate 154
 - for Data Recovery Manager transport certificate 160
 - for Registration Manager signing certificate 159
- downloading certificates 285–289
- DSA 129

E

- end entities
 - enrollment, steps in 30–32
 - enrollment scenarios for 33–47
 - forms for 52
 - life-cycle management and 48–53
- enrollment, initial administrator/agent 226–228
- enrollment scenarios 33–47
 - custom authentication, customer database 36
 - custom authentication, Kerberos 40–41

- firewall considerations 34
 - manual authentication 38–39
 - PIN-based authentication 42–43
 - routers 46–47
 - VPNs 44–45
 - Enterprise Server 3.x, using SSL with 291–309
 - event-driven notifications 61
 - export control information 311–315
 - extensions 247–284
 - adding to certificates 282
 - authorityKeyIdentifier 256, 276, 283
 - basicConstraints 257, 283
 - CA certificates and 155–156, 282–284
 - CAs and 130–131
 - certificateIssuer 279
 - certificatePolicies 258
 - CMS policy modules for 58
 - cRLDistributionPoints 259
 - CRLNumber 276
 - deltaCRLIndicator 277
 - extKeyUsage 261
 - holdInstructionCode 279
 - invalidityDate 280
 - issuerAltName 263, 278
 - issuingDistributionPoint 278
 - keyUsage 264
 - nameConstraints 267
 - netscape-cert-type 281, 283
 - netscape-comment 282
 - Netscape-defined 280–284
 - policyConstraints 269
 - policyMappings 270
 - privateKeyUsagePeriod 271
 - reasonCode 280
 - recommendations for usage 249–253
 - SSL server certificate 169–170
 - subjectAltName 272
 - subjectDirectoryAttributes 273
 - subjectKeyIdentifier 274
 - transport certificate 161
 - X.509 certificate, summarized 253–275
 - X.509 CRL, summarized 275–280
 - extKeyUsage 261
- F**
- FIPS PUBS 140-1 75
 - firewalls 34
 - fonts used in this book 15
- G**
- gateway
 - agent, for demo 92
 - end user, for demo 92
- H**
- hardware requirements for CMS installation 78
 - hardware token decisions, for deployment 132–??
 - holdInstructionCode 279
- I**
- installation 173–231
 - additional instances 229
 - demo 77–113
 - first user certificate for 93–95
 - Installation Wizard and 89–93
 - NT installation script for 87–89
 - overview of 80–84
 - passwords for 83–84
 - Unix installation script for 85–87
 - using 95–113
 - verifying 96–101
 - hardware requirements 78
 - location of
 - NT setup 145
 - Unix setup 142
 - overview 174
 - port considerations 138–140
 - software requirements 78
 - Solaris requirements 78, 80
 - system requirements 78–80
 - Windows NT requirements 79
 - wizard 182–224
 - worksheet 141–171

installation script
information requested by 142–149
NT
complete instructions 179–182
running for demo 87–89
worksheet for 145–149

Unix
complete instructions 176–179
running for demo 85–87
worksheet for 142–145

Installation Wizard
initial configuration steps 149–151
procedures for using 182–225
running for demo 89–93

installing certificates 285–289

instances, CMS
agents for additional 230–231
creating additional 229

internal CMS database 81

invalidityDate 280

IP addresses, and port assignments 140

issuerAltName 263, 278

issuingDistributionPoint 278

J

Java/JNI 73

JDK 1.1.6 73

job scheduler 61

JSS 73

K

KEYGEN tag 76

key length 129

keyUsage 264

L

LDAP 76

LDAP directory
configuration, demo and 80

DN pattern for authentication 104
internal CMS database, demo and 81
publishing decisions 133–134
testing authentication with 103–113

M

migrating from Certificate Server 1.x 152–153,
165–166, 233–245

N

nameConstraints 267

netscape-cert-type 281, 283

netscape-comment 282

Netscape Console
demo and 80
starting Installation Wizard from 182

notifications, event-driven 61

NSS 73

P

PKCS #10 76

PKCS #11 71–73, 76

PKCS #7 76

PKI. *Seedistinguished name (DN).*

PKI. *See installation script.*

PKI. *See Public-Key Infrastructure.*

PKIX 75

policyConstraints 269

policyMappings 270

policy modules 29–32, 56–60
decisions for deployment 137–138

port numbers
assignment of 138–140
for demo 82
IP addresses and 140

privateKeyUsagePeriod 271

Public-Key Infrastructure (PKI) 23

R

reasonCode 280

Registration Manager

 Certificate Manager and 120–122

 Certificate Manager and Data Recovery

 Manager and 125–128

 configuration of 157–159

 features of 62

 introduced 24

root versus subordinate CA 130

RSA 129

S

server certificate 167–171

server groups 118

servlets, CMS 29

setup script 113

signing algorithms 63

signing certificate

 CA 129, 153–157, 166–??

 Registration Manager 157–159

signing key, for CA 129

single sign-on password 171

software requirements for CMS installation 78

Solaris

 requirements for installation 80

Solaris requirements for installation 78

SSL 76

 cipher suites approved for export 315

 server certificate 167–171

 using with Enterprise Server 291–309

storage key, for Data Recovery Manager 136

subjectAltName 272

subjectDirectoryAttributes 273

subjectKeyIdentifier 274

subject name 141

subsystem certificate decisions 134–136

subsystem certificate decisions, for deployment

 Certificate Manager 135

 Data Recovery Manager 136

 SSL server 135

system requirements for CMS installation 78–80

T

terms used in this book 15

topology decisions, for deployment 118–128

transport certificate, for Data Recovery

 Manager 160–163

types used in this book 15

U

user/group directory

 NT setup 146

user/group directory server

 Unix setup 143

utilities, command-line 68–70

W

Windows NT, requirements for installation 79

X

X.509 certificates 76

