

Installation and Setup Guide

Sun™ ONE Certificate Server

4.7

September 2002
816-5548-10
Second Edition

Copyright © 2002 Sun Microsystems, Inc. All rights reserved.

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

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 Microsystems, Inc. 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.

Some pre-existing portions:

Copyright © 1998,1999 by Jef Poskanzer <jef@acme.com>. All rights reserved. Copyright © 1996 by Jef Poskanzer <jef@acme.com>. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) "HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 2002 Sun Microsystems, Inc. Tous droits réservés.

Sun, Sun Microsystems, le Sun logo, et iPlanet sont des marques dposes ou des marques dposes registre de Sun Microsystems, Inc. aux Etats-Unis et d'autres pays.

Le produit dé crit 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 Sun Microsystems, Inc., 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	23
What's in This Guide	23
What You Should Already Know	26
Conventions Used in This Guide	27
Where to Go for Related Information	29
Part 1 Overview and Demo Installation	31
Chapter 1 Introduction to Certificate Management System	33
Overview of Key Features	34
Flexible end-entity registration services framework	38
System Overview	42
Public-Key Infrastructure	43
CMS Subsystems or Managers	44
Certificate Manager	45
Registration Manager	47
Data Recovery Manager	48
Online Certificate Status Manager	49
Basic System Configuration	50
Plug-in Modules	55
Authentication Plug-in Modules	55
Policy Plug-in Modules	57
Job Plug-In Modules	61
Mapper and Publisher Plug-in Modules	62
Event-Driven Notifications	65
Auxiliary Components	65
Command-Line Utilities	65
CMS SDK	65
Entry Points for Various Types of Users	66

Agent Services Interface	68
Certificate Manager Agent Services	69
Registration Manager Agent Services	70
Data Recovery Manager Agent Services	71
Online Certificate Status Manager Agent Services Interface	71
End-Entity Services Interface	72
System Architecture	73
PKCS #11	75
NSS	76
JSS and the Java/JNI Layer	76
Middleware/Java 2 Layers	76
Authentication and Policy Modules	77
Standards Summary	77
Certificate Management Formats and Protocols	77
Security and Directory Protocols	78
Chapter 2 Certificate Enrollment and Life-Cycle Management	81
Steps in End-Entity Enrollment	81
Some Enrollment Scenarios	84
Firewall Considerations	84
Extranet/E-Commerce: Acme Sales Corp.	85
Enrolling Existing Customers	86
Enrolling New Customers	87
Enrolling Extranet Users	89
PIN Registration: Atlas Manufacturing	91
VPN Client Enrollment and Revocation	93
Router Enrollment and Revocation	96
End Entities and Life-Cycle Management	98
Life-Cycle Management Formats and Protocols	98
Access to Subsystems	99
HTML Forms for End Users	101
Netscape Personal Security Manager	102
Chapter 3 Default Demo Installation	105
System Requirements	106
Operating System and Software Required	106
Platform Requirements	106
Overview of the Default Demo	108
Demo Passwords	111
Installing the Default Demo	112
Step 1. Run the Installation Script — UNIX	112
Step 1. Run the Installation Script—Windows NT	114

Step 2. Run the Installation Wizard	123
Step 3. Get the First User Certificate	136
Enrolling for the First Agent Certificate	136
If You Need the First Agent Form Again	138
Using the Default Demo	139
Verify the Installation	139
Viewing Issued Certificates From the Agent Gateway	140
Enrolling for a Certificate From the End-Entity Gateway	141
Finding and Approving a Certificate Request	142
Setting Your Browser to Use the Agent Certificate	143
Testing Your New Certificate	143
Create a Policy	144
Configuring an RSA Key Length Policy	144
Use an LDAP Directory	146
Step 1. Enable Directory-Based Authentication	147
Step 2. Add a User to the Directory	148
Step 3. Enroll with Directory-Based Authentication	150
Publish Certificates to an LDAP Directory	151
Configure the Publishing Destination	152
Set Rules for Publishing Certificates	154
Update the Publishing Directory	155
Send Renewal Reminders	157
Configuring a Mail Server for Certificate Management System	158
Configuring Certificate Management System to Send Renewal Reminders	158

Part 2 Planning and Installation 163

Chapter 4 Planning Your Deployment	165
Topology Decisions	166
Server Groups and CMS Instances	166
Single Certificate Manager	167
Certificate Manager and Registration Manager	168
Certificate Manager and Data Recovery Manager	170
Certificate Manager, Data Recovery Manager, and Registration Manager	172
Cloned Certificate Manager	174
Certificate Authority Decisions	175
CA's Distinguished Name	175
CA Signing Key Type and Length	176
CA Signing Certificate's Validity Period	176
Self-Signed Root Versus Subordinate CA	176
CAs and Certificate Extensions	177

CA Certificate Renewal or Reissuance	178
Cryptographic Token Decisions	179
Publishing Decisions	179
Publishing to Certificates and CRLs to Files	180
Publishing to Certificates and CRLs to a Directory	180
Publishing CRLs to the Online Certificate Status Manager	181
Subsystem Certificate Decisions	182
SSL Server Certificates	182
Certificate Manager Certificates	182
Registration Manager Certificates	183
Data Recovery Manager Certificate and Storage Key	184
Online Certificate Status Manager Certificates	184
Authentication Decisions	185
Policy Decisions	185
Deployment Strategy and Port Assignments	186
Chapter 5 Installation Worksheet	189
Information for UNIX Installation Script	190
Installation Location	190
Configuration Directory Server	190
User/Group Directory Server	191
Configuration Directory Settings	191
Administration Server Information	192
Certificate Management System Identifier	193
Information for NT Installation Script	193
Installation Directory	193
Configuration Directory Server	193
User/Group Directory Server	194
Configuration Directory Settings	195
Configuration Directory Server Administrator	195
Directory Server Administration Domain	195
Directory Manager Settings	195
Administration Server Port	196
Certificate Management System Identifier	196
Initial Configuration	196
Internal Database	197
Administrator	197
Subsystems	197
Remote Certificate Manager	198
Remote Data Recovery Manager	198
Network Configuration	199
Certificate Manager Configuration	199
CA Signing Certificate	199

CA's Serial Number Range	199
Key-Pair Information for CA Signing Certificate	200
Subject Name for CA Signing Certificate	200
Validity Period for CA Signing Certificate	201
Extensions for CA Signing Certificate	201
CA Signing Certificate Request	202
Registration Manager Configuration	203
Registration Manager Signing Certificate Request	203
Key-Pair Information for Registration Manager Signing Certificate	203
Subject Name for Registration Manager Signing Certificate	204
Registration Manager Signing Certificate Issuer	204
Data Recovery Manager Configuration	205
Transport Certificate	205
Key-Pair Information for Transport Certificate	205
Subject Name for Transport Certificate	206
Validity Period for Transport Certificate	206
Extensions for Transport Certificate	207
Transport Certificate Request	208
Storage Key and Recovery Agent Configuration	208
Storage Key Creation	208
Data Recovery Scheme—1	208
Data Recovery Scheme—2	209
Online Certificate Status Manager Configuration	209
Online Certificate Status Manager Signing Certificate Request	209
Key-Pair Information for Online Certificate Status Manager Signing Certificate	210
Subject Name for Online Certificate Status Manager Signing Certificate	210
Online Certificate Status Manager Signing Certificate Issuer	211
Cloned Certificate Manager Configuration	211
CA Signing Certificate	212
CA's Serial Number Range	212
Cloned Key and Certificate Material	212
SSL Server Key and Certificate	213
SSL Server Certificate Configuration	213
SSL Server Certificate	213
Key-Pair Information for SSL Server Certificate	213
Subject Name for SSL Server Certificate	214
Validity Period for SSL Server Certificate	214
Extensions for SSL Server Certificate	215
SSL Certificate Request	216
Single Sign-On Password	216
Chapter 6 Installing Certificate Management System	217
Installation Overview	217

Installation Stages	218
Before You Begin the Installation	219
Stage 1. Running the Installation Script	221
Running the Installation Script on UNIX	221
Running the Installation Script on Windows NT	224
Stage 2. Running the Installation Wizard	227
Installing the Certificate Manager as a Root CA	229
Installing the Certificate Manager as a Subordinate CA	232
Installing a Standalone Registration Manager	244
Installing a Standalone Data Recovery Manager	255
Installing an Online Certificate Status Manager	266
Stage 3. Enrolling for Administrator/Agent Certificate	277
Agent Certificate for a Certificate Manager	277
Agent Certificate for Other CMS Managers	280
Stage 4. Further Configuration Options	283
Stage 5. Creating Additional Instances or CA Clones	284
Chapter 7 Installing and Uninstalling CMS Instances	285
Installing Multiple CMS Instances	286
Cloning a Certificate Manager	288
Step 1. Before You Begin	289
Step 2. Create Instances for Clone CAs	291
Installing Clone CA in Master CA's Server Group	291
Installing Clone CA in a Different Server Group	292
Installing Clone CA on a Separate Host	293
Step 3. Shutdown the Master CA	293
Step 4. Copy Master CA's Certificate and Key Database	294
Step 5. Start the Master CA	294
Step 6. Configure the Clone CA	294
Step 8. Establish Trust Between Master CA and Clone CAs	295
Step A. Locate the Master CA's SSL Server Certificate	296
Step B. Create a Privileged-User Entry for Clone CAs	298
Step 9. Test Clone-Master Connection	302
Step A. Request a Certificate from the Clone CA	302
Step B. Approve the Request	303
Step C. Download the Certificate to the Browser	303
Step D. Revoke the Certificate	304
Step E. Check Master CA's CRL for the Revoked Certificate	304
Step 10. Use Master CA's Agent Certificate in Clone CAs	305
Viewing Instance Information	306
Changing the Name of an Instance	308
Removing an Instance From a System	309
Uninstalling Certificate Management System	311

Uninstalling From the Command Line	311
Uninstalling by Using the Windows NT Add/Remove Programs Utility	311
Upgrading From Version 4.2 SP2 to Version 4.7	313
The CMS Migration Tool	313
Known Issues and Workarounds	313
Before You Begin	313
Running the Migration Tool on Unix	314
Running the Migration Tool on Windows	315
Upgrading to Version 4.2 SP2	316
Chapter 8 Starting and Stopping CMS Instances	321
Starting Certificate Management System	322
Required Start-up Information	322
Configuring the Server to Start Without the Single Sign-On Password	323
Configuring the Server to Read the Single Sign-on Password From a File	324
Starting From iPlanet Console	327
Starting From the Command Line	328
Starting From the Windows NT Services Panel	329
Stopping Certificate Management System	330
Stopping From iPlanet Console	330
Stopping From the Command Line	331
Stopping From the Windows NT Services Panel	332
Restarting Certificate Management System	332
Restarting From the CMS Window	332
Restarting From the Command Line	333
Checking System Status	334
Attending to an Unresponsive Server	335
CMS Watchdog Process	335
Password Cache	336
Password-Quality Checker	337
Chapter 9 Administration Tasks and Tools	339
iPlanet Console	340
Console Tab	340
Users and Groups Tab	341
iPlanet Administration Server	342
Starting Administration Server	343
Shutting Down Administration Server	344
Logging In to iPlanet Console	344
The CMS Window	346
Tasks Tab	347
Configuration Tab	347

Status Tab	350
Logging In to the CMS Window	351

Part 3 Configuration 353

Chapter 10 CMS Configuration	355
Effects of Installation Type on Configuration	355
Duplicating Configuration From One Instance to Another	357
Locating the Configuration File	358
Modifying the Configuration	359
Changing the Configuration From the CMS Window	359
Changing the Configuration by Editing the Configuration File	359
Guidelines for Editing the Configuration File	360
Sample Configuration File	363
Road Map to Configuring Subsystems	376
Step 1. Check Which Subsystems are Installed in the Instance	376
Step 2. Check the Port Numbers	376
Step 3. Verify Key Pair and Certificates	376
Step 4. Set up Privileged Users	377
Step 5. Customize End-Entity and Agent Forms	377
Step 6. Setup Authentication for End Users	377
Step 7: Enable Event-Driven Notifications	378
Step 8. Schedule Jobs	378
Step 9. Set up Policies	378
Step 10. Set up Publishing	379
Step 11. Set up Key Archival and Recovery	379
Step 12. Set up Logging	379
Step 13. Plan for Backing up CMS Configuration and Data	380
Chapter 11 Setting Up Ports	381
CMS Ports	381
Remote Administration Port	382
Agent Port	383
End-Entity Ports	383
Configuring Port Numbers	384
Step 1. Specify the Port Number	384
Step 2: Specify IP Addresses	387
Chapter 12 Setting Up Internal Database	389
Internal Database	389
Configuring the Internal Database	390

Step 1. Identify the Directory Server Instance	391
Step 2. Restrict Access to the Internal Database	392
Chapter 13 Managing Privileged Users and Groups	395
Privileged-User Types and Responsibilities	396
Administrators	396
Agents	397
Agent's Certificate for SSL Client Authentication	399
Revocation Status Checking of Agent Certificates	402
Trusted Managers	405
Subsystems That Can Function as Trusted Managers	405
Connectors for Linking Trusted Managers	406
Trusted Manager's Certificate for SSL Client Authentication	408
Groups and Their Privileges	409
Group for Administrators	409
Groups for Agents	410
Group for Certificate Manager Agents	410
Group for Registration Manager Agents	411
Group for Data Recovery Manager Agents	411
Group for Online Certificate Status Manager Agents	412
Group for Trusted Managers	412
Setting Up Privileged Users	413
Setting Up Administrators	413
Step 1. Find the Required Information	413
Step 2. Add the Information to the Internal Database	413
Setting Up Agents	416
Setting up Agents Using the Automated Process	416
Setting up Agents Using the Manual Process	417
Setting Up Trusted Managers	423
Setting up Trusted Managers Using the Automated Process	423
Setting Up a Registration Manager as a Trusted Manager	424
Setting Up a Certificate Manager as a Trusted Manager	436
Changing Privileged-User Information	444
Changing a Privileged User's Login Information	444
Changing a Privileged User's Certificate	445
Changing Members in a Group	446
Deleting a Privileged User	448
Chapter 14 Managing CMS Keys and Certificates	449
Keys and Certificates for the Main Subsystems	450
Certificate Manager's Key Pairs and Certificates	451
CA Signing Key Pair and Certificate	451

wTLS CA Signing Certificate	452
OCSP Signing Key Pair and Certificate	452
CRL Signing Key Pair and Certificate	453
SSL Server Key Pair and Certificate	455
Remote Administration Server Certificate	457
Registration Manager's Key Pairs and Certificates	459
Signing Key Pair and Certificate	459
SSL Server Key Pair and Certificate	459
Remote Administration Server Certificate	460
Data Recovery Manager's Key Pairs and Certificates	460
Transport Key Pair and Certificate	461
Storage Key Pair	461
SSL Server Key Pair and Certificate	462
Remote Administration Server Certificate	462
Online Certificate Status Manager's Key Pairs and Certificates	463
OCSP Signing Key Pair and Certificate	463
SSL Server Key Pair and Certificate	463
Remote Administration Server Certificate	464
Tokens for Storing CMS Keys and Certificates	464
Internal Token	465
External Token	465
Installing External Tokens	465
Installing Level 2 External Tokens	466
Installing Level 3 External Tokens	468
Managing Tokens Used by the Subsystems	476
Viewing Tokens	476
Changing a Token's Password	477
Hardware Cryptographic Accelerators	477
Certificate Setup Wizard	478
Using the Wizard to Request a Certificate	479
Step 1. Select the Operation	479
Step 2. Choose the Certificate	480
Step 3. Specify the Key-Pair Information	482
Step 4. Specify the Subject Name for the Certificate	484
Step 5. Specify the Validity Period	485
Step 6. Specify Extensions	486
Step 7. Copy the Certificate Signing Request	488
Step 8. Check the Certificate Request Status	492
Using the Wizard to Install a Certificate or Certificate Chain	493
Data Formats for Installing Certificates and Certificate Chains	494
Step 1. Select the Operation	495
Step 2. Select the Certificate or Certificate Chain	496
Step 3. Specify the Location of the Certificate	497

Step 4. View the Certificate or Certificate Chain	499
Step 5. Install the Certificate or Certificate Chain	499
Step 6. Verify the Certificate Status	500
Configuring the Server's Security Preferences	500
Configuring the Server to Use Separate SSL Server Certificates	500
Step 1. Get the Required SSL Server Certificates	501
Step 2: Update the Configuration	501
Getting an SSL Client Certificate for a Subsystem	502
Setting Up Cipher Preferences for SSL Communications	504
SSL Ciphers Supported in Certificate Management System	504
Configuring the Server to Use Specific Ciphers	506
Getting New Certificates for the Subsystems	507
Step 1. Plan for the New Certificate	507
Step 2. Request the New Certificate	510
Step 3. Install the New Certificate	510
Step 4. Deploy the New Certificate	511
Deploying Certificate Manager's CA Signing Certificate	511
Deploying Registration Manager's Signing Certificate	512
Deploying Data Recovery Manager's Transport Certificate	513
Deploying a Subsystem's SSL Server Certificate	514
Renewing Certificates for the Subsystems	515
Step 1. Plan for Certificate Renewal	516
Step 2. Renew the Existing Certificate	517
Step 3. Install the Renewed Certificate	518
Step 4. Deploy the Renewed Certificate	518
Deploying Certificate Manager's Renewed CA Signing Certificate	519
Deploying Registration Manager's Renewed Signing Certificate	519
Deploying Data Recovery Manager's Renewed Transport Certificate	520
Deploying a Subsystem's Renewed SSL Server Certificate	522
Step 5. Restart the Server	522
Managing the Certificate Database	523
Viewing the Certificate Database Content	523
Deleting a Certificate From the Certificate Database	525
Changing the Trust Settings of a CA Certificate	526
Installing a New CA Certificate in the Certificate Database	528
Installing a CA Certificate Chain in the Certificate Database	529
Chapter 15 Setting Up End-User Authentication	531
Introduction to Authentication	531
Privileged-User Authentication	532
Authentication of Administrators	532
Authentication of Agents	534
End-Entity Authentication	537

Authentication of End Entities During Certificate Enrollment	537
Authentication of End Users During Certificate Renewal	537
Authentication of End Users During Certificate Revocation	540
Configuring Authentication for End-User Enrollment	545
Step 1. Before You Begin	545
Step 2. Set Up the Directory for PIN-Based Enrollment	547
Step A. Check the Directory for User Entries	547
Step B. Update the Directory	547
Step C. Prepare the Input File	549
Step D. Run the Command Without the Write Option	549
Step E. Check the Output File	550
Step F. Run the Command Again with the Write Option	550
Step 3. Enable the AttributePresentConstraints Policy	550
Step 4: Add an Authentication Instance	553
Step 5. Set Up the Enrollment Interface	559
Step A. Associate the Authentication Instance With the Enrollment Form	559
Step B. Customize the Form	560
Step C. Hook Up the Certificate-Based Enrollment Form	560
Step D. Remove Unwanted Enrollment Options	562
Step 6. Enable End-Entity Interaction	563
Enabling End-Entity Interaction with a Certificate Manager	563
Enabling End-Entity Interaction with a Registration Manager	566
Step 7. Turn on Automated Notification	567
Step 8. Test Your Authentication Setup	567
Step 9. Deliver PINs to End Users	568
Managing Authentication Instances	569
Deleting an Authentication Instance	569
Modifying an Authentication Instance	570
Managing Authentication Plug-in Modules	572
Registering an Authentication Module	572
Deleting an Authentication Module	574
Chapter 16 Setting Up Automated Notifications	575
Automated Notifications	575
Notifications of Certificate Issuance to End Entities	576
Notification of New Request in Queue	577
Customizing Notification Messages	578
Templates for Event-Triggered Notifications	578
Customizing Message Templates	580
Tokens Available in Message Templates	581
Tokens for Certificate Issuance Notifications to End Entities	581
Tokens for Rejection Notifications to End Entities	582
Tokens for Request In Queue Notification Messages	583

Configuring a Subsystem to Send Notifications	583
Step 1. Before You Begin	584
Step 2. Turn On Certificate-Issuance Notification	584
Step 3. Turn on Request in Queue Notification	585
Step 4. Verify Mail Server Settings	586
Step 5. Test Your Configuration	587
Chapter 17 Scheduling Automated Jobs	589
Configuring a Subsystem to Run Automated Jobs	589
Step 1. Before You Begin	590
Step 2. Modify Existing Jobs	590
Step 3. Delete Unwanted Jobs	593
Step 4. Add New Jobs	593
Step 5. Schedule the Frequency	597
Step 6. Verify Mail Server Settings	598
Step 7. Test Your Configuration	599
Managing Job Plug-in Modules	599
Registering a Job Module	600
Deleting a Job Module	601
Chapter 18 Setting Up Policies	603
Introduction to Policy	603
What Is Policy?	604
Policy Rules	605
Types of Policy Rules	605
Using Predicates in Policy Rules	606
Expression Support for Predicates	606
Attributes for Predicates	608
Policy Processor	612
Configuring Policy Rules for a Subsystem	613
Step 1. Before You Begin	614
Step 2. Modify Existing Policy Rules	614
Step 3. Delete Unwanted Policy Rules	618
Step 4. Add New Policy Rules	618
Step 5. Reorder Policy Rules	623
Step 6. Restart the Server	624
Step 7. Test Policy Configuration	624
Step A. Enroll for a Certificate	624
Step B. Approve the Request	625
Step C. Check the Certificate Details	625
Using JavaScript for Policies	626
Managing Policy Plug-in Modules	626

Registering a Policy Module	626
Deleting a Policy Module	628
Chapter 19 Setting Up LDAP Publishing	629
Publishing of Certificates to a Directory	629
Timing of Directory Updates	631
Directory Update Process	633
Directory Synchronization	634
Publishing of CRLs	634
What's a CRL?	635
Reasons for Revoking a Certificate	636
Revocation Checking by Netscape Clients	637
Revocation Checking by iPlanet Servers	637
Publishing of CRLs to an LDAP Directory	637
CRL Issuing Points	638
Configuring a Certificate Manager to Publish Certificates and CRLs	639
Step 1. Before You Begin	640
Step 2. Set Up the Directory for Publishing	641
Step A. Verify the Directory Schema	642
Step B. Add an Entry for the CA	643
Step C. Identify an Entry That Has Write Access	645
Step D. Verify Entries for End Entities	645
Step E. Specify the Directory Authentication Method	646
Step F. Modify the Certificate Mapping File	656
Step G. Restart Directory Server	660
Step 3. Configure the Certificate Manager to Publish Certificates	660
Step A. Modify the Default Mappers, Publishers, and Publishing Rules	660
Step B. Add Mappers, Publishers, and Publishing Rules	666
Step 4. Configure the Certificate Manager to Publish CRLs	672
Step A. Specify CRL Details	673
Step B. Set the CRL Extensions	675
Step C. Create a Mapper for the CRL	676
Step D. Create a Publisher for the CRL	677
Step E. Create a Publishing Rule for the CRL	679
Step 5. Identify the Publishing Directory	680
Step 6. Test Certificate and CRL Publishing	682
Step A. Decide a Directory Entry for Requesting a Certificate	683
Step B. Request a Certificate	683
Step C. Approve the Request	683
Step D. Download the Certificate to the Browser	684
Step E. Check if the Directory Has the Certificate	684
Step F. Revoke the Certificate	685
Step G. Check the Directory for the CRL	686

Manually Updating Certificates and CRLs in a Directory	686
Manually Updating Certificates in the Directory	687
Manually Updating the CRL in the Directory	688
Chapter 20 Publishing Certificates and CRLs to a File	691
Configuring Certificate Manager to Publish to Files	691
Step 1. Before You Begin	692
Step 2. Configure the Certificate Manager	693
Step A. Create a Publisher for the File	693
Step B. Create Publishing Rules for Certificates	695
Step C. Create a Publishing Rule for CRLs	697
Step D. Specify CRL Details	698
Step E. Set the CRL Extensions	700
Step F. Make Sure Publishing is Enabled	702
Step 3. Test Publishing	702
Step A. Request a Certificate	702
Step B. Approve the Request	703
Step C. Download the Certificate to the Browser	704
Step D. Check the File for the Certificate	704
Step E. Revoke the Certificate	706
Step F. Check the File for the CRL	707
Managing Mapper and Publisher Plug-in Modules	709
Registering a Mapper or Publisher Module	709
Deleting a Mapper or Publisher Module	711
Chapter 21 Setting Up an OCSP Responder	713
What's an OCSP-Compliant PKI Setup?	714
How to Get an OCSP Responder?	716
How Certificate Manager's OCSP-Service Feature Works	716
How Online Certificate Status Manager Works	717
How to Get OCSP-Compliant Clients?	718
Setting Up a Certificate Manager with OCSP Service	719
Step 1. Before You Begin	719
Step 2. Install OCSP-Compliant Client	720
Step 3. Enable Certificate Manager's HTTP Port	721
Step 4. Enable Certificate Manager's OCSP Service	723
Step 5. Configure Certificate Manager for Extensions	724
Step 6. Restart the Certificate Manager	726
Step 7. Test Your CA's OCSP Service Setup	727
Step A. Turn On Revocation Checking in the Browser	727
Step B. Request a Certificate	728
Step C. Approve the Request	728

Step D. Download the Certificate to the Browser	729
Step E. Make Sure the CA is Trusted by the Browser	729
Step F. Verify the Certificate in the Browser	730
Step G. Check the Status of Certificate Manager's OCSP Service	730
Step H. Revoke the Certificate	731
Step I. Verify the Certificate in the Browser	731
Step J. Check the Certificate Manager's OCSP Service Status Again	731
Setting Up a Remote OCSP Responder	732
Step 1. Before You Begin	733
Step 2. Install an OCSP-Compliant Client	734
Step 3. Identify the CA to the OCSP Responder	735
Step 4. Configure the Certificate Manager to Publish CRLs	737
Step A. Specify CRL Format and Publishing Interval	738
Step B. Set the CRL Extensions	739
Step C. Create a Publisher for the CRL	740
Step D. Create a Publishing Rule for the CRL	742
Step E. Make Sure Publishing is Enabled	744
Step 5. Configure Certificate Manager for Required Extension Policies	745
Step 6. Configure the Online Certificate Status Manager	747
Step 7. Restart the Certificate Manager	751
Step 8. Restart the Online Certificate Status Manager	752
Step 9. Verify Certificate Manager and Online Certificate Status Manager Connection	752
Step 10. Test Your OCSP Responder Setup	753
Step A. Turn On Revocation Checking	753
Step B. Request a Certificate	754
Step C. Approve the Request	754
Step D. Download the Certificate to the Browser	755
Step E. Make Sure the CA is Trusted by the Browser	755
Step F. Verify the Certificate in the Browser	756
Step G. Check the Status of Online Certificate Status Manager	756
Step H. Revoke the Certificate	757
Step I. Verify the Certificate in the Browser	757
Step J. Check the Online Certificate Status Manager Status Again	757
Chapter 22 Setting Up Key Archival and Recovery	759
PKI Setup for Key Archival and Recovery	759
Clients That Can Generate Dual Key Pairs	760
Data Recovery Manager	760
Forms for Users and Key Recovery Agents	761
Key Archival Process	761
Why You Should Archive Keys	761
Where the Keys are Stored	762
How Key Archival Works	763

Key Recovery Process	765
Key Recovery Agents and Their Passwords	765
Secret Sharing of Storage Key Password	765
Interface for the Key Recovery Process	766
Local Versus Remote Key Recovery Authorization	767
How Agent-Initiated Key Recovery Works	768
Key Recovery Agent Scheme	771
Changing the Key Recovery Agent Scheme	771
Changing Key Recovery Agents' Passwords	773
Configuring Key Archival and Recovery Process	775
Step 1. Set Up the Key Archival Process	775
Step A. Deploy Clients That Can Generate Dual Key Pairs	776
Step B. Connect the Enrollment Authority and the Data Recovery Manager	776
Step C. Customize the Certificate Enrollment Form	777
Step D. Configure Key Archival Policies	782
Step 2. Set Up the Key Recovery Process	782
Step A. Verify the m of n Scheme	783
Step B. Facilitate the Key Recovery Agents to Change the Passwords	783
Step C. Determine the Authorization Mode for Key Recovery	783
Step D. Customize the Key Recovery Form	784
Step E. Configure Key Recovery Policies	784
Step 3. Test Your Key Archival and Recovery Setup	784
Step A. Test Your Key Archival Setup	784
Step B. Verify the Key	786
Step C. Delete the Certificate	786
Step D. Test Your Key Recovery Setup	787
Step D. Restore the Key in the Browser's Database	788
Chapter 23 Managing CMS Logs	789
Introduction to Logs	789
Logs Maintained by the Server	790
Services That Are Logged	791
Log Levels (Message Categories)	792
Log File Locations	793
Log File Naming Conventions	794
Active Log File Naming Convention	794
Rotated Log File Naming Convention	794
Buffered Versus Unbuffered Logging	794
Rotation of Log Files	795
Timing of Log File Rotation	795
Location of Rotated Log Files	796
Deletion of Log Files	796
How to Conserve Disk Space	796

Timing of Log File Deletion	796
Configuring CMS Logs	797
Step 1. Before You Begin	797
Step 2. Modify the Existing Listeners	797
Step 3. Delete Unwanted Listeners	799
Step 4. Create New Listeners	800
Monitoring CMS Logs	803
Monitoring System Logs	804
Monitoring Error Logs	806
Monitoring Audit Logs	808
Using System Tools for Monitoring the Server (Windows NT Only)	811
Logging to Windows NT Event Log	811
Using Event Viewer	811
Avoiding Event Log From Getting Filled	812
Archiving of Rotated Log Files	813
Signing Log Files	814
Managing Log Modules	816
Registering a Log Module	816
Deleting a Log Module	817

Part 4 Issuing and Managing Certificates 819

Chapter 24 Issuing and Managing Server Certificates	821
Certificate Issuance to Servers	821
How the Manual Server Enrollment Process Works	822
Getting Server SSL Certificates for iPlanet Servers	824
Getting Certificates for Version 3.x Servers	824
Step 1. Generate the Server Certificate Request	825
Step 2. Submit the Server Certificate Request	826
Step 3. Install Your Server's SSL Certificate	827
Step 4. Accept a CA as Trusted in Your Server	827
Step 5. Verify Your Server's SSL and CA Certificates	829
Getting Certificates for iPlanet Servers	829
Renewal of Server Certificates	831
Revocation of Server Certificates	831
Chapter 25 Setting Up CEP Enrollment	833
CEP Enrollment	833
CEP Enrollment Using the Script	834
Setting up CEP Enrollment Manually	835
Step 1. Set up the Directory for Publishing Certificates and CRLs	836

Step 2. Configure the Certificate Manager for Publishing Certificates and CRLs	837
Step 3. Set Up Automated Enrollment	840
Step 4. Set Up Multiple CEP Services	844
Certificate Issuance to Routers or VPN Clients	845
Step 1. Before You Begin	846
Step 2. Generate the Key Pair for the Router	847
Step 3. Request the CA's Certificate	848
Step 4. Submit the Certificate Request to the CA	848
Example	849

Part 5 Appendixes 853

Appendix A Certificate Download Specification	855
Data Formats	855
Binary Formats	855
Text Formats	856
Importing Certificate Chains	857
Importing Certificates into Netscape Communicator	857
Importing Certificates into iPlanet Servers	858
Object Identifiers	858

Appendix B Using SSL with iPlanet Web Server, Enterprise Edition 4.x	861
Creating a New Server	862
Obtaining a Server Certificate	863
Creating a Trust Database	863
Submitting a Certificate Signing Request	864
Importing the Certificate	866
Enabling SSL on the Server	869
Enabling Encryption on the Server	869
Trusting the Root CA Certificate	870
Enabling Client Authentication for All Requests	871
Specifying the Authentication Directory	871
Note for CGI Programmers	873
Modifying the Configuration File	873
Modifying the Access Control Lists	875
Testing Client Authentication	877

Appendix C Export Control Information	879
Approved Export Operations and Key Sizes	880
SSL Cipher Suite Profiles for Export	882

Appendix D Smart Card Login with Windows 2000	883
Overview	883
Part 1. Set Up the Windows 2000 Environment	883
Part 2. Configuring Certificate Server 4.7	884
Part 3. Customization Notes	893
3d. Certificate Verification	895
About otherName in Subject Alt Name Extension	895
Glossary	897
Index	913

About This Guide

The *Installation and Setup Guide* explains how to install, configure, and maintain iPlanet Certificate Management Server (CMS), and use it for issuing and managing certificates to various end entities, such as web browsers (users), servers, Virtual Private Network (VPN) clients, and Cisco™ routers.

NOTE Sun™ ONE Certificate Server was previously known as iPlanet™ Certificate Management System. The product was renamed shortly before the launch of this 4.7 release.

The late renaming of this product has resulted in a situation where the new product name is not fully integrated into the shipping product. In particular, you will see the product referenced as iPlanet Certificate Management Server (CMS) within the product GUI and within the product documentation. For this release, please consider iPlanet Certificate Management Server and Sun™ ONE Certificate Server as interchangeable names for the same product.

This preface has the following sections:

- What's in This Guide (page 23)
- What You Should Already Know (page 26)
- Conventions Used in This Guide (page 27)
- Where to Go for Related Information (page 29)

What's in This Guide

This guide covers topics that are listed below. You should use this guide in conjunction with the other CMS documentation, such as the ones that explain all the plug-ins and command-line tools that are provided for Certificate Management System. For a complete list of CMS documentation, see section “Where to Go for Related Information” on page 29.

- “About This Guide” Describes what’s covered in this guide, what you should already know, and where to look for more information.

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, “Certificate Enrollment and Life-Cycle Management” Provides sample deployment scenarios.
- Chapter 3, “Default Demo Installation” Describes how to set up a simple pilot that demonstrates the basic capabilities of a Certificate Manager.

Part 2, “Planning and Installation”

- Chapter 4, “Planning Your Deployment” Reviews basic decisions you should make as you plan your initial deployment.
- Chapter 5, “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 6, “Installing Certificate Management System” Describes the procedure for installing CMS subsystems on the basis of the information collected in Chapter 5.
- Chapter 7, “Installing and Uninstalling CMS Instances” Describes how to create multiple instances, delete unwanted instances, clone instances, upgrade from a previous CMS version, and so on.
- Chapter 8, “Starting and Stopping CMS Instances” Describes how to start, restart, and stop CMS instances.

Part 3, “Configuration”

- Chapter 9, “Administration Tasks and Tools” Explains the GUI-based administration tools, iPlanet Console and CMS window.
- Chapter 10, “CMS Configuration” Shows a sample configuration file and explains the rules for editing the configuration file.
- Chapter 11, “Setting Up Ports” Describes various ports used by a CMS instance and explains how to set up these ports.
- Chapter 12, “Setting Up Internal Database” Describes the function of internal database and explains how to set it up.

- Chapter 13, “Managing Privileged Users and Groups” Describes privileged users, their access rights, and how to create them for managing a CMS instance.
- Chapter 14, “Managing CMS Keys and Certificates” Describes keys and certificates used by a CMS instance and explains how to renew and reissue them. Also provides information on installing hardware tokens.
- Chapter 15, “Setting Up End-User Authentication” Describes authentication methods for different types of CMS users, and explains how to configure a Certificate Manager or Registration Manager to use a specific authentication method for end-user enrollment.
- Chapter 16, “Setting Up Automated Notifications” Describes how to enable the automated notification feature—such as notifying agents when a request gets queued and notifying users when their certificates are issued—to ease administration overheads.
- Chapter 17, “Scheduling Automated Jobs” Describes how to schedule jobs that automatically perform certain certificate-related tasks at regular intervals—such as removing expired certificates from the directory and notifying users before their certificates expire—to ease administration overheads.
- Chapter 18, “Setting Up Policies” Describes how to configure a CMS manager to use policy rules that govern the formulation and issuance of certificate content, such as key size, signing algorithm, validity period, extensions, and so on.
- Chapter 19, “Setting Up LDAP Publishing” Provides an overview of LDAP publishing and describes how to configure a Certificate Manager to publish certificates and CRLs to an LDAP directory.
- Chapter 20, “Publishing Certificates and CRLs to a File” Describes how to configure a Certificate Manager to publish certificates and CRLs to files for importing to other repositories.
- Chapter 21, “Setting Up an OCSP Responder” Provides an overview of OCSP-compliant PKI setup and describes how to set up an OCSP-compliant PKI setup.
- Chapter 22, “Setting Up Key Archival and Recovery” Describes how to archive end users’ encryption private keys and recover them, if there’s a need.
- Chapter 23, “Managing CMS Logs” Describes how to enable logging, use logs to monitor the server’s activities, and archive log files.

Part 4, “Issuing and Managing Certificates”

- Chapter 24, “Issuing and Managing Server Certificates” Describes how to issue SSL server certificates to other servers and manage the certificates.
- Chapter 25, “Setting Up CEP Enrollment” Describes how to configure the server to issue router and VPN client certificates.

Part 5, “Appendixes”

- Appendix A, “Certificate Download Specification” Describes the data formats used by Netscape Communicator 4.x for installing certificates.
- Appendix B, “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 C, “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.

What You Should Already Know

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

This guide assumes that you

- Are familiar with the basic concepts of public-key cryptography and the Secure Sockets Layer (SSL) protocol.
 - SSL cipher suites
 - The purpose of and major steps in the SSL handshake
- Understand the concepts of intranet, extranet, and the Internet security and the role of digital certificates in a secure enterprise. These include the following topics:
 - Encryption and decryption
 - Public keys, private keys, and symmetric keys
 - Significance of key lengths

- Digital signatures
- Digital certificates, including various types of digital certificates
- The role of digital certificates in a public-key infrastructure (PKI)
- Certificate hierarchies

If you are new to these concepts, we recommend you read the security-related documents available online at this URL:

http://docs.sun.com/db?p=coll/S1_nsCMS_42_Resources

You may also refer to the security-related appendixes (Appendix D and Appendix E) of the accompanying manual, *Managing Servers with iPlanet Console*.

- Are familiar with the role of iPlanet Console in managing iPlanet servers. Otherwise, see the accompanying manual, *Managing Servers with iPlanet Console*.
- Are reading this guide in conjunction with the documentation listed in section “Where to Go for Related Information” on page 29.

Conventions Used in This Guide

The following conventions are used in this guide:

- `Monospaced font`—This typeface is used for any text that appears on the computer screen or text that you should type. It’s also used for filenames, functions, and examples.

Example: `Server Root` is the directory where the CMS binaries are kept.

- **Italic**—Italic type is used for emphasis, book titles, and glossary terms.

Example: This control depends on the access permissions the *superadministrator* has set up for you.

- Text within “quotation marks”—Indicates cross-references to other topics within this guide.

Example: For more information, see “Issuing a Certificate to a New User” on page 154.

- **Boldface**—Boldface type is used for various UI components such as captions and field names, and the terminology explained in the glossary.

Example:

Rotation frequency. From the drop-down list, select the interval at which the server should rotate the active error log file. The available choices are Hourly, Daily, Weekly, Monthly, and Yearly. The default selection is Monthly.

- **Monospaced []**—Square brackets enclose commands that are optional.

Example: `PrettyPrintCert <input_file> [<output_file>]`

`<input_file>` specifies the path to the file that contains the base-64 encoded certificate.

`<output_file>` specifies the path to the file to write the certificate. This argument is optional; if you don't specify an output file, the certificate information is written to the standard output.

- **Monospaced <>**—Angle brackets enclose variables or placeholders. When following examples, replace the angle brackets and their text with text that applies to your situation. For example, when path names appear in angle brackets, substitute the path names used on your computer.

Example: Using Netscape Communicator 4.7 or later, enter the URL for the administration server: `http://<hostname>:<port_number>`

- **/**—A slash is used to separate directories in a path. If you use the Windows NT operating system, you should replace / with \ in paths.

Example: Except for the Security Module Database Tool, you can find all the other command-line utilities at this location: `<server_root>/bin/cert/tools`

- **Sidebar text**—Sidebar text marks important information. Make sure you read the information before continuing with a task.

Examples:

NOTE You can use iPlanet Console only when Administration Server is up and running.

CAUTION A caution note documents a potential risk of losing data, damaging software or hardware, or otherwise disrupting system performance.

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 (which you specify 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 iPlanet Console*

Provides background information on basic cryptography concepts and the role of iPlanet Console. To view the HTML version of this guide, open this file:

```
<server_root>/manual/en/admin/help/contents.htm
```

- *CMS Installation and Setup Guide* (this guide)

Describes how to plan for, install, and administer Certificate Management System. To access the installation and configuration information from within the CMS Installation Wizard or from the CMS window (within iPlanet Console), click any help button.

To view the HTML version of this guide, open this file:

```
<server_root>/manual/en/cert/setup_guide/contents.htm
```

- *CMS Plug-Ins Guide*

Provides detailed reference information on CMS plug-ins. To access this information from the CMS window within iPlanet Console, click any help button.

To view the HTML version of this guide, open this file:

```
<server_root>/manual/en/cert/plugin_guide/contents.htm
```

- *CMS Command-Line Tools Guide*

Provides detailed reference information on CMS tools.

To view the HTML version of this guide, open this file:

```
<server_root>/manual/en/cert/tools_guide/contents.htm
```

- *CMS Customization Guide*

Provides detailed reference information on customizing the HTML-based agent and end-entity interfaces.

To view the HTML version of this guide, open this file:

`<server_root>/manual/en/cert/custom_guide/contents.htm`

- *CMS Agent's Guide*

Provides detailed reference information on CMS agent interfaces. To access this information from the Agent Services pages, click any help button.

To view the HTML version of this guide, open this file:

`<server_root>/cert-<instance_id>/web/agent/manual/agent_guide/contents.htm`

- 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.

To view the HTML version of this guide, open this file:

`<server_root>/cert-<instance_id>/web/ee/manual/ee_guide/contents.htm`

NOTE 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 for 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, check this site: http://docs.sun.com/?p=coll/S1_s1CertificateServer_47

Overview and Demo Installation

Chapter 1, “Introduction to Certificate Management System”

Chapter 2, “Certificate Enrollment and Life-Cycle Management

Chapter 3, “Default Demo Installation”

Introduction to Certificate Management System

This chapter introduces iPlanet Certificate Management Server (CMS), a highly configurable set of software components and tools for creating, deploying, and managing certificates. Based on open standards for certificate management, Certificate Management System leverages iPlanet Directory Server and iPlanet Console to provide a complete, scalable, high-performance certificate management solution for extranets and intranets.

Whether you are looking for a security solution for your enterprise or setting up an independent certificate authority (CA) service, Certificate Management System offers a robust, customizable, and scalable foundation for your public-key infrastructure (PKI).

The chapter has the following sections:

- Overview of Key Features (page 34)
- System Overview (page 42)
- Auxiliary Components (page 65)
- Entry Points for Various Types of Users (page 66)
- System Architecture (page 73)
- Standards Summary (page 77)

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 26.

Overview of Key Features

Certificate Management System has many core features:

Support for open standards

With its support for open standards, Certificate Management System gives organizations confidence that they will be able to communicate within a heterogeneous computing environment. Specifically, Certificate Management System does the following:

- Formulates, signs, and issues industry-standard X.509 version 3 public-key certificates; version 3 certificates include extensions that make it easy to include organization-defined attributes. This means that you can use these certificates for extranet and Internet authentication as well.

For details on setting extensions in certificates, see Chapter 18, “Setting Up Policies.”

- Supports issuance of Wireless Transport Layer Security (wTLS)-compliant certificates for use with wireless applications.
- Supports RSA public-key algorithm for signing and encryption, DSA public-key algorithm for signing, and MD2, MD5, and SHA-1 for hashing.
- Supports signature key lengths of up to 1024 bits (DSA) and 4096 (RSA) on both hardware and software tokens. For details, see Appendix C, “Export Control Information.”
- Supports multiple message formats, such as KEYGEN/SPAC, CRMF/CMMF, CRS/CEP/SCEP, and PKCS #10 and CMC for certificate requests. All requests are delivered to Certificate Management System over HTTP or HTTPS; in the case of CRS/CEP/SCEP protocol, the delivery method is always over HTTP. For a description of the acronyms, see “Standards Summary” on page 77.
- Supports certificate formats that encompass certificates for SSL-based client and server authentication, secure Multipurpose Internet Mail Extensions (S/MIME) message signing and encryption, object signing, VPN clients, and Cisco™ routers.
- Supports generation and publication of CRLs conforming to X.509 version 1 and 2.
- Publishes certificates and certificate revocation lists (CRLs) to the any LDAP-compliant directory over LDAP and HTTP/HTTPS connections. For more information, see Chapter 19, “Setting Up LDAP Publishing.”

- Publishes certificates and CRLs to a flat file for importing into other resources. For example, the sample code for Flat File CRL and certificate publisher can be customized to store certificates and CRLs in an Oracle RDBMS™. For more information, see Chapter 20, “Publishing Certificates and CRLs to a File.”
- Publishes CRLs to an online validation authority (or OCSP responder), enabling real-time verification of certificates by OCSP-compliant clients. For more information, see Chapter 21, “Setting Up an OCSP Responder.”

Separate subsystems for certificate and key operations

Certificate Management System includes four servers, the *Certificate Manager*, *Registration Manager*, *Data Recovery Manager*, and *Online Certificate Status Manager*.

- The Certificate Manager functions as the certificate authority (CA); it is the entity named in the issuer field of a certificate. The Certificate Manager can sign and revoke certificates and generate CRLs. It can accept certificate requests directly from end entities and via Registration Managers to which it has delegated certain certificate management functions, such as authentication of an end entity. The Certificate Manager also maintains a database of issued certificates so that it can track renewal, expiration, and revocation.
- The Registration Manager is an optional component in the PKI; it is a subordinate server to which a Certificate Manager can delegate some certificate management functions. For example, a Registration Manager may act as a front end to a Certificate Manager, performing tasks such as end-entity authentication and formulation of the certificate request for the Certificate Manager.
- The Data Recovery Manager is an optional component in the PKI. It provides key archival and recovery services for end users’ encryption private keys.
- The Online Certificate Status Manager is an optional, but important component in the PKI. It enables real-time verification of certificates issued by one or more Certificate Managers.

For an overview of these subsystems, see “CMS Subsystems or Managers” on page 44.

Single CA supports multiple registration authorities

Certificate Management System lets you separate the registration process from the certificate-signing process with the help of Registration Managers. You can run multiple Registration Managers remotely, all reporting to a single Certificate Manager, to verify user identities and process certificate signing requests. The remote Registration Managers forward their completed and approved requests to the Certificate Manager for it to sign and issue the certificate automatically.

The Certificate Manager's ability to support multiple Registration Managers makes it more scalable and also adds an extra layer of security for the CA. For example, you can set a policy that requires all clients to go through a remote Registration Manager, and then have the remote Registration Manager route all client requests to the Certificate Manager located inside a firewall.

For more information, see "Trusted Managers" on page 405.

Ability to function as both a root and a subordinate CA in a CA hierarchy

Certificate Management System can function as a *root* or *parent CA*; in this case, the server signs its own CA signing key as well as other CA signing keys, enabling you to create your own CA hierarchy. You can also install the server to function as a *subordinate CA*; in this case, the server gets its CA signing key signed by another CA in an existing CA hierarchy.

For details on installing the Certificate Manager as a root or subordinate CA, see Part 2, "Planning and Installation."

Ability to function as a linked CA

Certificate Management System can function as a *linked CA*, chaining up to many third-party or public CAs for validation; this provides cross-company trust, so applications can verify certificate chains outside the company certificate hierarchy. You chain a Certificate Manager to a third-party CA by requesting the Certificate Manager's *CA signing certificate* from the third-party CA.

For details on installing the Certificate Manager as a linked CA, see Part 2, "Planning and Installation."

CA scalability via cloning

If you don't want to create a CA hierarchy comprising root and subordinate CAs, you can create multiple clones of a Certificate Manager and configure each clone to issue certificates that fall within a distinct range of serial numbers. Because clone CAs use the same CA signing key and certificate (as that of the master CA) to sign the certificates they issue, the *issuer name* in all the certificates in your PKI setup would be the same (as if they've been issued by a single CA).

For details on cloning a Certificate Manager, see “Cloning a Certificate Manager” on page 288.

PKCS #11 hardware support for smart cards and crypto accelerators

Certificate Management System supports smart cards and crypto accelerators provided by various third-party vendors of PKCS #11 version 2.1-compliant products. You can configure the server to use different PKCS #11 modules to generate and store key pairs (and certificates) for the Certificate Manager, Registration Manager, and Data Recovery Manager. Using hardware for key storage (especially for Certificate Manager and Data Recovery Manager key pairs) reduces the risk of key compromise, because hardware tokens don't reveal keys or provide means for them to be revealed, once the keys are generated in the hardware. Note that PKCS#11 hardware devices also provide key backup and recovery features for backup and recovery of the key material stored on the hardware token. Be sure to refer to the PKCS #11 vendor documentation on this subject.

For information on configuring Certificate Management System to use hardware tokens for generating and storing its key pairs and certificates, see “Tokens for Storing CMS Keys and Certificates” on page 464.

Support for Netscape client and iPlanet server products; client independence for non-Netscape products

Certificates issued by Certificate Management System work with existing Netscape client and iPlanet server products that support SSL. The certificates also work (out of the box) with a variety of non-Netscape, standards-compliant applications.

Highly scalable certificate data store

Certificate Management System uses a highly scalable, high-performance certificate storage facility—a preconfigured version of iPlanet Directory Server that's automatically installed with Certificate Management System—enabling you to issue and manage a large number of certificates. For more information, see Chapter 12, “Setting Up Internal Database.”

Flexible end-entity registration services framework

The registration services framework for end entities includes the most commonly expected PKI features: manual, directory-based, directory- and PIN-based, NIS-based, and portal enrollments; certificate-authenticated renewals and revocations (based on SSL client authentication); certificate life-cycle operations that include automated certificate renewal and expiration notifications. These features are available out of the box for both Certificate Manager and Registration Manager.

For information on enrollment, renewal, and revocation operations, see Chapter 15, “Setting Up End-User Authentication.” For information on automated notifications, see Chapter 16, “Setting Up Automated Notifications.”

Built-in plug-in modules for authentication, policy, job scheduling, and publishing

Certificate Management System simplifies the details involved in certificate issuance and management with its built-in, configurable, and extensible authentication, policy, job scheduling, and publishing components. Each of these components come with a set of default modules that enable you to configure Certificate Management System for your PKI requirements. For example, you can configure policy modules to determine the outcome of operations, such as certificate formulation (extensions, signing algorithm, key length, validity period, and so on), issuance, renewal, and revocation.

For information about all plug-in modules (such as authentication, job, policy, and publishing modules) that are provided for Certificate Management System, see “Plug-in Modules” on page 55.

Single administration point achieved via LDAP-compliant directory integration

Certificate Management System works seamlessly with any LDAP-compliant directory services for easy distribution of certificates and CRLs, thus lowering the cost of information management. The shared directory architecture enables you to manage users, including their security credentials and other shared data, at a single place. Certificate Management System can do the following:

- Authenticate users based on the information that exists in the LDAP directory.
- Integrate certificate-related information with the user and group information that exists in the LDAP directory.
- Automatically publish certificates (when they are issued) and CRLs (when created or on a periodic basis) to the LDAP directory, from which they can be easily distributed to clients and servers.

- Automatically delete expired and revoked certificates from the directory.
- Connect to the directory using password-based (basic) or certificate-based (in the context of LDAP over SSL) authentication using a digital certificate.

Supports many methods for verifying the revocation status of certificates

Revocation status of a certificate can be made available to PKI entities by publishing the CRL to various repositories. To aid you in this process, the Certificate Manager supports publishing of CRLs to the following repositories:

- An LDAP-compliant directory; see , “Setting Up LDAP Publishing.”
- A flat file; see , “Publishing Certificates and CRLs to a File.”
- An Online Certificate Status Protocol (OCSP)-compliant validation authority or OCSP responder; see , “Setting Up an OCSP Responder.”

Applications in your enterprise may use any of these repositories to verify the revocation status of a certificate.

Supports certificate generation for dual key pairs—separate key pairs for signing and encrypting mail messages

To support separate key pairs for signing and encrypting data, Certificate Management System supports generation of dual certificates for end entities capable of generating dual key pairs. If a client makes a certificate request for dual key pairs, the server issues two separate certificates.

For more information, see “Clients That Can Generate Dual Key Pairs” on page 760.

Works with Netscape Personal Security Manager, that can generate dual key pairs

Certificate Management System works seamlessly with Netscape Personal Security Manager, which when plugged into Netscape Communicator version 4.7x enables it to support protocols such as OCSP and CMC and generation dual key pairs. Personal Security Manager is a standards-based, client-independent application that performs PKI operations on behalf of Communicator 4.7x and other applications. For details, see “Netscape Personal Security Manager” on page 102.

Key archival and recovery for encryption private keys

If your organization uses S/MIME to encrypt mail messages, you can use the key archival feature offered by Certificate Management System to back up users' encryption private keys. This feature is useful when a key becomes unavailable—as, for instance, in the following cases:

- An employee loses an encryption private key (for example after a disk crash or by forgetting the password to the key file) and is unable to read previously encrypted data.
- An employee leaves the company, and company officials need to perform an audit that requires gaining access to the employee's encrypted data.

For more information, see Chapter 22, “Setting Up Key Archival and Recovery.”

Encrypted key storage and password-protected recovery

Certificate Management System stores users' encryption private keys in an encrypted key repository. Keys can be retrieved only by authorized key recovery agents. The key repository is encrypted using a Data Recovery Manager's storage private key, which is protected with one or more recovery agents' passwords. Only these designated recovery agents can authorize and initiate a key recovery process.

For more information, see “Where the Keys are Stored” on page 762.

Extensive audit and log records for detection of tampering

Certificate Management System maintains audit trails for all events—certificate requests and issuance, revocation requests, CRL publication, and so on. These audit records enable you to detect any unauthorized access or activity. In addition, extensive system and error logs record various events and system errors so that you can monitor and debug the system. All log records are stored in your local file system for quick and easy retrieval.

For more information, see Chapter 23, “Managing CMS Logs.”

Supports signing of log files for tamper detection

Certificate Management System allows you to sign log files digitally before archiving them or distributing them for audit purposes. This feature enables you to check whether the log files were tampered with after being signed.

For more information, see “Signing Log Files” on page 814.

Java SDK extension mechanism for customization

The software development kit (SDK) provided with Certificate Management System includes APIs and tutorials for customizing different aspects of the system. You can write the following custom modules:

- Authentication—for authenticating end entities during certificate enrollment.
- Policy—for setting the rules applied by the individual subsystems.
- Jobs—for PKI-related jobs that run with the individual systems.
- Mapper and publisher classes—for publishing certificates and CRLs to an LDAP-compliant directory, flat file, and an OCSP responder.

For information about writing custom plug-ins, see “CMS SDK” on page 65.

For information on customizing end-entity and agent interfaces (HTML forms and templates), see *CMS Customization Guide*.

Easy upgrade from previous versions of Certificate Management System

Certificate Management System provides an easy upgrade path from its previous version. If you want to install a separate, stand-alone version of iPlanet Console for any reason, you can download it from this site:

<http://www.iplanet.com/downloads/patches/>

GUI-based server installation and management

An installation wizard automates the installation and initial configuration process, helping you install Certificate Management System quickly and easily. Then after installation, you can locally or remotely administer Certificate Management System from various computers on your network (using the encryption, message integrity, and authentication services of SSL) with the help of an administration interface called the Certificate Management System window or the CMS window.

For more information, see “The CMS Window” on page 346.

System Overview

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)
- Online Certificate Status Protocol (OCSP)
- Wireless Transport Layer Security (wTLS)
- 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 iPlanet Directory Server and iPlanet 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, wireless applications, 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.

You can use Certificate Management System to set up and manage your own public-key infrastructure or to deploy a public certification authority. Certificate Management System meets the needs of an enterprise, leveraging your existing enterprise resources and services, and will grow with your business needs to meet the demand of Internet-scale deployments.

With Certificate Management System, you can do the following operations:

- Process certificate requests from various end entities, such as web browsers, servers, routers, and virtual private network (VPN) clients, and issue certificates that conform to X.509 version 3 standard. The server can also process certificate requests from wireless applications and issue certificates that conform to wTLS standard.
- Employ specific authentication methods for end-entity certificate enrollment, renewal, and revocation.
- Specify policy restrictions on certificate-related operations, such as certificate formulation, issuance, renewal, and revocation.
- Specify policy restrictions on key-related operations, such as archival and recovery of end users' encryption private keys.
- Revoke certificates, and maintain and publish a list of revoked certificates.
- Enable real-time verification of certificates by OCSP-compliant clients.
- Search for certificates issued by the server.
- Set up hierarchies of certificate authorities—multiple subordinate CAs chained up to a root CA. (Certificate Management System can also chain under popular public CAs that are already pretrust in popular client and server products.)
- Publish certificate information to an LDAP-compliant directory, such as iPlanet Directory Server, and maintain this information. Publish the list of revoked certificates (CRLs) to an LDAP-compliant directory, a flat file, and an online-validation authority.

This chapter describes the basic features and capabilities of Certificate Management System. Chapter 3, “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.

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.

CMS Subsystems or Managers

Certificate Management System comprises four servers (also referred to as *subsystems* or *CMS managers*) namely:

- Certificate Manager
- Registration Manager
- Data Recovery Manager
- Online Certificate Status Manager

To meet the widest possible range of configuration requirements, Certificate Management System permits the independent installation of these four subsystems, and each subsystem plays a distinct role in a PKI. Each subsystem consists of built-in, system-level components such as authentication framework for

various types of users, schedulable jobs for automating server functions, policy framework for evaluating certificate requests and formulating certificate contents, publishing framework for publishing certificates and CRLs to various repositories, and logging framework for monitoring server's activities. Certificate Management System supports a plug-in architecture for authentication, policy, job, publishing, and log components; for example, Java code modules can be plugged in to authenticate user identities and to enforce certificate issuance policies.

The Certificate Manager, Registration Manager, Data Recovery Manager, and Online Certificate Status 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.

The sections that follow explain each subsystem in detail. For descriptions of some basic deployment options, see Chapter 4, "Planning Your Deployment".

Certificate Manager

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 to an LDAP directory and a file, and CRLs to an LDAP directory, a file, and an OCSP responder. The Certificate Manager 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 person, identified in this document as *Certificate Manager agent*) 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 iPlanet Console*.)

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 from end entities and 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 notify agents of requests pending in the queue
- Can renew certificates
- Can revoke certificates
- Can publish certificates to an LDAP directory (LDAP 2.0 or higher) and to files
- Can publish CRLs to an LDAP directory (LDAP 2.0 or higher), a file, and the Online Certificate Status Manager.

Note that the publishing tasks can be performed by the Certificate Manager only. The Certificate Manager also has a built-in OCSP service, enabling OCSP-compliant clients to directly query the Certificate Manager about the revocation status of a certificate that it has issued. For example, if you plan to deploy a PKI comprising a master CA and many clone CAs, you can enable the OCSP service of the master CA. This way, all clients in your PKI setup can verify the revocation status of a certificate by querying the master Certificate Manager.

The Certificate Manager can issue certificates with the following characteristics:

- X.509 version 3
- Internationalized subject names
- Customized components in subject names
- Customized extensions

The Certificate Manager supports the following signing algorithms for both certificates and CRLs: RSA with MD2, RSA with MD5, RSA with SHA-1, and DSA with SHA-1.

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.

Registration Manager

A Registration Manager is an optional component in the PKI, enabling you to separate the registration process from the certificate-signing process. A Registration Manager is typically installed on a different machine from the Certificate Manager that it serves. During installation, you connect the Registration Manager to a Certificate Manager and configure the Certificate Manager to *trust* the Registration Manager. Once the trust is established, the Registration Manager can perform 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 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. The Certificate Manager processes the requests and issues the certificates. The Registration Manager then distributes the certificates to the end entities.

Note that you can run multiple Registration Managers remotely, all reporting to a single CA—a Certificate Manager—to verify user identities and process certificate signing requests. The Certificate Manager's ability to support multiple Registration Managers makes it more scalable and also adds an extra layer of security for the CA. For example, you can set a policy that requires all clients to go through a remote Registration Manager, and then have the remote Registration Manager route all client requests to the Certificate Manager located inside a firewall.

The 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
- Coordinating certificate renewal
- Coordinating storage of end users' private encryption keys 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 98.

Data Recovery Manager

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. End-entities do not have direct access to the Data Recovery Manager.

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; for example, Communicator versions 4.7x (with Personal Security Manager installed) and Netscape 6 support generation of dual key pairs. 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.

Note that the Data Recovery Manager archives encryption keys. It does not archive signing keys, since such archival would undermine nonrepudiation properties of dual-key certificates. This crucial element of a PKI allows an authorized *key-recovery agent* to recover an encryption key that has been lost or corrupted without changing the signing certificate or signing key pair. For example, if agents or administrators are authorized to perform key recover operations, they can

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.

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 1-1.

Table 1-1 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

Online Certificate Status Manager

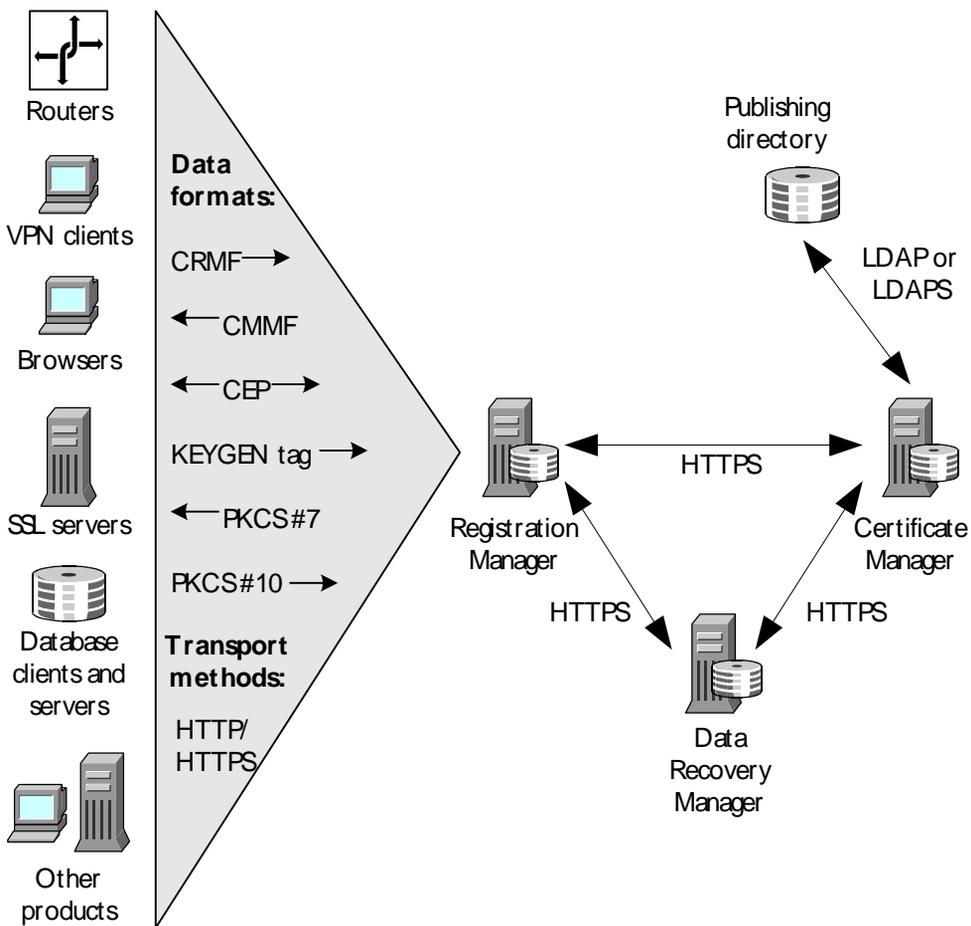
A Online Certificate Status Manager performs the task of an online certificate validation authority, by enabling OCSP-compliant clients to do real-time verification of certificates. The Online Certificate Status Manager can receive CRLs from multiple Certificate Managers and clients can query the Online Certificate Status Manager for the revocation status of certificates issued by all these Certificate Managers. For example, if you plan to create a CA hierarchy comprising a root CA and many subordinate CAs, you can configure each of these CAs to publish their CRLs to the Online Certificate Status Manager. This way, all clients in your PKI deployment can verify the revocation status of a certificate by querying the Online Certificate Status Manager.

Note that an online certificate-validation authority is often referred to as *OCSP responder*.

Basic System Configuration

Figure 1-1 illustrates some of the data formats and protocols used among the four 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 98 and “Standards Summary” on page 77.

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 four subsystems are installed in separate CMS instances (whether on the same machine or on different machines), they use proprietary connectors to communicate with each other over HTTPS—that is, HTTP over SSL, as shown in Figure 1-1. For information about the connectors, see “Trusted Managers” on page 405.

The Certificate Manager maintains complete record of issued certificates and can publish certificates and CRLs many repositories, such as a directory using LDAP or LDAP over SSL (LDAPS), a file, or the Online Certificate Status Manager. If the Certificate Manager and directory are inside the firewall and 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 to implement an automated certificate enrollment method.

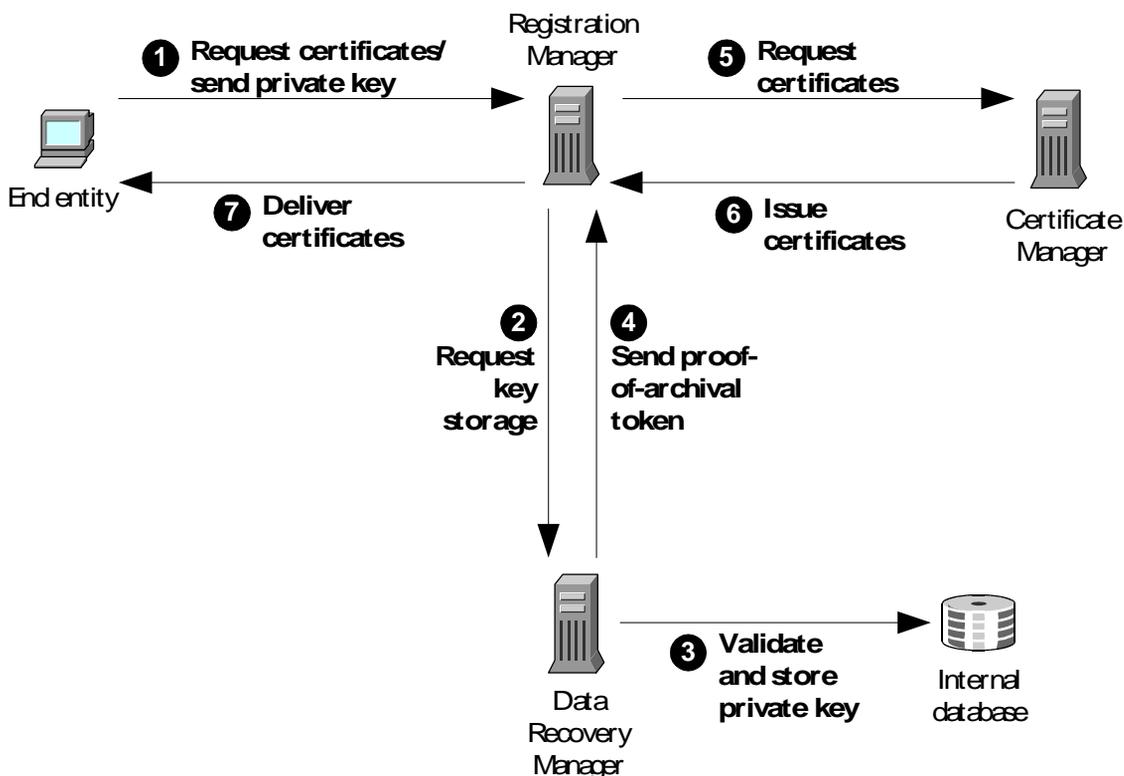
As mentioned earlier, the Data Recovery Manager performs the long-term archival and recovery of end users' private encryption keys. A Certificate Manager or Registration Manager can be configured to archive end users' private encryption keys with a Data Recovery Manager as part of the process of issuing new certificates. End-entities do not have direct access to the Data Recovery Manager.

The following steps summarize the key storage process during end-entity enrollment through a Registration Manager. Figure 1-2 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).
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 1-2 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).

Each CMS manager has its own database for storing private information such as certificate records, key archival records, and the request queue. This database is a preconfigured iPlanet Directory Server installed transparently at the time of CMS installation. In this guide, the Directory Server instance used by a subsystem for storing its data is called an *internal database*. For example, the Certificate Manager uses its internal database for storing certificates and certificate requests; the Registration Manager uses its internal database for storing certificate requests (but not certificates, which are stored by the Certificate Manager only); the Data Recovery Manager uses its internal database for storing archived encryption keys; and the Online Certificate Status Manager uses its internal database for storing CRLs published by Certificate Managers. Using Directory Server as an internal database allows Certificate Management System to leverage the scalability and industry-leading performance of Directory Server, replacing the Relational Database Management System (RDBMS) used in Certificate Server 1.0x.

Some deployments require installation of two subsystems in a single CMS instance on a single machine, for example, Certificate Manager and Data Recovery Manager, Registration Manager and Data Recovery Manager, or Data Recovery Manager and Online Certificate Status Manager. In these dual-manager installations, both subsystems use the same internal database for storing data and 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*, and *Online Certificate Status 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 iPlanet Console, and agents manage end-entity requests and certificates through HTML pages. For more information about facilities available to administrators and agents, see Chapter 13, “Managing Privileged Users and Groups.”

Plug-in 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.

The sections that follow provide an overview of the plug-in modules provided with Certificate Management System. For detailed information about all the plug-in modules, refer to *CMS Plug-Ins Guide*. To locate this guide, see “Where to Go for Related Information” on page 29.

Authentication Plug-in 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.

Both the Certificate Manager and Registration Manager support client SSL certificate-based authentication (for both agents and end entities). iPlanet Console supports user ID- and password-based authentication for administrators. Registration Managers and Certificate Managers can also be configured to use any of the authentication modules provided for authenticating end-users during certificate enrollments; see Table 1-2.

Table 1-2 Authentication plug-in modules for end-user enrollments

Plug-in module name	Description
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.
Single Sign-On authentication	Makes it possible for a Sun™ ONE Identity Server 6.0 user to authenticate himself to the Certificate Server by providing his Single Sign-On token instead of userID and password. The user can also apply for a general-purpose user certificate with a single click of a button, eliminating the need to manually import or install the certificate.

When you configure a Registration Manager or Certificate Manager authentication module, 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.

You can also write custom authentication modules, for example to authenticate end entities by using existing customer databases or security systems.

Tutorials and sample code provided as a part of CMS software development kit (SDK) demonstrate how to write a custom authentication module. For details, see section "CMS SDK" on page 65.

For information about ways customized authentication modules can be used during enrollment, see “Some Enrollment Scenarios” on page 84.

Policy Plug-in 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.

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. The processor validates the contents of a certificate request for each 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 extension 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.

For an introduction to the role of policy modules in the enrollment process, see “Authentication and Policy Modules” on page 77.

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.

Table 1-3 Policy plug-in modules for checking and formulating certificate contents

Plug-in module name	Description
AttributePresentConstraints	Rejects a request if an LDAP attribute is not present in the enrolling user’s directory entry or if the attribute does not have a specified value.
DSAKeyConstraints	Allows the server to certify only DSA keys of specified lengths.

Table 1-3 Policy plug-in modules for checking and formulating certificate contents *(Continued)*

Plug-in module name	Description
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.
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.
RSAPKeyConstraints	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.

Table 1-4 Policy plug-in modules for setting extensions in certificates

Plug-in module name	Description
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.

Table 1-4 Policy plug-in modules for setting extensions in certificates *(Continued)*

Plug-in module name	Description
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.

Table 1-4 Policy plug-in modules for setting extensions in certificates (*Continued*)

Plug-in module name	Description
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.
RemoveBasicConstraintsExt	Detects and removes the <i>Basic Constraints</i> extension in certificate requests.
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.

Table 1-4 Policy plug-in modules for setting extensions in certificates (*Continued*)

Plug-in module name	Description
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). For details, see section “CMS SDK” on page 65.

For detailed information about using certificate extensions, see Appendix C, “Certificate and CRL Extensions” of *CMS Plug-Ins Guide*. To locate this guide, see “Where to Go for Related Information” on page 29.

Job Plug-In Modules

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.

Table 1-5 Plug-in modules for schedulable jobs

Plug-in module name	Description
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.

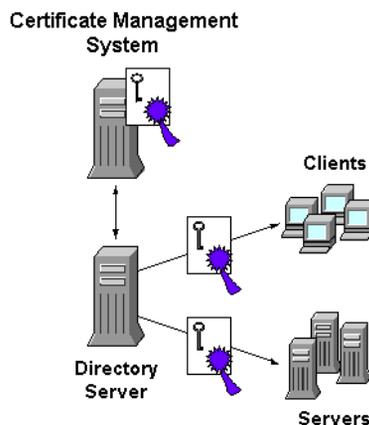
Table 1-5 Plug-in modules for schedulable jobs *(Continued)*

Plug-in module name	Description
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.

Mapper and Publisher Plug-in Modules

Mapper and publisher plug-in modules enable Certificate Management System to establish a connection with the configured repository and publish certificates and CRLs. For example, LDAP-related mapper and publisher plug-in modules enable Certificate Management System to function seamlessly with an LDAP-compliant directory, such as iPlanet Directory Server, that organizations typically use to maintain corporatewide data about user and group accounts and other network resources. You can set up Certificate Management System to automatically publish certificate information and CRLs to a directory. The advantage of publishing certificates and CRLs to the directory is multifold:

- You can keep users' certificate-related information with the rest of the user information. This way, when you update the user information, the certificate-related information automatically gets updated. For example, when you delete a user entry, the security credentials of that user automatically gets deleted from the directory.
- If you are using S/MIME-enabled clients (for example, Netscape Communicator), publishing all certificates to a central directory enables your users to import others' certificates from the global directory.

Figure 1-3 Seamless integration with any LDAP-compliant directory

Seamless integration with any LDAP-compliant directory (see Figure 1-3) makes possible the following:

- Corporate IS organizations can generate and manage certificates as an integral part of their user and group management.
- Independent CAs can issue and manage certificates to their users listed in any LDAP-compliant directory.

For more information on setting up Certificate Management System to publish certificates and CRLs, see Chapter 19 through Chapter 21.

Table 1-6 lists the mapper modules supported by Certificate Management System out of the box. Mapper modules help you configure a Certificate Manager to use specific rules to map or locate a specific entry, such as a CA's entry or an end-entity's entry, in a specified LDAP directory; once the correct entry is located, the server publishes the certificate or CRL to the correct attribute in the entry using a publisher module (explained later in this section). Because it's not required to map entries in a file and in an online validation authority, no mapper modules are provided for mapping objects in a file or a Online Certificate Status Manager.

Table 1-6 Default mapper plug-in modules for mapping certificates and CRLs

Plug-in module name	Function
LdapCaSimpleMap	Maps the CA certificate to the CA's directory entry by formulating the entry's DN from components specified in the certificate's issuer name and attribute variable assertion (AVA) constants. Optionally, the plug-in can also create an entry for the CA in the directory.

Table 1-6 Default mapper plug-in modules for mapping certificates and CRLs *(Continued)*

Plug-in module name	Function
LdapDNCompsMap	Maps a certificate to a directory entry by formulating the entry's DN from components (such as CN, OU, O, and C) in the certificate's subject name and using it as the search DN to locate the entry in the directory.
LdapDNExactMap	Maps a certificate to a directory entry by searching for the entry whose DN exactly matches the certificate subject name.
LdapSimpleMap	Maps a certificate to a directory entry by formulating the entry's DN from components specified in the certificate's subject name and attribute variable assertion (AVA) constants.
LdapSubjAttrMap	Maps a certificate to a directory entry by searching for the entry that contains the LDAP attribute named <code>certSubjNameAttr</code> whose value exactly matches the certificate subject name.

Table 1-7 lists the publisher modules supported by Certificate Management System out of the box. Publisher modules help you configure a Certificate Manager to publish certificates and CRLs to the mapped directory entries, to files, or to the Online Certificate Status Manager.

Table 1-7 Default publisher plug-in modules for publishing certificates and CRLs

Plug-in module name	Function
FileBasedPublisher	Publishes certificates and CRLs to a flat file (for exporting into other repositories).
LdapCaCertPublisher	Publishes or unpublishes a certificate to the <code>caCertificate;binary</code> attribute of the mapped directory entry as a DER encoded binary blob. Also converts the object class to a <code>certificationAuthority</code> if it's not one already; similarly, removes the <code>certificationAuthority</code> object class on unpublish if the CA has no other certificates.
LdapCrlPublisher	Publishes (replaces) a CRL to the <code>certificateRevocationList;binary</code> attribute of the mapped directory entry as a DER encoded binary blob. The entry should be a <code>certificationAuthority</code> object class.
LdapUserCertPublisher	Publishes or unpublishes a certificate to the <code>userCertificate;binary</code> attribute of the mapped directory entry as a DER encoded binary blob.
OCSPPublisher	Publishes CRLs to a Online Certificate Status Manager.

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.

For more information, see Chapter 16, “Setting Up Automated Notifications.”

Auxiliary Components

In addition to the core components that are discussed in the preceding sections, Certificate Management System also comes with command-line utilities or tools and Software Development Kit.

Command-Line Utilities

A number of command-line utilities or tools are bundled with Certificate Management System. These tools are useful for troubleshooting any problems that you may encounter with Certificate Management System. The binaries for all the utilities are located in this directory: `<server_root>/bin/cert/tools`

For detailed information about these utilities, see *CMS Command-Line Tools Guide*.

CMS SDK

CMS Software Development Kit (SDK) includes information that’s useful for developing new plug-in modules and for customizing various aspects of Certificate Management System. During installation, files for CMS SDK are copied to this directory: `<server_root>/cms_sdk/`

Below is an overview of what’s contained in the above directory:

- CMS SDK, which includes Javadocs, Samples, and Tutorials for developing Java plug-ins:

Javadocs—complete javadoc specification of the CMS Application Programming Interface (API).

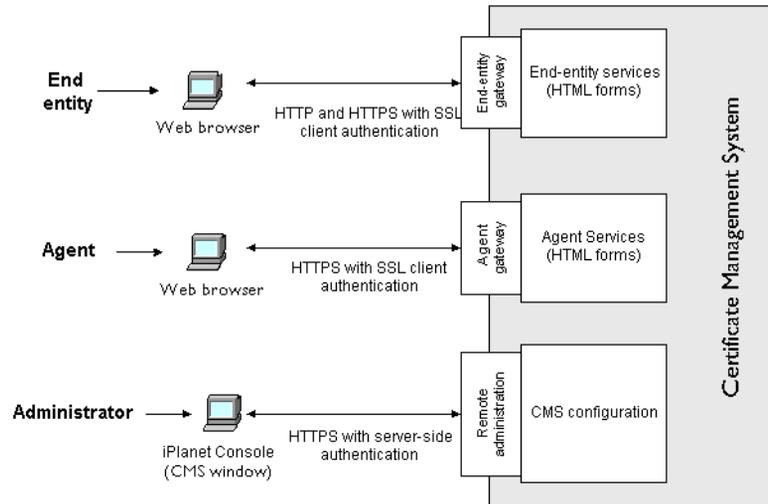
Samples—sample source code of various plug-in modules that are included in Certificate Management System out-of-the-box. This source code has been included for reference purposes only, and is only used to demonstrate how a particular CMS feature was implemented. Since a sample represents the actual code currently present in Certificate Management System, it does not require to be recompiled. You will find examples for the authentication, jobs, listeners, mappers, passwords, policies, publishers, and servlets modules.

Tutorials—“How To” tutorial to help demonstrate how you can create your own plug-in modules for Certificate Management System. Each tutorial includes sample Java source code, environment and build scripts for both UNIX and Windows NT, and a detailed “cookbook” describing how to build and install these plug-in modules. Additionally, if necessary, some tutorials may also contain sample configuration files. A tutorial has been included for authentication, job, listener, mapper, password, policy, publisher, and servlet modules.

- White papers about HTTP-related abilities of Certificate Management System including “How to add extra parameters to request from the Manual approval page” and “The CMS 4.x Bulk Generation Interface Specification”.
- Miscellaneous information about CMS features such as an AutoInstaller, an AutoRestart, script for UNIX, and a large zip file containing a sophisticated demonstration of ObjectSigning capabilities.
- Examples of how to use Certificate Management System with some third-party products.

Entry Points for Various Types of Users

Certificate Management System provides entry points for various kinds of user interaction.

Figure 1-4 Entry points for different types of CMS users

As illustrated in Figure 1-4, the server provides three separate user entry points; each entry point addresses the needs of a specific user type. This is explained in Table 1-8.

Table 1-8 Certificate Management System user entry points

User type	Component/Tool	CMS interface
End entity	Web browser	End Entity Services
		This interface provides the general front end for end-entity interactions with the server. Through this interface, the Certificate Manager or Registration Manager serves the appropriate HTML forms for end-entity operations (the Data Recovery Manager and Online Certificate Status Manager do not have an end-entity interface). These include forms for certificate enrollment, retrieval, query, renewal, import, and revocation. For details, see “End-Entity Services Interface” on page 72.

Table 1-8 Certificate Management System user entry points (*Continued*)

User type	Component/Tool	CMS interface
Agent	Web browser	<p>Agent Services</p> <p>This interface provides the general front end for agent interactions with the server. Through this interface, a Certificate Manager, Registration Manager, Data Recovery Manager, or Online Certificate Status Manager serves the appropriate HTML forms for agent tasks. For details, see “Agent Services Interface” on page 68.</p> <p>Accessing Agent Services is a privileged operation; agents must use designated certificates for SSL client authentication to Certificate Management System.</p>
Administrator	iPlanet Console (CMS window)	<p>The remote administration interface supports a GUI-based administration tool called iPlanet Console that provides the general administration and management interface for Certificate Management System. For details, see Chapter 9, “Administration Tasks and Tools.”</p> <p>Administrators can use this tool to perform day-to-day operational and managerial duties, such as changing the server configuration, stopping and restarting the server, requesting and installing certificates, managing resources (certificates and requests), and setting up privileged-user information and associated access controls.</p> <p>The CMS window can only be launched from within iPlanet Console. Accessing this window is a privileged operation requiring a password-based authentication to Certificate Management System.</p>

Agent Services Interface

As an administrator, you can designate privileged users, called *agents*, for each subsystem. Agents are responsible for the day-to-day operation of requests from end entities. For details, see “Agents” on page 397.

To enable agents to accomplish their duties, Certificate Management System provides a set of HTML forms for Certificate Manager, Registration Manager, Data Recovery Manager, and Online Certificate Status Manager agents. Collectively, these forms are called the *Agent Services* interface.

Depending on the choices you made during installation, a combination of the following agent services will be installed:

- Certificate Manager Agent Services
- Registration Manager Agent Services
- Data Recovery Manager Agent Services
- Online Certificate Status Manager Agent Services Interface

The sections that follow give an overview of these interfaces. For a complete list of agent forms and output templates that come with Certificate Management System, see *CMS Customization Guide*.

For tasks associated with Agent Services interface, see *CMS Agent's Guide*. For information on locating this guide, see “Where to Go for Related Information” on page 29.

Certificate Manager Agent Services

The Certificate Manager Agent Services interface enables a Certificate Manager agent to interact with the Certificate Manager (the server). Figure 1-5 shows the Certificate Manager Agent Services interface.

Figure 1-5 Certificate Manager Agent Services interface

The screenshot displays the Certificate Manager Agent Services interface. At the top, the header includes the iPlanet logo and the text 'Certificate Management System Agent Services'. Below this is a sub-header 'Certificate Manager'. On the left side, there is a vertical navigation menu with the following options: 'List Requests' (highlighted), 'List Certificates', 'Search for Certificates', 'Revoke Certificates', 'Display Revocation List', 'Update Revocation List', 'Update Directory Server', and 'OCSP Service'. The main content area is titled 'List Requests' and contains the following text: 'Use this form to show a list of certificate requests.' Below this text are three input fields: 'Request type:' with a dropdown menu set to 'Show enrollment requests', 'Request status:' with a dropdown menu set to 'Show pending requests', and 'Starting request identifier: (optional)' with a text input field. At the bottom of the form, there is a 'Find' button, followed by the text 'first 5 records', and a 'Help' button.

Using the default forms, a Certificate Manager agent can accomplish tasks such as these:

- Listing *deferred* certificate requests from end entities and process them
- Listing certificates issued by the server
- Searching for certificates issued by the server
- Revoking certificates issued by the server
- Updating certificates and certificate revocation lists (CRLs) maintained in the publishing directory

Registration Manager Agent Services

The Registration Manager Agent Services interface enables a Registration Manager agent to interact with the Registration Manager (the server). Figure 1-6 shows the Registration Manager Agent Services interface.

Figure 1-6 Registration Manager Agent Services interface

The screenshot displays the 'Registration Manager Agent Services' interface. The top navigation bar includes 'iPlanet Certificate Management System' and 'Agent Services'. A sub-tab 'Registration Manager' is active. On the left, a vertical menu lists various actions: 'List Requests' (highlighted), 'List Certificates', 'Search for Certificates', 'Revoke Certificates', 'Display Revocation List', 'Update Revocation List', 'Update Directory Server', and 'OCSP Service'. The main content area is titled 'List Requests' and contains the following form elements:

- A heading: 'List Requests' followed by the instruction: 'Use this form to show a list of certificate requests.'
- A 'Request type:' dropdown menu with 'Show enrollment requests' selected.
- A 'Request status:' dropdown menu with 'Show pending requests' selected.
- A 'Starting request identifier:' text input field with '(optional)' below it.
- A 'Find' button, a 'first' label, a text input field containing '5', a 'records' label, and a 'Help' button.

Using the default forms, a Registration Manager agent can list *deferred* certificate requests from end entities and process them.

Data Recovery Manager Agent Services

The Data Recovery Manager Agent Services interface enables a Data Recovery Manager agent to interact with the Data Recovery Manager (the server). Figure 1-7 shows the Data Recovery Manager Agent Services interface.

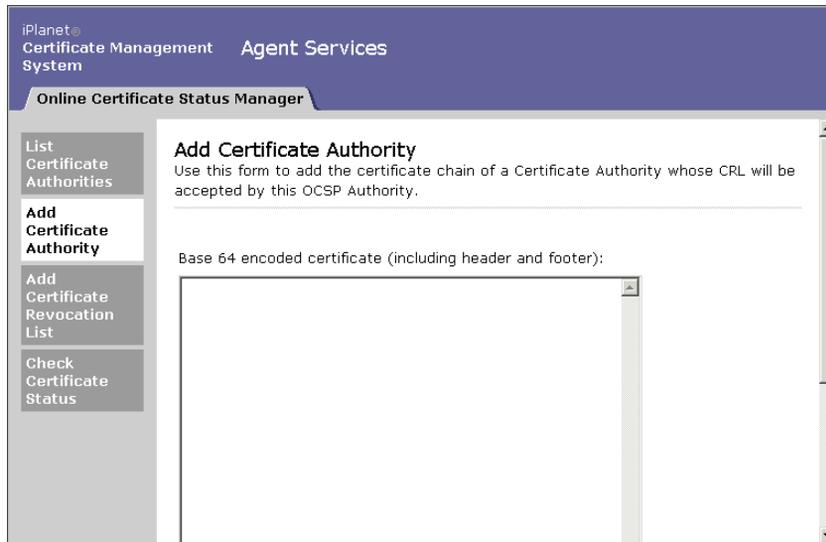
Figure 1-7 Data Recovery Manager Agent Services interface

The screenshot shows a web interface for 'iPlanet Certificate Management System Agent Services'. The main heading is 'Data Recovery Manager'. On the left, there is a vertical menu with the following items: 'List Requests', 'Search for Keys', 'Recover Keys', and 'Authorize Recovery'. The 'Authorize Recovery' option is selected. The main content area is titled 'Authorize Recovery (for Recovery Agents)' and includes the text 'Use this form to approve a key recovery.' Below this text is a text input field with the label 'Recovery authorization reference number:'. At the bottom right of the form area, there are two buttons: 'Examine' and 'Help'.

Using the default forms, a Data Recovery Manager agent can search for and recover end users' encryption private keys from the key archive. (Key recovery requires authorization from key recovery agents; see “Key Recovery Process” on page 765.)

Online Certificate Status Manager Agent Services Interface

The Online Certificate Status Manager Agent Services interface enables a Online Certificate Status Manager agent to interact with the Online Certificate Status Manager (the server). Figure 1-8 shows the Online Certificate Status Manager Agent Services interface.

Figure 1-8 Online Certificate Status Manager Agent Services interface

Using the default forms, a Online Certificate Status Manager agent can perform tasks such as checking which CAs are currently configured to publish their CRLs to the Online Certificate Status Manager, identifying a Certificate Manager to the Online Certificate Status Manager, adding CRLs directly to the Online Certificate Status Manager, and viewing the status of OCSP service requests submitted by OCSP-compliant clients.

End-Entity Services Interface

Certificate Management System provides HTML forms for various entities—people, routers, servers, and others—that use certificates to identify themselves and that need to be able to request certificate issuance and management operations. These forms, collectively identified as *End-Entity Services Interface*, use different protocols and life-cycle management procedures for different kinds of end entities. For example, the Certificate Manager provides separate certificate enrollment forms for clients such as Netscape Navigator 3.x, versions of Netscape Communicator later than 4.5, and Microsoft Internet Explorer. The reason for this is that end entities running Navigator 3.x and Communicator versions earlier than 4.5 present an enrollment form based on the use of the HTML tag `KEYGEN` to generate keys; end entities running Internet Explorer present a form based on PKCS #10, the RSA standard for certificate request syntax.

For a summary of the various end entities, protocols, cryptographic algorithms, and key pairs (single or dual) supported by Certificate Management System, see “End Entities and Life-Cycle Management” on page 98.

Figure 1-9 shows the end-entity services interface of a Certificate Manager.

Figure 1-9 End-entity services interface

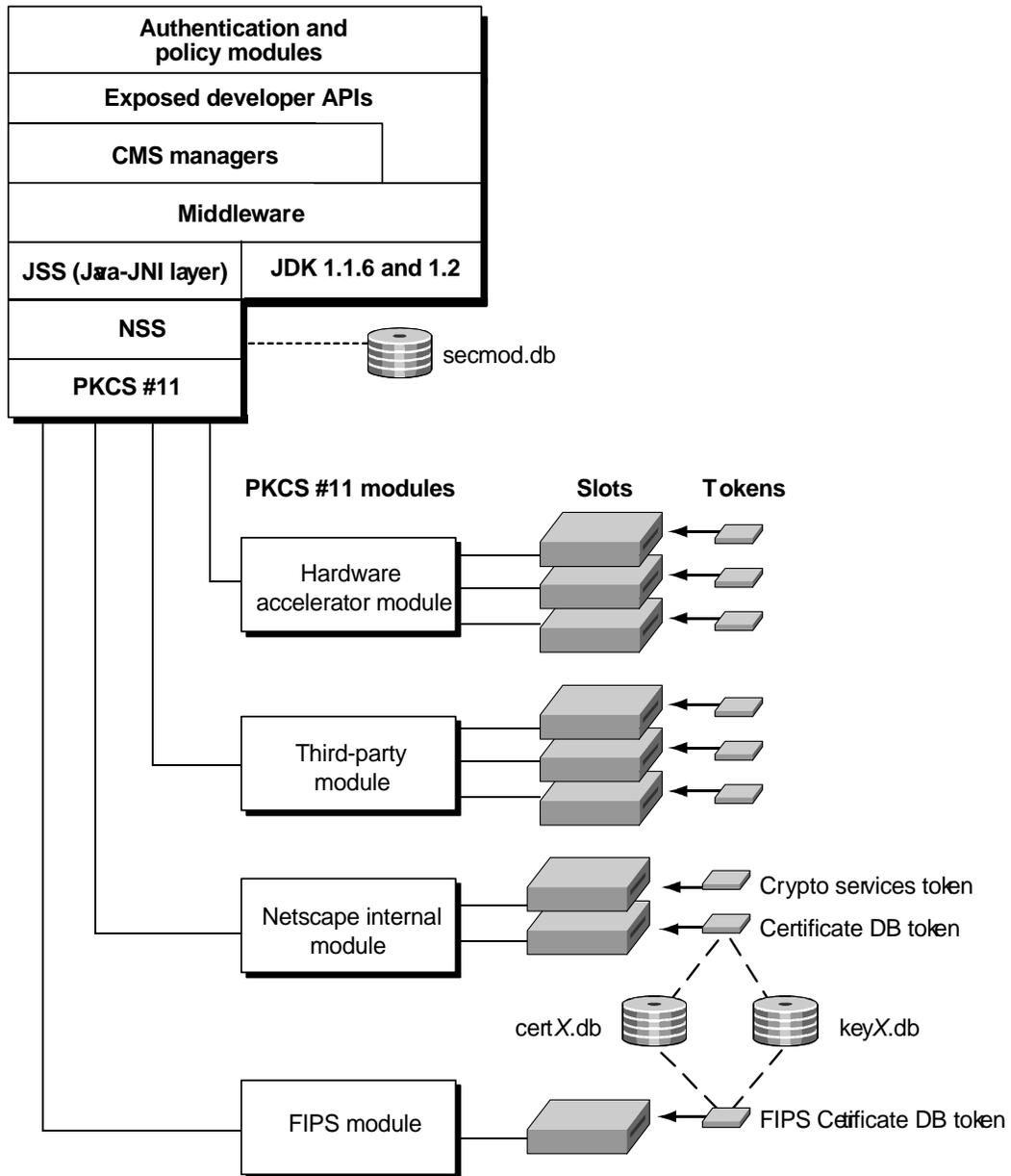
The screenshot displays the 'Certificate Manager' web interface. At the top, it says 'iPlanet® Certificate Management System' and 'Certificate Manager'. Below the header are four tabs: 'Enrollment', 'Renewal', 'Revocation', and 'Retrieval'. The 'Enrollment' tab is selected. On the left is a vertical navigation menu with categories: 'Browser' (containing 'Manual'), 'Server', 'SSL Server', 'Registration Manager', 'Certificate Manager', 'OCSP Responder', 'Other', 'Object Signing (Browser)', and 'Object Signing'. The 'Manual' option under 'Browser' is highlighted. The main content area is titled 'Manual User Enrollment' and contains the following text: 'Use this form to submit a request for a personal certificate. After you click the Submit button, your request will be submitted to an issuing agent for approval. When an issuing agent has approved your request you will receive the certificate in email, along with instructions for installing it.' Below this is an 'Important:' note: 'Be sure to request your certificate on the same computer on which you plan to use the certificate.' The 'User's Identity' section instructs the user to 'Enter values for the fields you want to have in your certificate. Your site may require you to fill in certain fields. (* = required field)'. It includes four input fields: '* Full name:', 'Login name:', 'Email address:', and 'Organization unit:'.

Note that the Data Recovery Manager and Online Certificate Status Manager do not provide end-entity interfaces because end entities do not directly interact with these servers. For a complete list of the end-entity forms—for enrollment, renewal, retrieval, revocation, and key recovery—that come with Certificate Management System, see *CMS Customization Guide*.

System Architecture

Figure 1-10 shows the internal architecture of Certificate Management System. The sections that follow describe the basic elements of this architecture, starting at the bottom of the figure.

Figure 1-10 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 1-10, 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; see “Installing Level 2 External Tokens” on page 466.

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 1-10) 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 1-10), 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 explained in the *CMS Command-Line Tools Guide*. For example, you need to modify `secmod.db` if you are installing hardware accelerators for use in signing operations.

NSS

Network 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.

For more information about NSS, check this site:

<http://www.mozilla.org/projects/security/pki/nss/>

As shown in Figure 1-10, 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/Java 2 Layers

A middleware layer above JSS and the Java/JNI layer provides a range of services required by the Registration Manager, Certificate Manager, Data Recovery Manager, and Online Certificate Status Manager. The middleware layer is based on Java 2.0, SDK 1.3.0, 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 1-10 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.

Certificate Enrollment and Life-Cycle Management

This chapter explains how you can use iPlanet Certificate Management Server (CMS) for issuing certificates to end entities such as we browsers, servers, routers, and so on.

The chapter has the following sections:

- Steps in End-Entity Enrollment (page 81)
- Some Enrollment Scenarios (page 84)
- End Entities and Life-Cycle Management (page 98)

This chapter assumes that you've read the previous chapter, Chapter 1, "Introduction to Certificate Management System."

Steps in End-Entity Enrollment

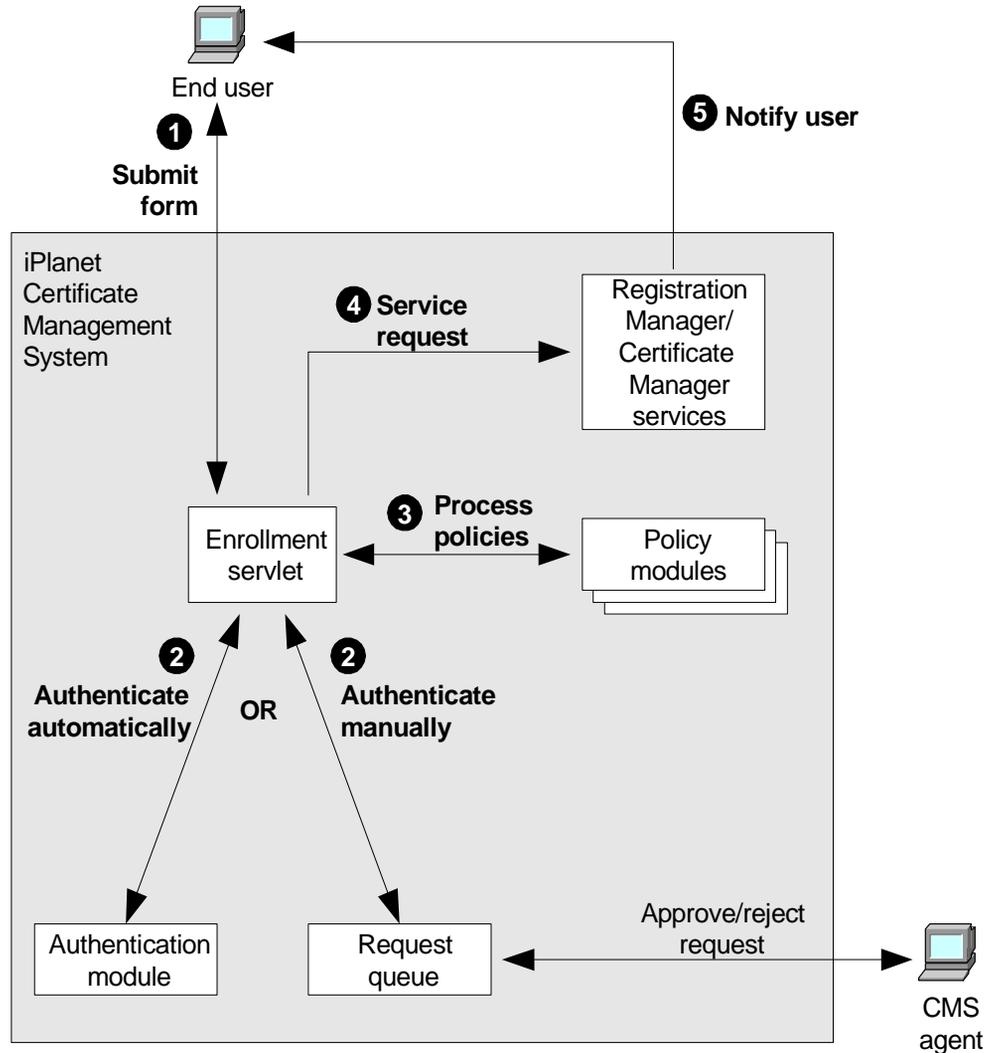
The following steps take place when a Registration Manager or a Certificate Manager handles an enrollment request from an end user. Figure 2-1 shows a simplified view of how this works.

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 73.

Figure 2-1 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 2-1 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
- Extranet/E-Commerce: Acme Sales Corp.
- PIN Registration: Atlas Manufacturing
- VPN Client Enrollment and Revocation
- Router Enrollment and Revocation

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 48.

For more information about certificate life-cycle management, see “End Entities and Life-Cycle Management” on page 98.

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, iPlanet 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, iPlanet 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
- Enrolling New Customers
- Enrolling Extranet Users

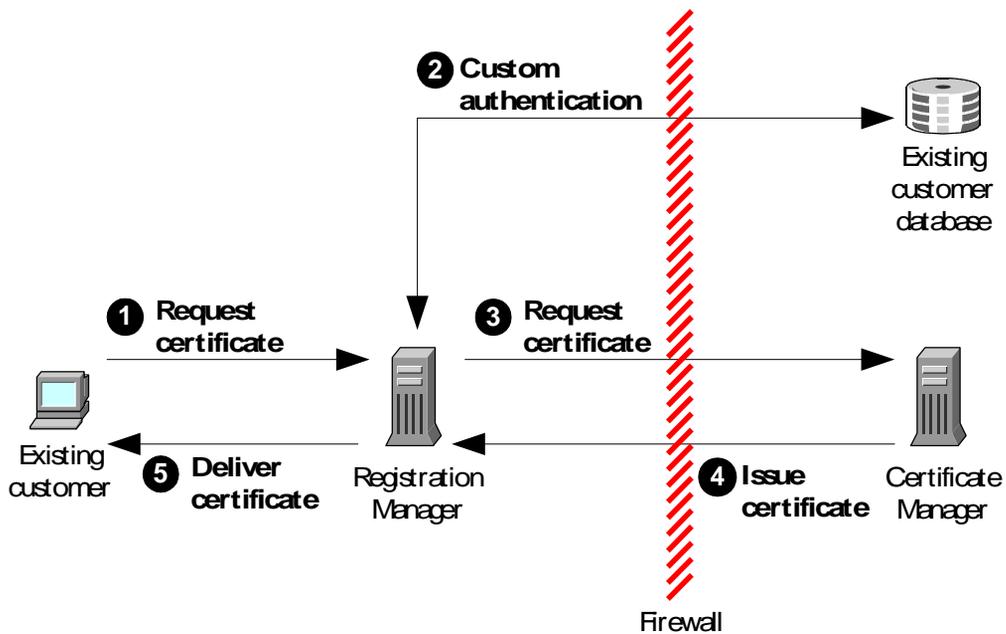
In all cases, Acme has decided to place its Certificate Manager behind the firewall and its Registration Manager outside the firewall, for reasons summarized in “Firewall Considerations” on page 84.

Enrolling Existing Customers

Acme has decided on the following process for registering its existing customers, as shown in Figure 2-2.

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 2-2 Custom authentication against an existing customer database



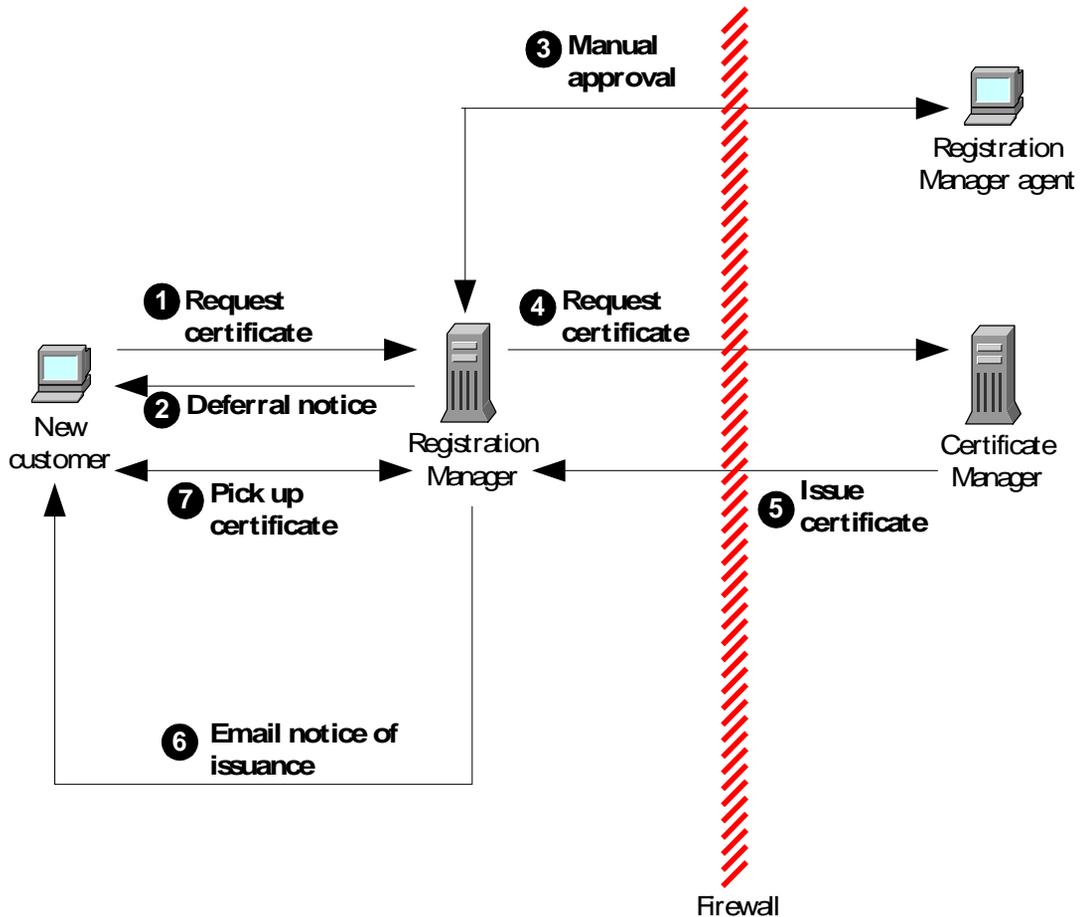
Enrolling New Customers

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 2-3 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 2-3 Manual authentication of new customers



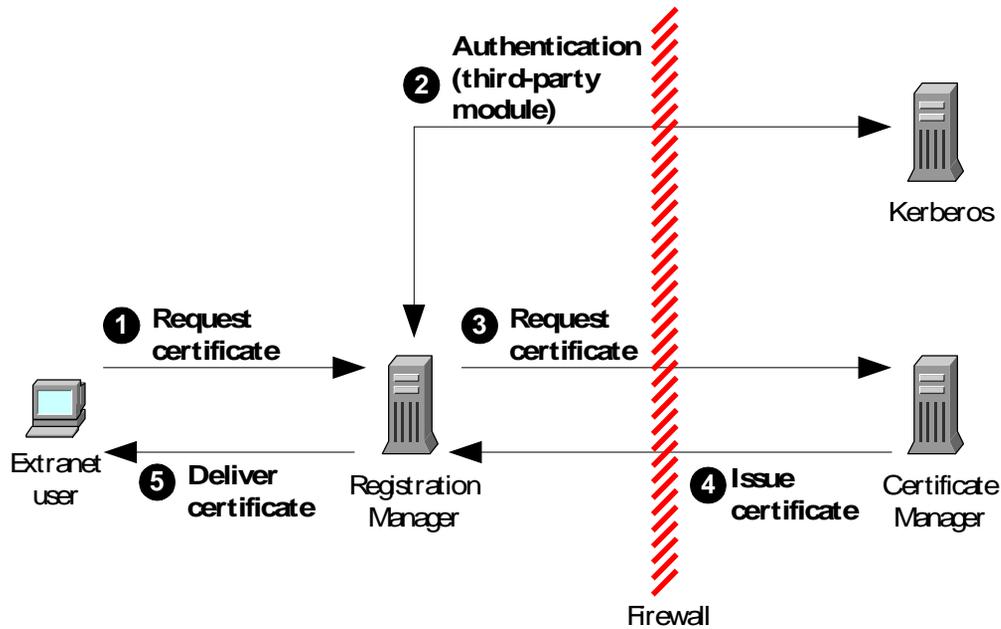
Enrolling Extranet Users

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 DASCUM 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 2-4. (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 2-4 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 iPlanet 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.

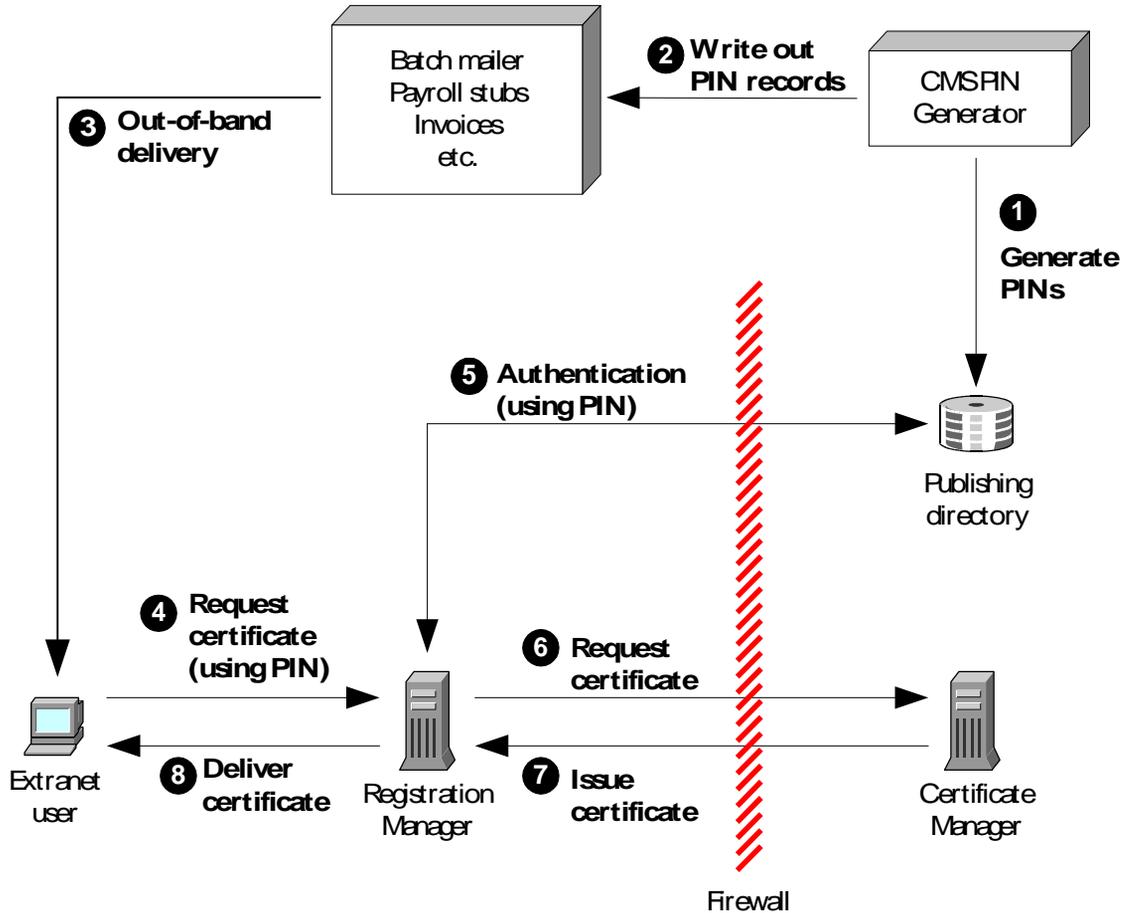
PINs are salted and hashed before storage in the directory. *Salting* refers to the inclusion of additional information from the distinguished name (DN) with the PIN to ensure unique hashing. *Hashing*, in this case, involves generating a number of fixed length from the PIN and DN information. Even if 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 2-5):

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 2-5 PIN-based enrollment



VPN Client Enrollment and Revocation

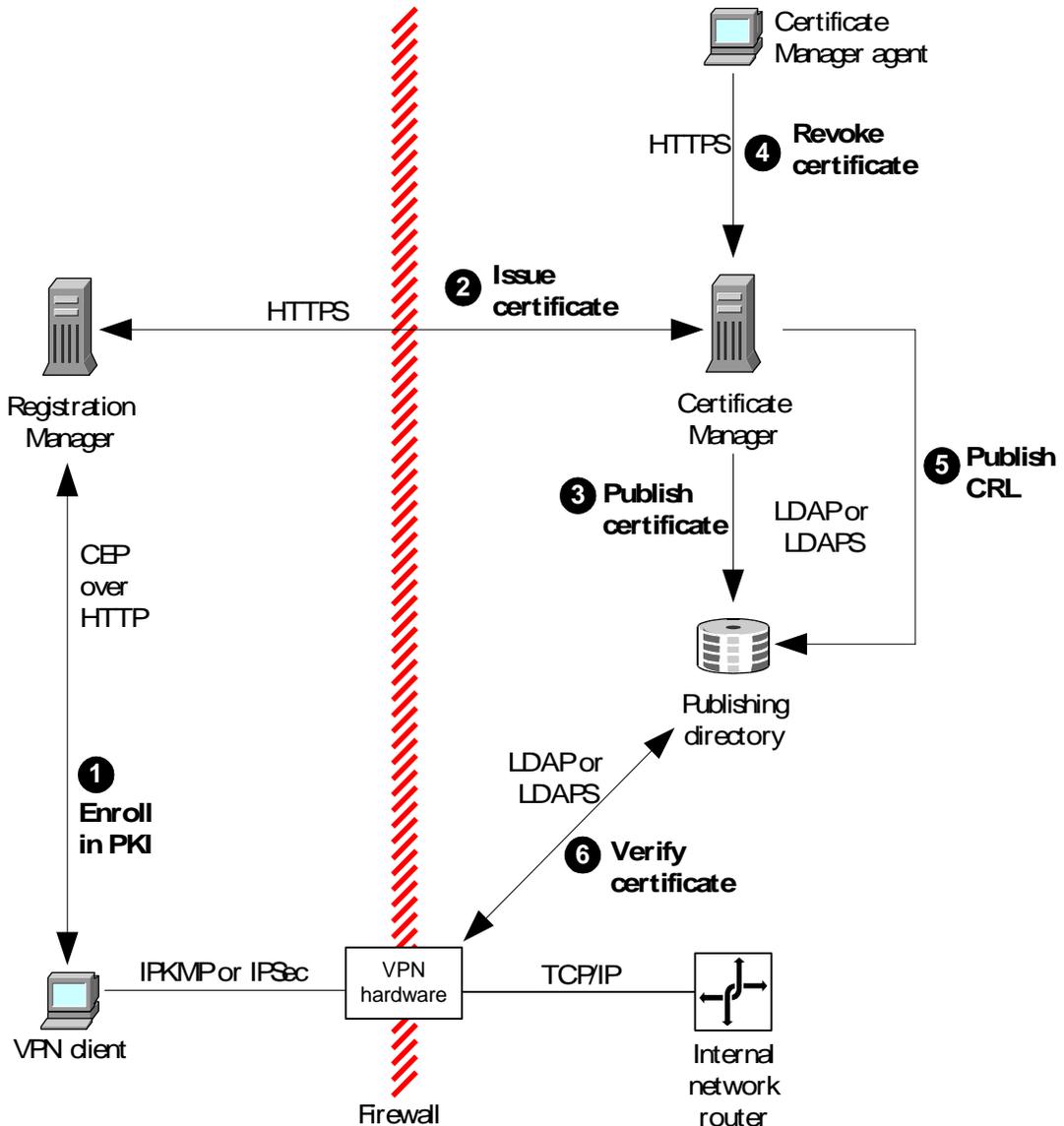
Virtual private network (VPN) client software runs on a user's desktop, outside the firewall, and uses the IP Key Management Protocol (IPKMP) or IP Security (IPSec) protocol to establish encrypted communication with VPN hardware that straddles the firewall. These protocols allow VPN hardware to authenticate 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 2-6.

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.
6. **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 2-6 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.

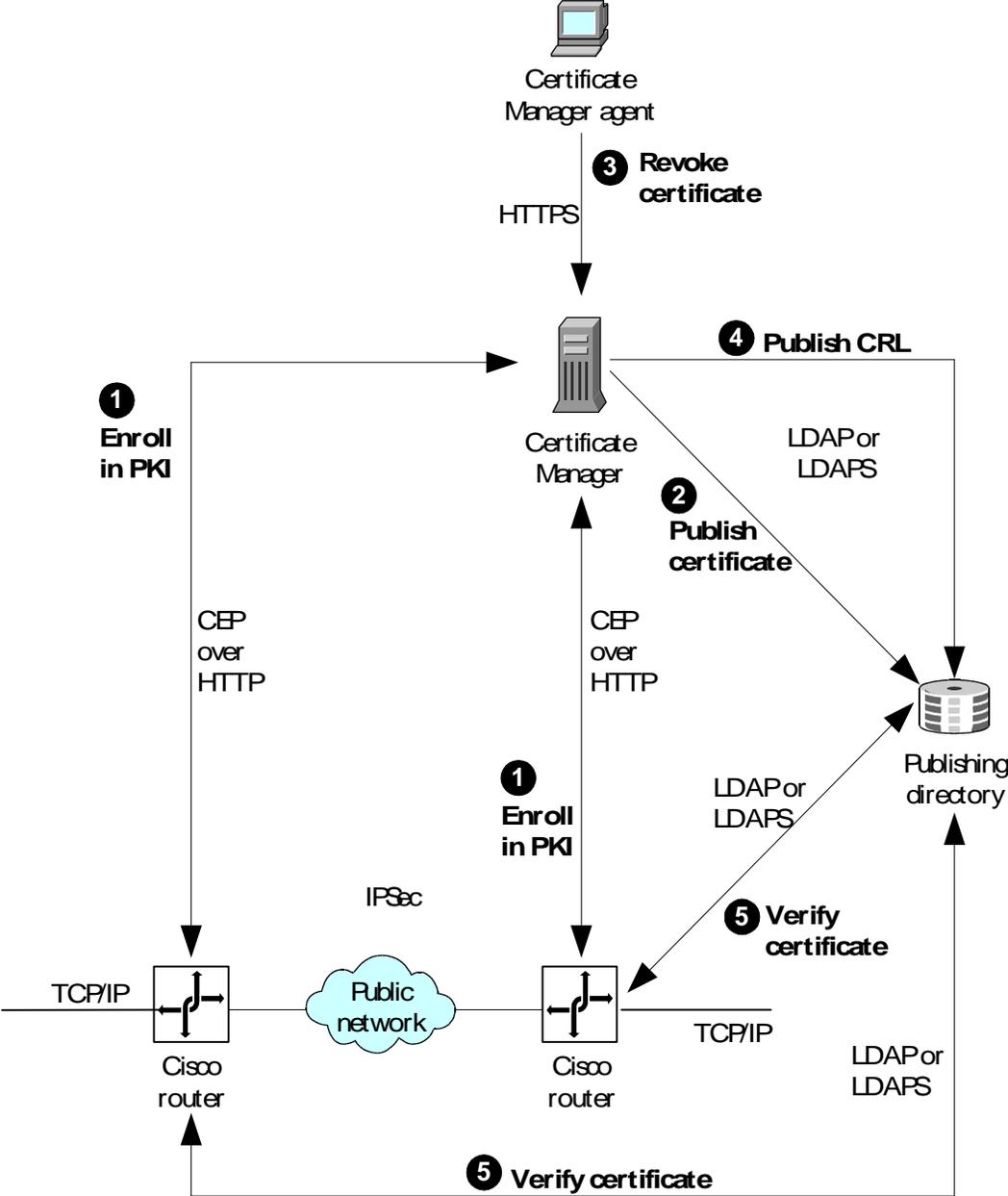
Router Enrollment and Revocation

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 2-7.

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 2-7 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 4.7x versions.

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

- Life-Cycle Management Formats and Protocols
- Access to Subsystems
- HTML Forms for End Users
- Netscape Personal Security Manager

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 Manager or Registration Manager in a variety of nonstandard and standards-based formats.

Table 2-1 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 2-1, see “Standards Summary” on page 77.

Table 2-1 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
Communicator 4.7x and later versions	CRMF and CMMF based on new JavaScript API	Signing and encryption: RSA Signing only: RSA, DSA	Single or dual key pairs
iPlanet 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

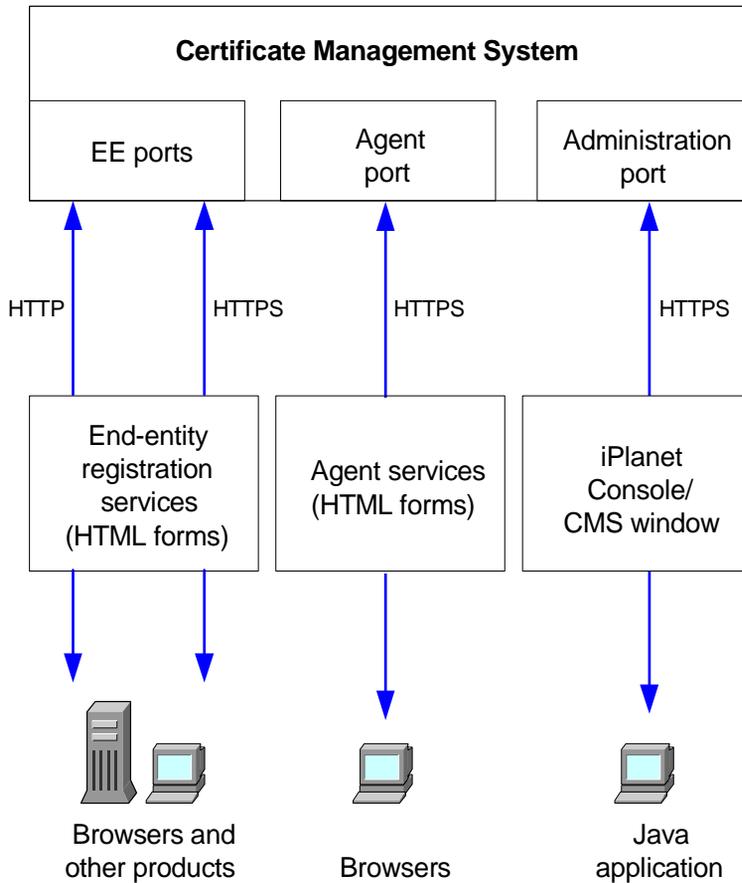
Access to Subsystems

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 2-8 shows the ports used by administrators, agents, and end entities. All agent and administrator interactions with CMS subsystems occur over HTTPS.

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 3, “Default Demo Installation.”

Figure 2-8 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 2-9 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 2-9 Default manual enrollment form for end users

The screenshot displays the iPlanet Certificate Management System interface. The top navigation bar includes the iPlanet logo, "Certificate Management System", and "Certificate Manager". Below the navigation bar are tabs for "Enrollment", "Renewal", "Revocation", and "Retrieval". A left-hand menu contains categories: "Browser", "Manual" (highlighted), "Server", "SSL Server", "Registration Manager", "Certificate Manager", "OCSP Responder", "Other", "Object Signing (Browser)", "Object Signing (PKCS10)", "CMC", and "Enrollment". The main content area is titled "Manual User Enrollment" and contains the following text: "Use this form to submit a request for a personal certificate. After you click the Submit button, your request will be submitted to an issuing agent for approval. When an issuing agent has approved your request you will receive the certificate in email, along with instructions for installing it." Below this is an "Important:" note: "Be sure to request your certificate on the same computer on which you plan to use the certificate." The "User's Identity" section instructs users to "Enter values for the fields you want to have in your certificate. Your site may require you to fill in certain fields. (* = required field)". The form includes input fields for: "* Full name:", "Login name:", "Email address:", "Organization unit:", "Organization:", and "Country:" (with "US" selected in a dropdown menu).

Table 2-2 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 2-2 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 2-1, see “Standards Summary” on page 77.

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.
- 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, Personal Security Manager 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.

Keep in mind that Personal Security Manager works only with Netscape Communicator, version 4.7x, which can be downloaded from this site:

<http://home.netscape.com/download/>

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 iPlanet Certificate Management Server (CMS) procedures.

CAUTION 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 4, “Planning Your Deployment” and the chapters that follow.

This chapter has the following sections:

- System Requirements (page 106)
- Overview of the Default Demo (page 108)
- Installing the Default Demo (page 112)
- Using the Default Demo (page 139)

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
- Platform Requirements

NOTE Be sure to check the Release Notes that came with the product. It would contain any last-minute changes to the information specified in this section.

Operating System and Software Required

Operating systems supported:

- Sun Solaris 8 (with relevant Java 2 patches)
- Windows 2000 and Windows NT 4.0 with Service Pack 6

Other required software:

- iPlanet Administration Server 5.1 (included)
- iPlanet Directory Server 5.1 (included)
- Browser software that supports SSL

Platform Requirements

Each platform has slightly different requirements. In addition to the requirements listed in Table 3-1, make sure you have ample swap space or virtual memory allocated for the system on which you intend to install Certificate Management System.

Table 3-1 Software and hardware requirements

Solaris Platform Requirements

Table 3-1 Software and hardware requirements (*Continued*)

OS Version	Solaris 8 (with relevant Java 2 patches)
Machine	Ultra 1 or faster
RAM	128 MB (256 recommended)
Hard disk storage space requirements	Total required is approximately 450 MB, broken down as follows: <ul style="list-style-type: none"> • Total transient space required during installation: 100 MB • Hard disk storage space required for installation (approximate values): Space required for setup, configuration, and running the server: 300 MB Additional space to allow for database growth in pilot deployment: 50 MB (this may be reduced to 10 MB for default demo installation) Total disk storage space for installation: 350 MB

Windows NT Platform Requirements

OS Version	Windows 2000, Windows NT 4.0 with Service Pack 6
Machine	Pentium II 400 or faster
File system	NTFS or FAT
RAM	128 MB of RAM (256 recommended)
Hard disk storage space requirements	Total required is approximately 350 MB, broken down as follows: <ul style="list-style-type: none"> • Total transient space required during installation: 100 MB • Hard disk storage space required for installation (approximate values): Space required for setup, configuration, and running the server: 200 MB Additional space to allow for database growth in pilot deployment: 50 MB (this may be reduced to 10 MB for default demo installation) Total disk storage space for installation: 250 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.
 - On a Windows NT system, you must install as Administrator or a user with Administrator privileges (that is, the user must be in the Administrators group).
-

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 4, “Planning Your Deployment” and the chapters that follow.

The default demo installation includes the following iPlanet software:

- **iPlanet Console.** iPlanet Console is described in a separate guide, *Managing Servers with iPlanet Console*. It is a standalone Java application used to manage iPlanet server instances with the aid of a configuration directory and a user directory. For this demo, iPlanet Console controls just the server instances listed here; the configuration and user directories are combined in a single iPlanet Directory Server instance. In real deployments, iPlanet 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.
- **iPlanet Administration Server.** This lightweight HTTP server acts as the back end to iPlanet Console. An instance of Administration Server manages operation requests involving any iPlanet 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 (iPlanet Directory Server).** This is an instance of Directory Server with two subtrees. The user subtree keeps track of users and groups and their privileges (for the Administration Server, not for Certificate Management System). The configuration subtree keeps track of the location on the network of iPlanet 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 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 Certificate Management System contains a Certificate Manager that is configured to perform registration tasks as well as CA tasks.

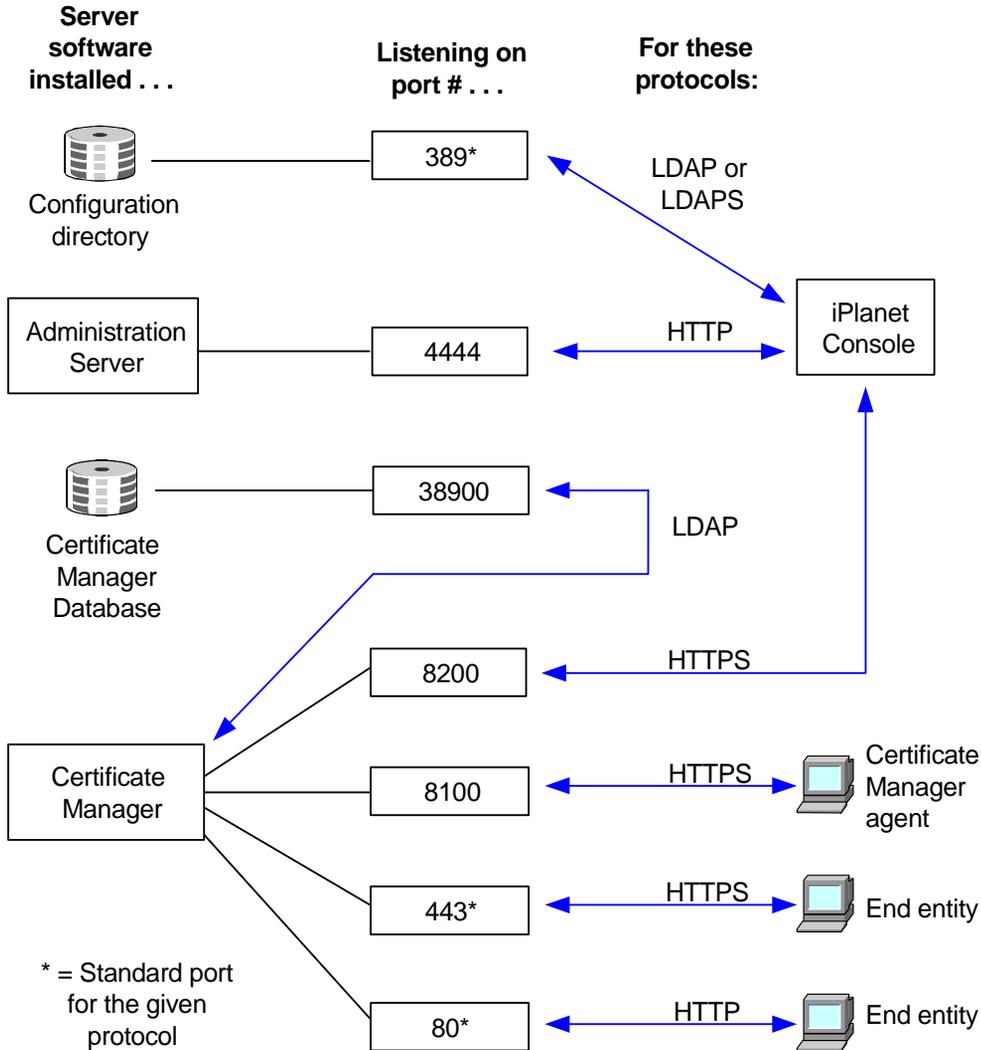
- **Internal Database (iPlanet Directory Server) for Certificate Management System.** For each instance of Certificate Management System you install an instance of iPlanet Directory Server that acts as the internal database for certificate and request information.

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

iPlanet 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 iPlanet Console window for a particular Certificate Management System instance, iPlanet Console acts directly on a configuration file stored with that instance. (For more information about the configuration file, see Chapter 10, “CMS Configuration.”)

As you proceed with the default demo installation and configuration, you will be asked to assign several port numbers, names, and passwords. Figure 3-1 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.

Figure 3-1 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 4, “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 iPlanet Console and the Installation Wizard.
<code><dir mgr password></code>	Manager for the configuration directory. (This password must be at least eight characters.)
<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 iPlanet 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> .

<single-signon
password>

This password protects the <intdb password> and
<token password>. 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 or
Step 1. Run the Installation Script—Windows NT
- Step 2. Run the Installation Wizard
- Step 3. Get the First User Certificate

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 114.

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/iplanet/servers]:** 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 iPlanet 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. **iPlanet 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. **Certificate Management System Server identifier [localhost]:** Type `demoCA` 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 iPlanet 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

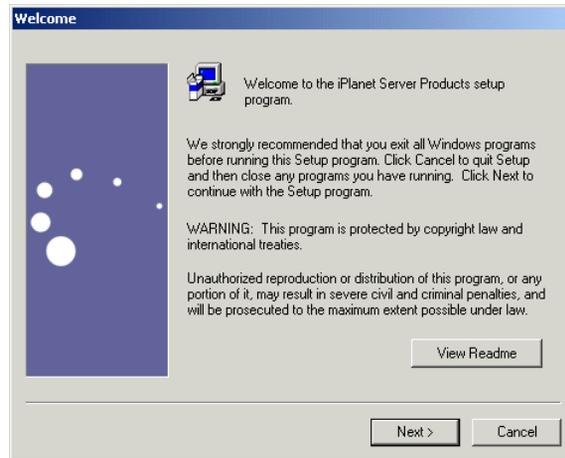
These instructions assume that you have the initial distribution of Certificate Management System available, either on a CD or on your hard disk.

If you are using a UNIX system, see “Step 1. Run the Installation Script — UNIX” on page 112.

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.

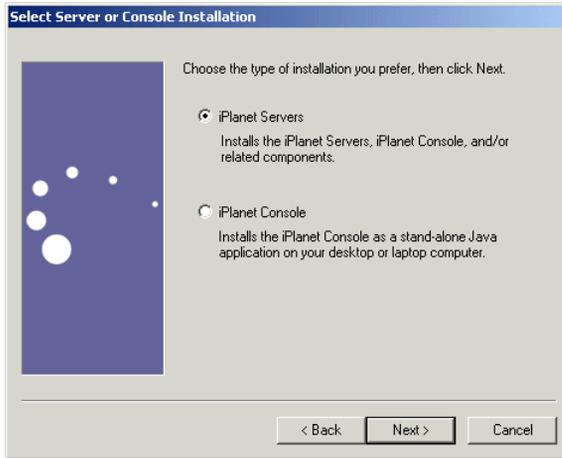
2. Welcome. Click Next.



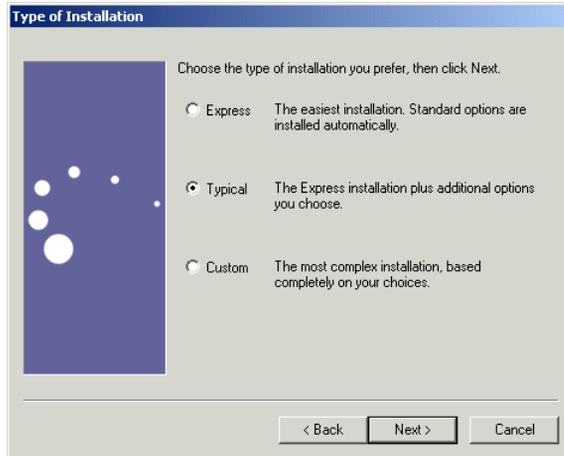
3. Software License Agreement. Click Yes.



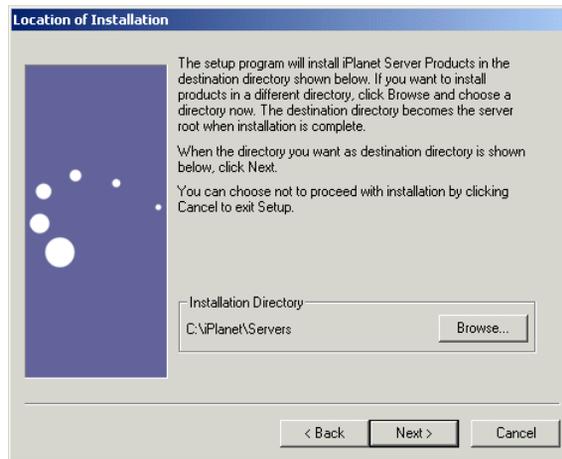
- Select Server or Console Installation.** Leave the default setting (iPlanet Servers) selected and click Next.



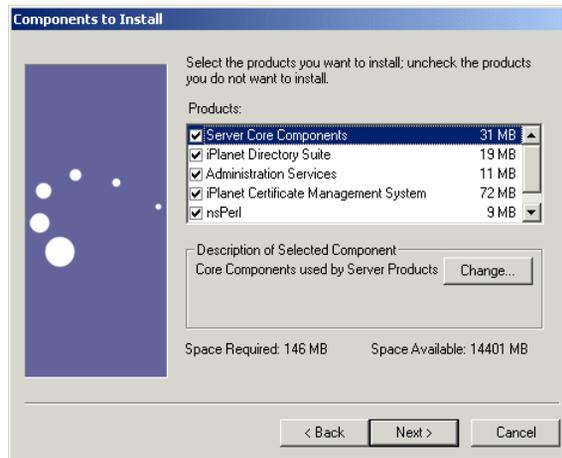
- Choose the Installation Type.** Leave the default setting (Typical) selected and click Next.



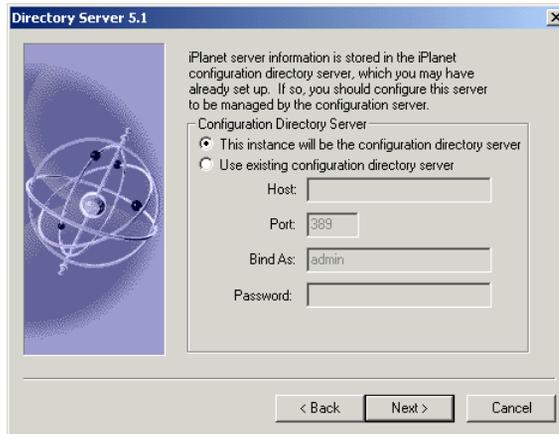
6. **Choose Installation Directory.** Leave the default setting (C:\iPlanet\Servers) selected and click Next.



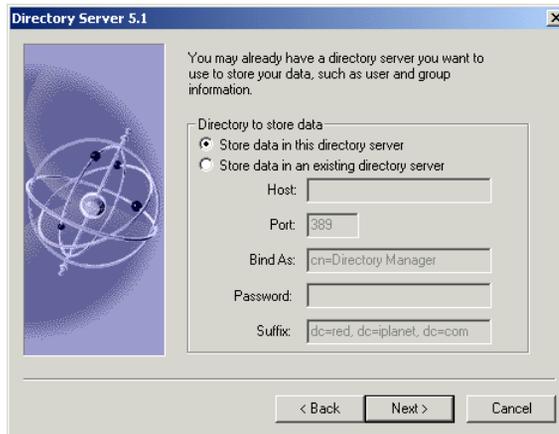
7. **Select Products.** Leave all four components selected and click Next.



- 8. Directory Server 4.13.** Leave the default setting (This instance will be the configuration directory server) selected and click Next.



- 9. Directory Server 4.13.** Leave the default setting (Store data in this directory server) selected and click Next.

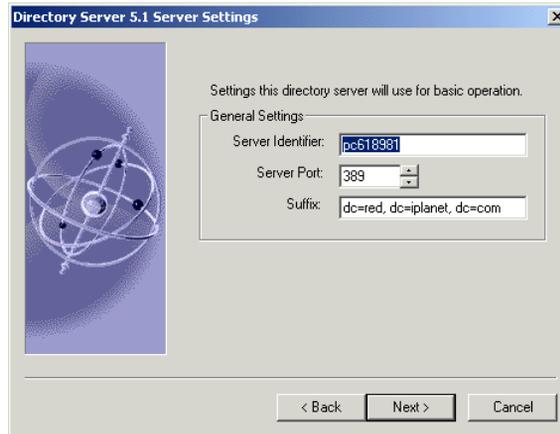


- 10. Directory Server 4.13 Server Settings.** Type the following values, then click Next:

Server identifier: Accept the default.

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=<your_domain>.<domain>`.

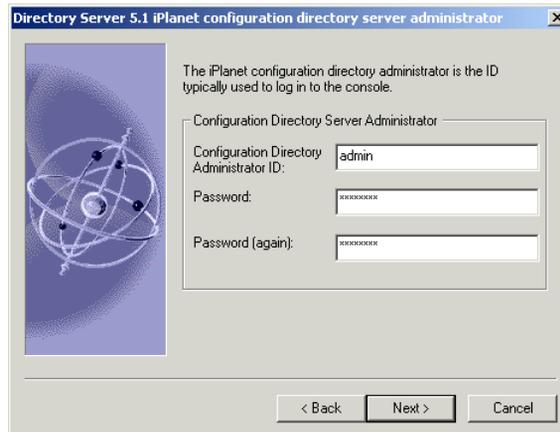


11. **Directory Server 5.1 iPlanet Configuration Directory Server Administrator.**
Type the following values, then click Next:

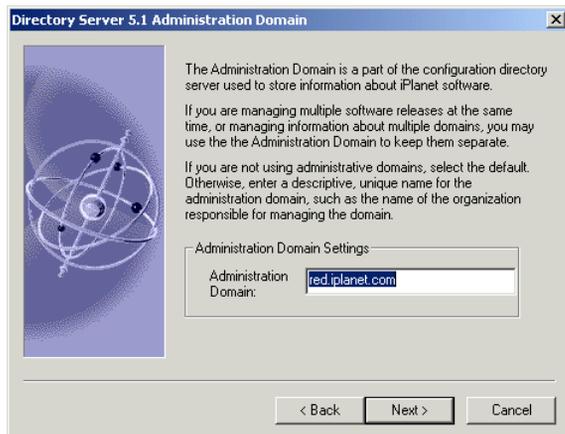
Configuration Directory Administrator ID: admin

Password: <admin password>

Password (again): <admin password>



12. **Directory Server 4.13 Administration Domain.** Accept the default, which should be your company's domain name, in the form <your_domain>.<domain>.

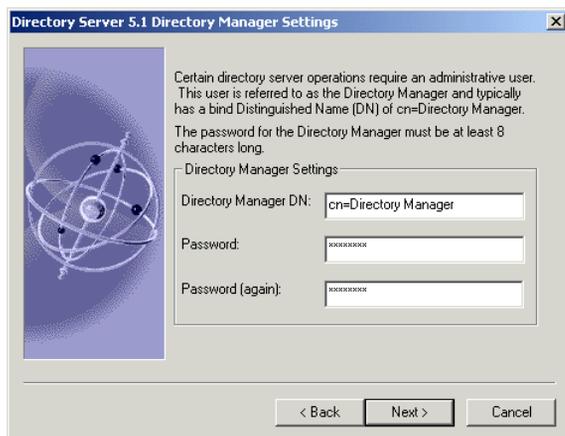


13. **Directory Server 4.13 Directory Manager Settings.** Type the following values, then click Next:

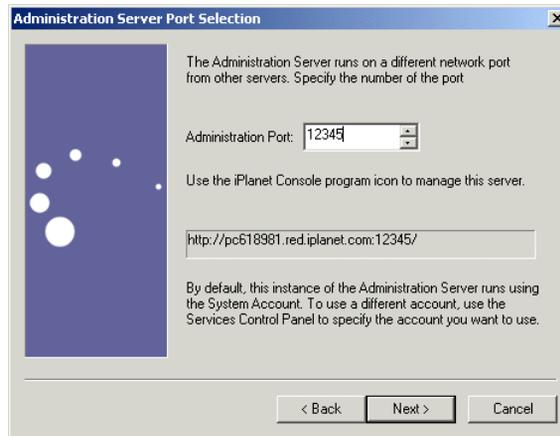
Directory Manager DN: cn=Directory Manager

Password: <dir mgr password>

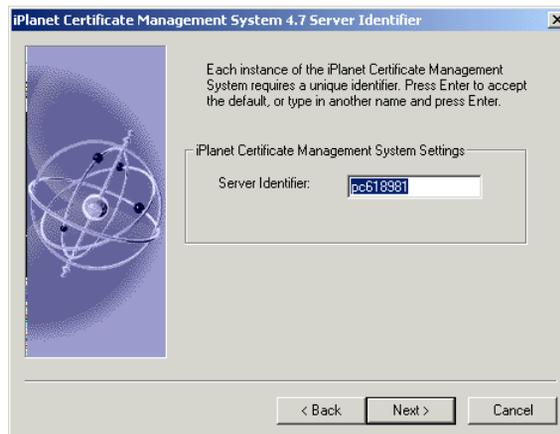
Password (again): <dir mgr password>



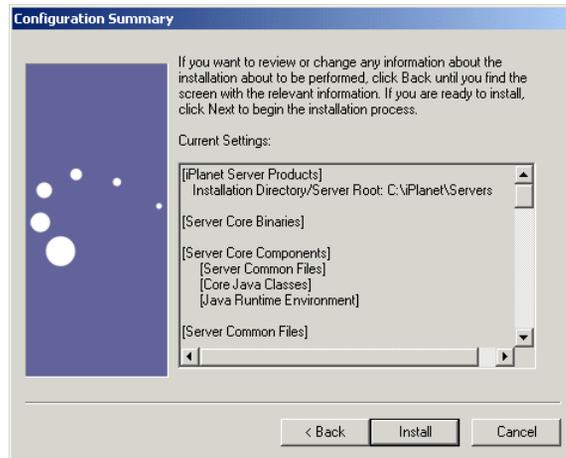
14. **Administration Server Port Selection.** Type the value 12345 and click Next.



15. **Certificate Management System Server identifier.** Accept the default, and click Next.

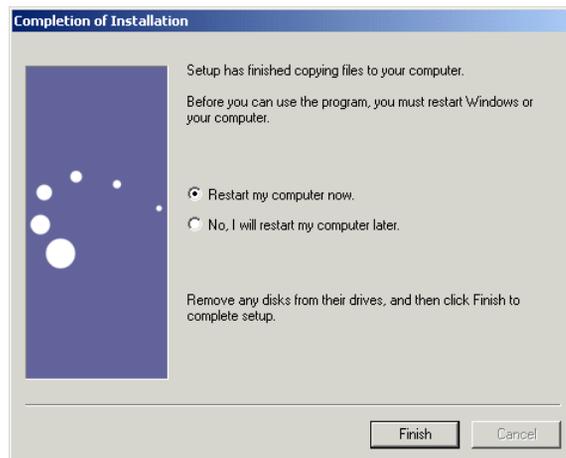


16. Configuration Summary. Click Next.



17. 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.

18. 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 iPlanet 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. If iPlanet Console is not running, start it.
 - On a Windows NT system, click Start, and then choose Programs, iPlanet, and iPlanet Console, in that order. Alternatively, click the iPlanet Console shortcut in the iPlanet directory that opens on your desktop after setup completes.
 - On a Unix system, open a command shell, change to the directory /usr/iplanet/servers, and execute the file startconsole.
2. Log in as admin, giving the password <admin password>.

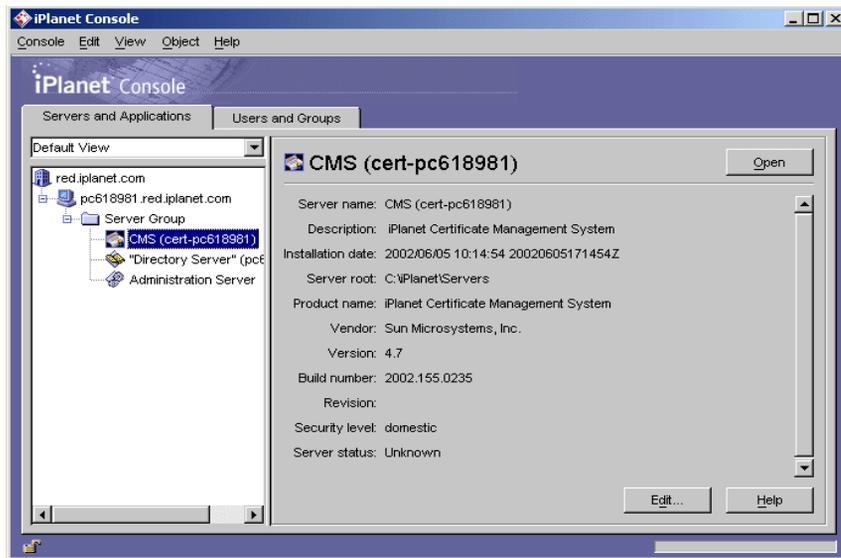
The main window of iPlanet Console appears.

If the Administration URL is not filled in, enter `http://<myhost>:12345`



3. In the navigation tree at the left, open your computer, then open Server Group.

4. Select `cert-<identifier>` and double-click it; alternatively, you can also click the Open button on the Certificate Management System panel on the right.



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.

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 (<identifier>-db).

Port number: Accept the default (38900).

Directory Manager DN: cn=Directory Manager

Password: <intdb password>

Password (again): <intdb password>

Installation Wizard

Internal Database

CMS needs access to an LDAP server instance to store requests and certificate records. This server instance is referred to as the internal database. You can either have CMS create a new instance for you, or use an existing directory. For security reasons, you should not delegate control of this directory to unauthorized persons.

Create a new Internal Database (recommended)

Instance ID: pc618981-db

Port number: 38900

Directory manager DN: cn=Directory Manager

Password:

Password (again):

Use an existing remote LDAP server

Host name:

Port number:

Base DN for this instance: o=pc618981-db, o=NetscapeCertificateServer

Directory manager DN: cn=Directory Manager

Password:

Database Name (for DS 5.x only):

Add CMS-Specific Schema and Indexes to Datab...

<Back Next> Cancel Help

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

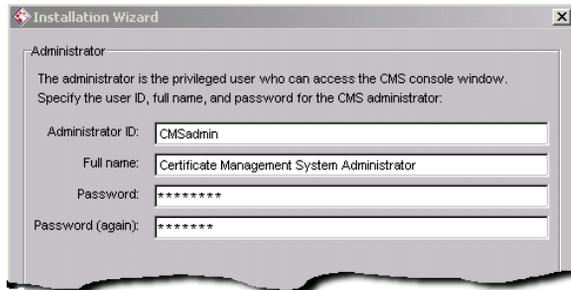
3. **Administrator.** Type the following values, then click Next:

Administrator ID: CMSadmin

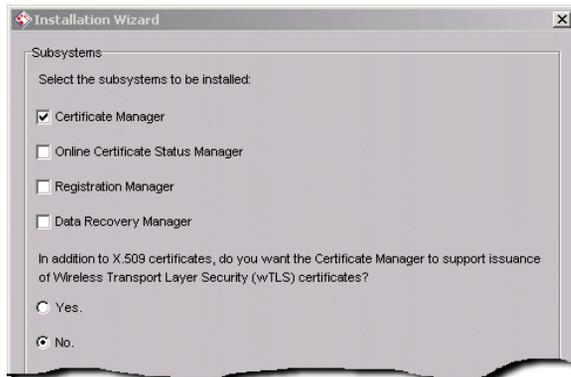
Full name: Accept the default value.

Password: <CMS password>

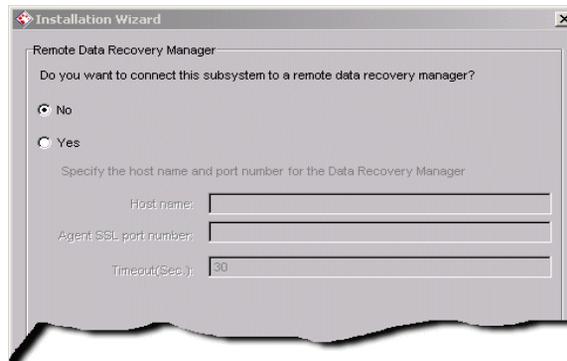
Password (again): <CMS password>



4. **Subsystems.** Click Next to accept the default selection (Certificate Manager only).

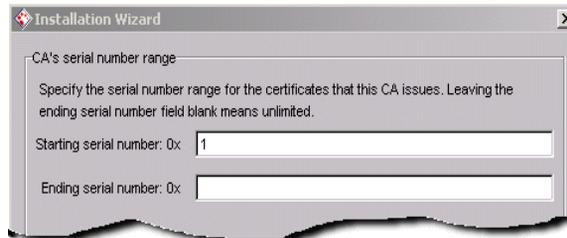


- 5. Remote Data Recovery Manager.** Click Next to accept the default selection (No).

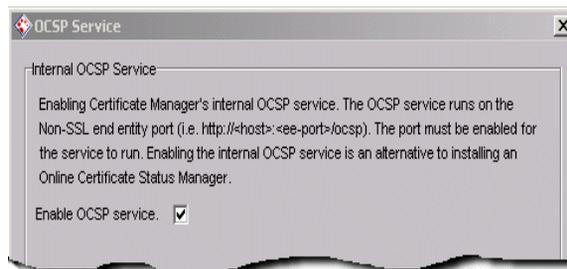


At this point the system configures the internal database, which can take some time.

- 6. CA's serial number range.** Click Next to accept the default (start at 0x1 with no upper limit).



- 7. Internal OCSP Service.** Click Next to accept the default (the option is selected).



- 8. 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.

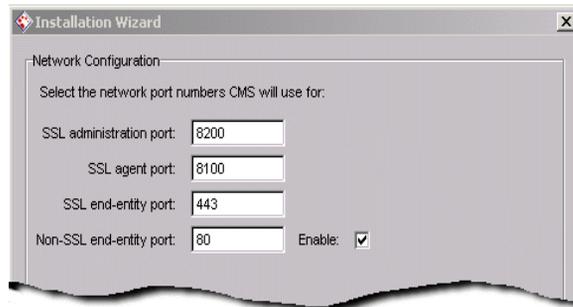
SSL administration port: 8200

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



- 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.

Installation Wizard

Key-Pair Information for Certificate Manager CA Signing Certificate

Select the token (cryptographic device) for the key pair:

Token:

FIPS Level 3

Initialize the selected token:

Password:

Password (again):

Security officer password:

Specify the key type and key length:

Key type:

Key length: bits

Enter a value for the customized key length:

11. **Message Digest Algorithm.** Click Next to accept the default (SHA1).

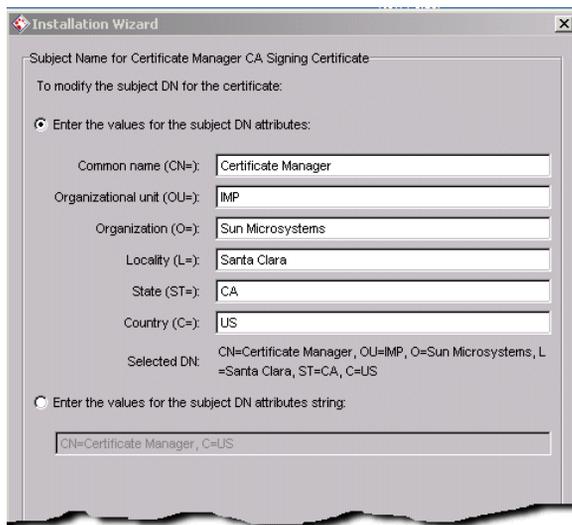
Installation Wizard

Message Digest Algorithm

Select hashing algorithm to use when computing signature on this certificate:

- Subject Name for Certificate Manager CA Signing Certificate.** Type the following values, then click Next:

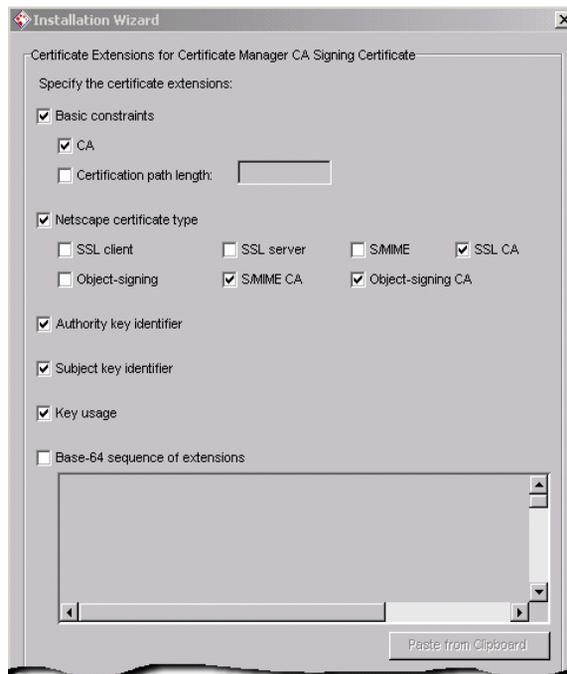
Common name (CN=): Certificate Manager
Organization Unit (OU=): <name of your organizaion>
Organization (O=): <name of your company>
Locality (L=): <name of your locality>
State (ST=): <name of your state, province, or territory>
Country (C=): <two-letter code for your country>



- 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.



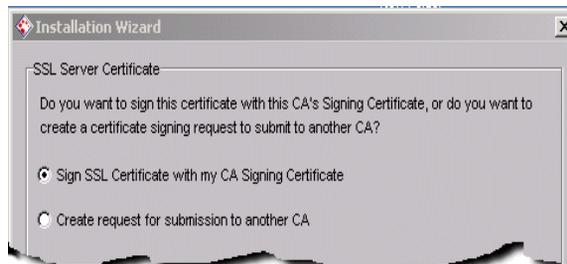
- 14. Certificate Extensions for Certificate Manager CA Signing Certificate.** Click Next to accept the default selections.



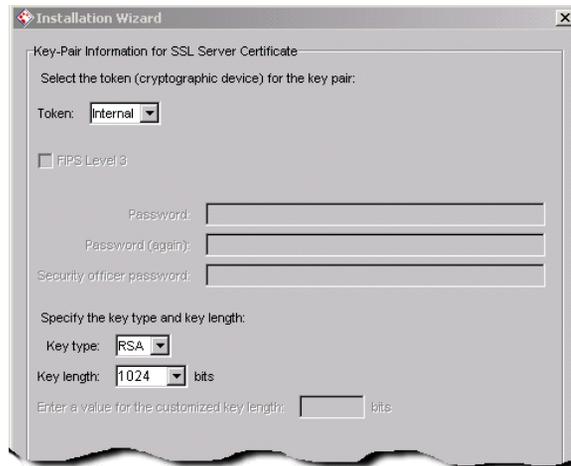
- 15. Certificate Manager CA Signing Certificate Creation.** Click Next.



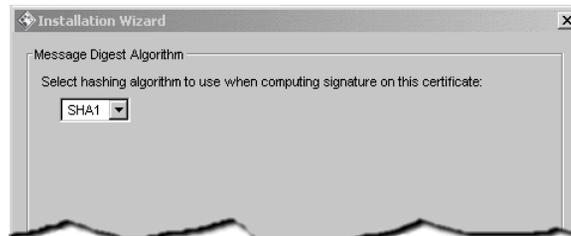
- 16. SSL Server Certificate.** Click Next to accept the default selection (Sign SSL certificate with my CA signing certificate selected.).



- 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=): <hostname, in the "machine.domain.com" form>
Organization Unit (OU=): <name of your organization>
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>

Installation Wizard

Subject Name for SSL Server Certificate

WARNING: If this is a clone CA, which resides on the same machine as the master CA, please be sure to make the DN different than that of the master's (this can be achieved by manipulating fields other than the cn attribute). Failure to do so will result in an error during startup.

To modify the subject DN for the certificate:

Enter the values for the subject DN attributes:

*Common name (CN=):

Organizational unit (OU=):

Organization (O=):

Locality (L=):

State (ST=):

Country (C=):

Selected DN: CN=pc618981.red.iplanet.com, OU=IMP, O=Sun Microsystems, L=Santa Clara, ST=CA, C=US

Enter the values for the subject DN attributes string:

- 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.

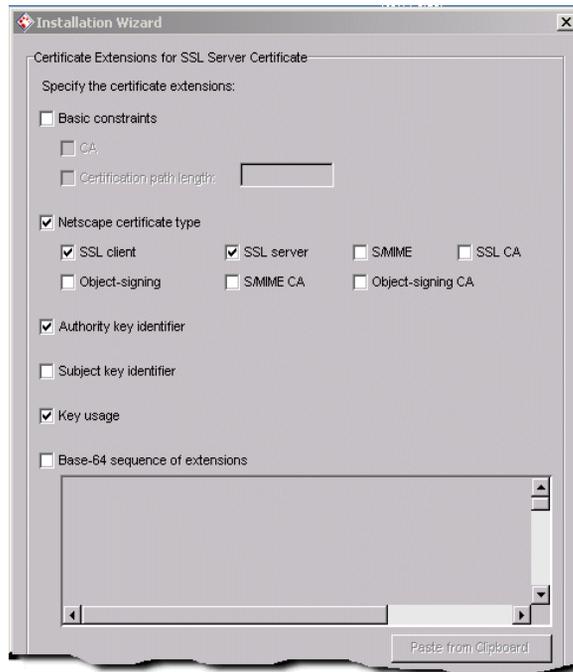
Installation Wizard

Validity Period for SSL Server Certificate

Specify the validity period for the certificate:

	YYYY	MM	DD	HH	mm	SS
Begin on:	2002	6	5	00	00	00
Expire on:	2004	6	5	00	00	00

- 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>

Installation Wizard

Create Single Sign-On Password

The single sign-on password is used to encrypt all the passwords that are required to start CMS.

Single sign-on password:

Single sign-on password (again):

Token for the Certificate Manager signing key and certificate:

Token name:

Token password:

Token for the SSL server key and certificate:

Token name:

Token password:

24. Configuration Status. Click Done. Certificate Management System starts automatically.

Installation Wizard

Configuration Status

You have successfully configured the Certificate Management System.
 You should now use a browser to connect to the agent port to get an agent certificate:
<https://pc618981.red.ipplanet.com:8100>

<Back Done Cancel Help

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 now 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://<machine_name>.<your_domain>.<domain>:8100`
- The SSL end-user gateway URL is:
`https://<machine_name>.<your_domain>.<domain>:443`
- The non-SSL end-user gateway URL is:
`http://<machine_name>.<your_domain>.<domain>: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 iPlanet 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://pc618981.red.iplanet.com:8100`.

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

The screenshot shows a Netscape browser window titled "Admin Enrollment form. - Netscape". The address bar shows the URL "https://pc618981.red.iplanet.com:8100/ca/adminEnroll.html". The main content area is titled "Administrator/Agent Certificate Enrollment" and contains the following text:

To access the Agent Services pages and approve requests for certificates, you must have a personal client SSL certificate so that Certificate Management System can authenticate your identity. You must also be designated as an agent, or privileged user.

Use this form to request this first personal certificate to be issued by the system. When you submit the form, the certificate is issued immediately and returned to you. The system also adds you automatically to the list of agents. You must import the new certificate into your browser before you can access the Agent Services pages.

After you submit this form, it is automatically disabled. To enroll again, or to enroll other users, please see the documentation.

Important: Be sure to request your certificate on the same computer on which you plan to use the certificate.

Authentication Information
Enter the user ID and password for the administrator/agent.

User ID:

Password:

Subject Name
Enter values for the DN components you want to have in your certificate.

Full name:

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 iPlanet 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 iPlanet Console, open `cert-demoCA`.
The server requests your <CMS password>.
2. Click the icon labeled "Stop the Server".
3. Go to this directory: `<server_root>/cert-demoCA/config`
4. Open the file `CMS.cfg` in a text editor, and find the following line:
`agentGateway.enableAdminEnroll=false`

5. Change `false` to `true`, and save the file.
6. Start the server from the CMS window where you stopped it.
Alternatively, right-click on `cert-demoCA` in the left frame and choose Start Server.
7. Enter your `<single-signon password>`.
The next time you access `https://<hostname>:8100`, the Administrative/Agent 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 139): Accessing the various web gateways and using the default versions of the forms to enroll for and issue a certificate.
- **“Create a Policy,”** (page 144): Configuring the Certificate Manager to reject certificate requests that do not use at least 1024-bit key lengths.
- **“Use an LDAP Directory,”** (page 146): Adding a user to the configuration directory you just installed and using directory-based authentication to enroll as that user.
- **“Publish Certificates to an LDAP Directory,”** (page 151): Publishing client certificates to the directory.
- **“Send Renewal Reminders,”** (page 157): 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://<hostname>: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 iPlanet 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 iPlanet 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://www.sun.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 Manager or Certificate Manager received the request. You can find out more about policies in Chapter 18, "Setting Up Policies."

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.

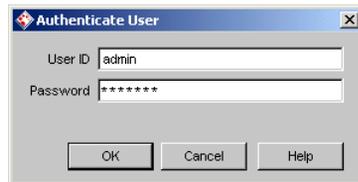
You will create a policy that requires all certificate requests use RSA key pairs that are 1024-bit or longer. This ensures that all of the certificates you issue meet a minimum level of security. Later, you will try to enroll for a certificate using a shorter-length key pair (512 bits) to show how the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

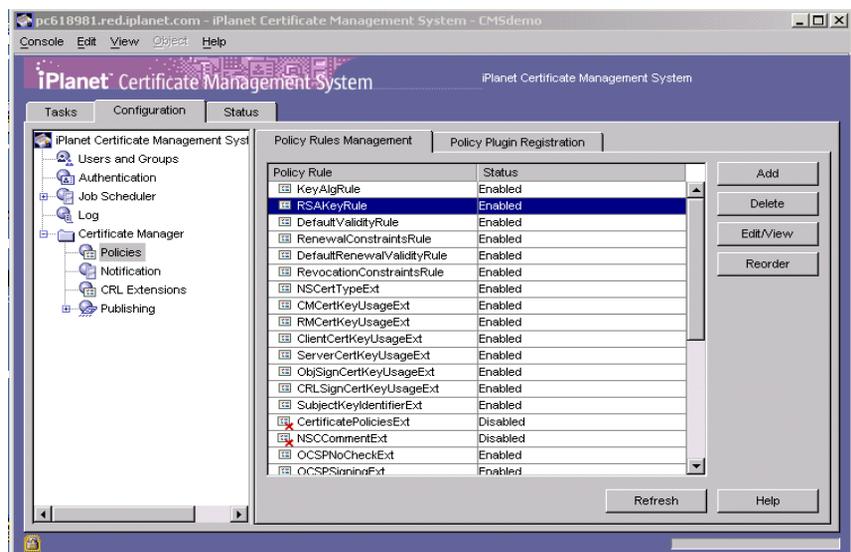
1. Start iPlanet Console:
 - o On a Windows NT system, click Start, then choose Programs, then iPlanet, then iPlanet Console.
 - o On a UNIX system, open a command shell, change to the directory `/usr/iplanet/servers`, and execute the file `startconsole`.

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



iPlanet 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:

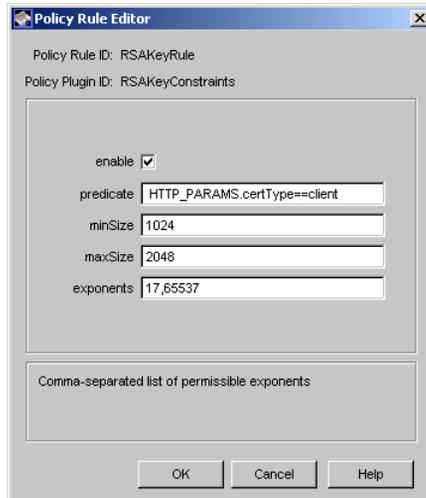
minSize: 1024

maxSize: 2048

exponents: accept the default setting

enable: true

predicate: HTTP_PARAMS.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 `RSAKeyRule` 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 iPlanet 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:

- Step 1. Enable Directory-Based Authentication
- Step 2. Add a User to the Directory
- Step 3. Enroll with Directory-Based Authentication

You can find out more about authentication in Chapter 15, “Setting Up End-User Authentication.”

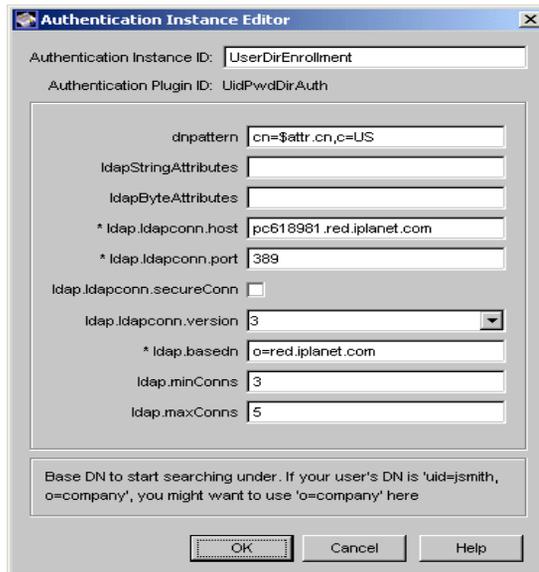
Step 1. Enable Directory-Based Authentication

To enable directory-based authentication for the Certificate Manager:

1. If the CMS console window is not still open, start iPlanet Console again (or go back to the main window) and open the window for Certificate Management System.
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: `<hostname>`
ldap.ldapconn.port: `389`
ldap.ldapconn.secureConn: `false`

ldap.ldapconn.version: 3
ldap.basedn: o=<your domain>.<domain>
ldap.minConns: 3
ldap.maxConns: 5



6. Click OK.

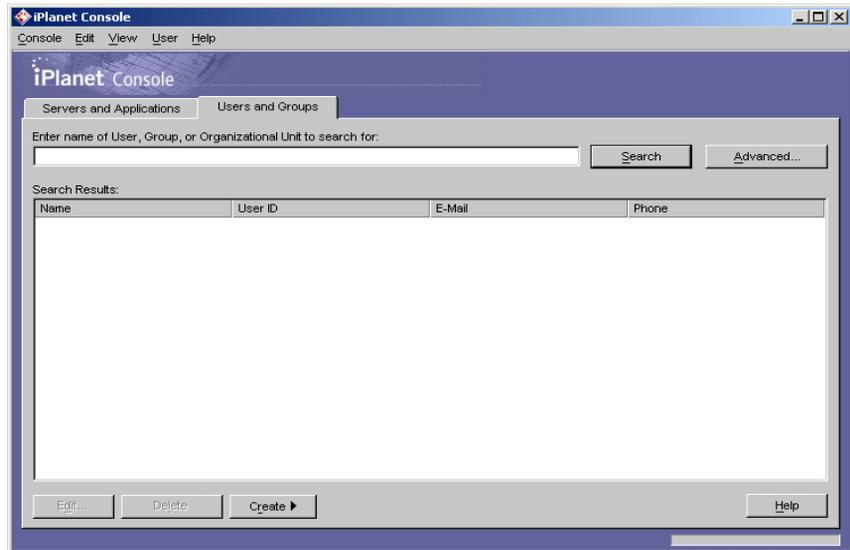
NOTE If you leave the `dnpattern` field blank, the `dnpattern` used by default is `E=$attr.mail,CN=$attr.cn,O=dn.o,C=$dn.c`. This pattern works well with Communicator and other browsers. For the demo, you used a simpler `dnpattern` to avoid configuring other things. The simpler pattern should not be 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.

Step 2. 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 iPlanet 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.

Step 3. 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 window and go to the Certificate Manager's end-entity interface: `https://<machine_name>.<your_domain>.<domain>:444`
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 to an LDAP 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 iPlanet 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 iPlanet Console again (or go back to the main Console window) and open the window for Certificate Management System.
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: <machine_name>.<your_domain>.<domain>

Port Number: 389

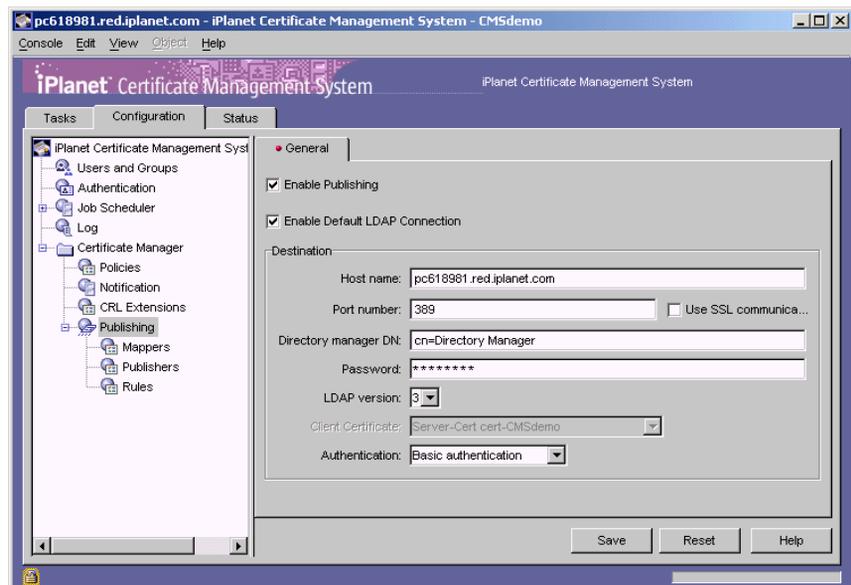
Directory Manager DN: cn=Directory Manager

Password: <dir mgr password>

Password (again): <dir mgr password>

Version: 3

Authentication: Basic authentication



5. Click Save.

A dialog box appears that indicates whether Certificate Management System 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.

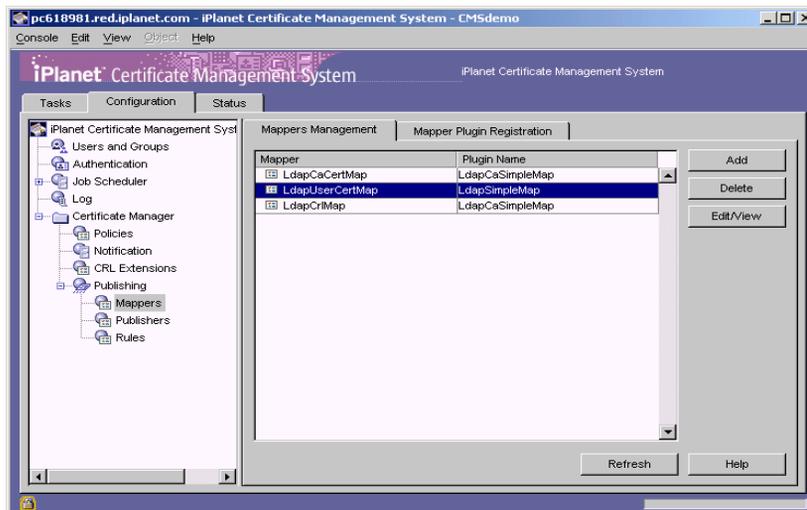
Directory publishing is now enabled. Certificate Management System will publish any new certificates to the directory according to the publication rules. The next step is to set those rules.

Set Rules for Publishing Certificates

In this section, you configure Certificate Management System to map client certificates to People entries in the `o=<your_domain.><domain>` directory tree using the user ID from the certificate request.

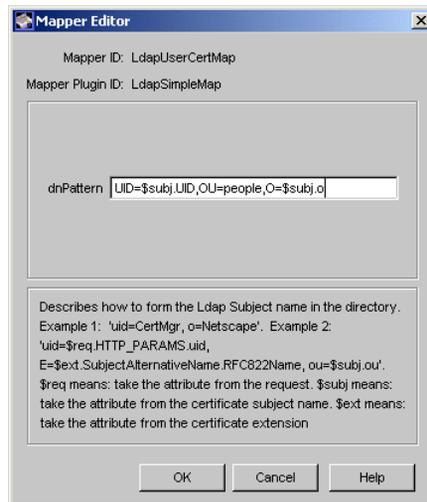
To configure Certificate Management System to publish user certificates to an LDAP directory:

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=<your domain>.<domain>`



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.

Certificate Management System 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 “Step 2. Add a User to the Directory” and “Step 3. 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 iPlanet 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, `siroe.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 Certificate Management System 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, Certificate Management System 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 Certificate Management System.

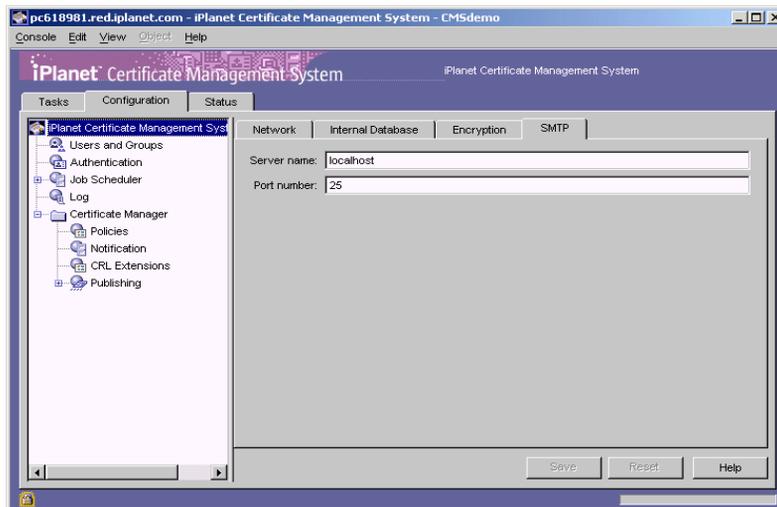
If you are sure that the machine on which Certificate Management System is running is also capable of receiving SMTP requests on port 25, skip to “Configuring Certificate Management System to Send Renewal Reminders.”

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

Configuring a Mail Server for Certificate Management System

To configure the server from which Certificate Management System 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 Certificate Management System 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>

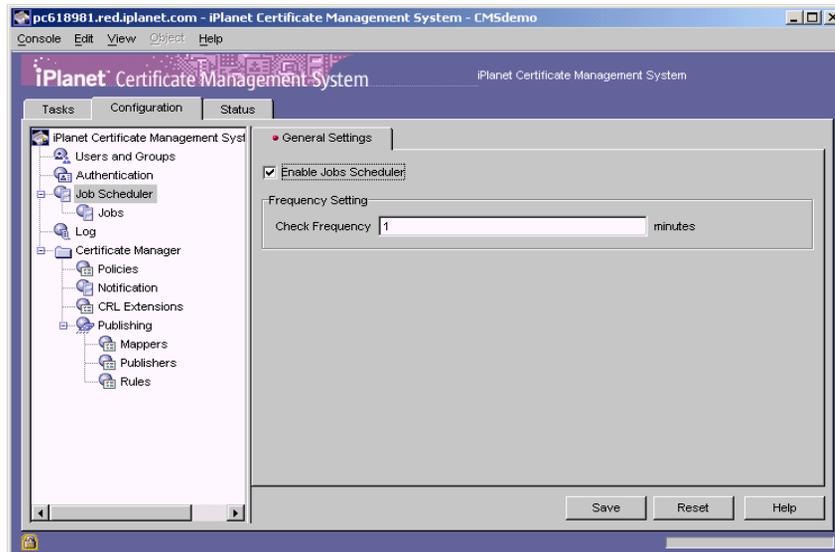
summary.senderEmail: <your email address>

The screenshot shows the 'Job Instance Editor' dialog box for the 'certRenewalNotifier' job. The 'Job Instance ID' is 'certRenewalNotifier' and the 'Job Plugin ID' is 'RenewalNotificationJob'. The 'enabled' checkbox is unchecked. The 'cron' field contains '0 3**1-5'. The 'notifyTriggerOffset' and 'notifyEndOffset' fields both contain '30'. The 'senderEmail' field contains 'jdoe@sun.com'. The 'emailSubject' field contains 'Certificate Renewal Notification'. The 'emailTemplate' field contains 'C:/Planet/Servers/cert-CMSdemo/emails/rn.Job1.txt'. The 'summary.enabled' checkbox is checked. The 'summary.recipientEmail' field contains 'jdoe@sun.com'. The 'summary.senderEmail' field contains 'jdoe@sun.com'. The 'summary.emailSubject' field contains 'Certificate Renewal Notification Summary'. The 'summary.itemTemplate' field contains 'C:/Planet/Servers/cert-CMSdemo/emails/rn.Job1Item.txt'. The 'summary.emailTemplate' field contains 'C:/Planet/Servers/cert-CMSdemo/emails/rn.Job1Summary.txt'. At the bottom, there is an 'Enable this Job' button and 'OK', 'Cancel', and 'Help' buttons.

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 4, “Planning Your Deployment” and the chapters that follow.

After you are finished using the demonstration installation, remove it from your system. For instructions, see “Uninstalling Certificate Management System” on page 311.

Using the Default Demo

Planning and Installation

Chapter 4, “Planning Your Deployment”

Chapter 5, “Installation Worksheet”

Chapter 6, “Installing Certificate Management System”

Chapter 7, “Installing and Uninstalling CMS Instances”

Chapter 8, “Starting and Stopping CMS Instances”

Planning Your Deployment

Before installing iPlanet Certificate Management Server (CMS) 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 5, "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 166)
- Certificate Authority Decisions (page 175)
- Cryptographic Token Decisions (page 179)
- Publishing Decisions (page 179)
- Subsystem Certificate Decisions (page 182)
- Authentication Decisions (page 185)
- Policy Decisions (page 185)
- Deployment Strategy and Port Assignments (page 186)

Topology Decisions

Certificate Management System allows you to install the Certificate Manager, Registration Manager, Data Recovery Manager, and Online Certificate Status 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 166)
- Single Certificate Manager (page 167)
- Certificate Manager and Registration Manager (page 168)
- Certificate Manager and Data Recovery Manager (page 170)
- Certificate Manager, Data Recovery Manager, and Registration Manager (page 172)
- Cloned Certificate Manager

Server Groups and CMS Instances

As described in *Managing Servers with iPlanet Console*, iPlanet servers installed in a single server root directory are called a *server group* and are managed by a single instance of iPlanet Administration Server. As shown in Figure 4-1, a CMS instance in a server group can contain a single subsystem 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

Other combinations are not permitted.

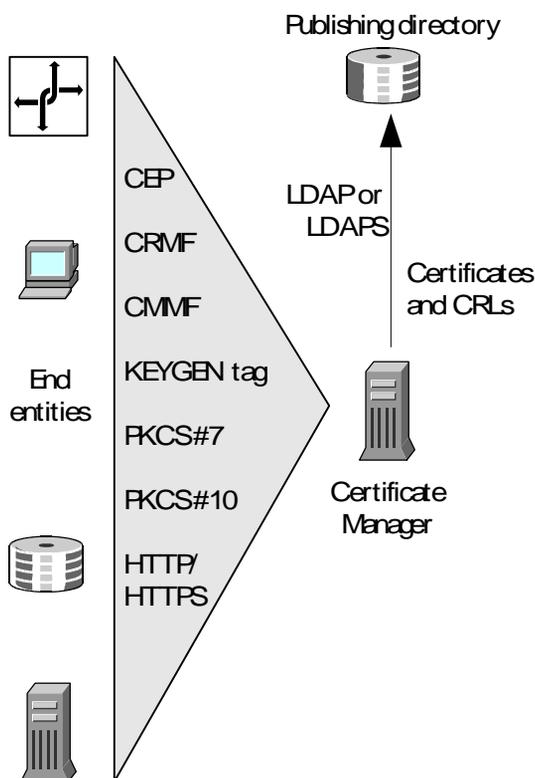
A Certificate Manager and a Registration Manager or Online Certificate Status Manager are always permitted in separate instances, whether the instances are in the same server group, in separate server groups on the same machine, or in separate server groups on separate machines.

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 4-1 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 4-1 Single root Certificate Manager



The arrangement shown in Figure 4-1 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

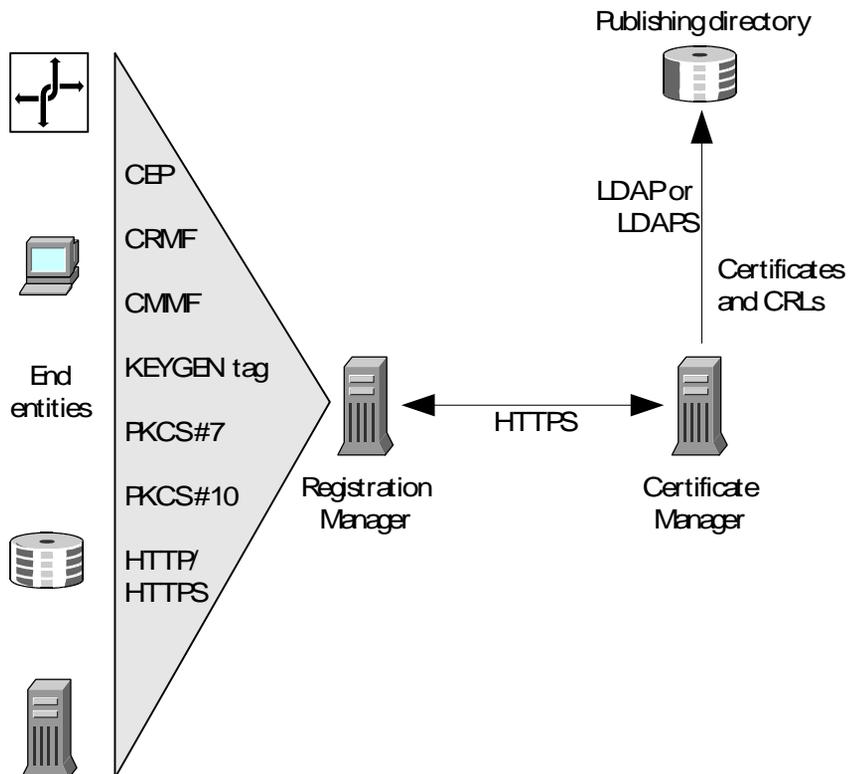
Many organizations need to separate the role of the Registration Manager from the role of the Certificate Manager. This separation can be useful, for example, if different groups of end entities are subject to different authentication policies or work in different geographic locations.

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, thus shielding the CA from end entities.

As stated earlier, 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 4-2 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 4-2 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.

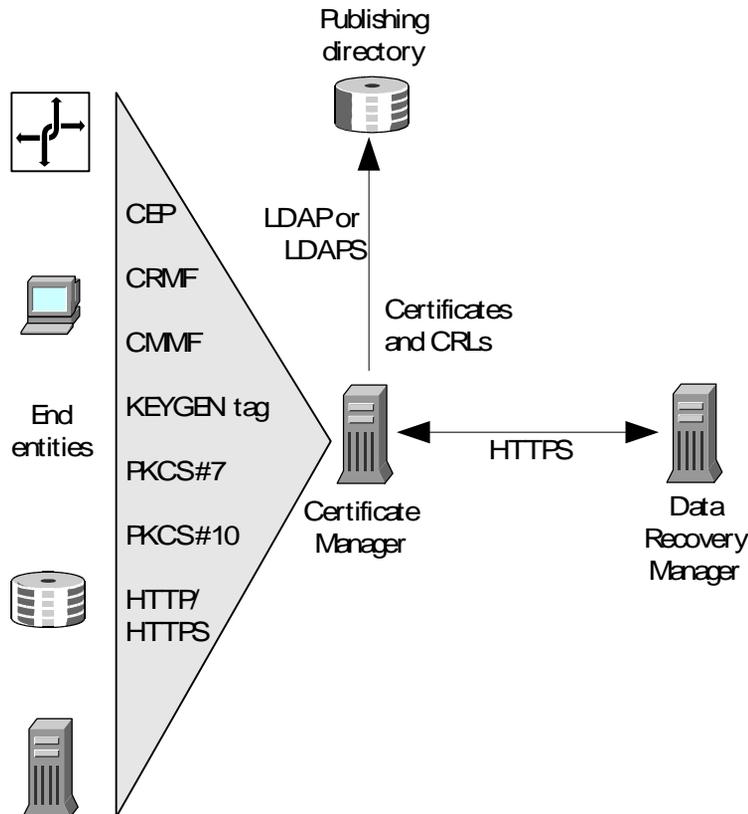
Decisions about the number of, locations of, and relationships among Certificate Managers and Registration Managers depend on many factors. These include firewall considerations, the physical security required for each subsystem, the physical location of the end entities that the Registration Manager 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 4-2, 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 4-3 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 or additional configurations.

Figure 4-3 Certificate Manager and Data Recovery Manager in different instances

The Data Recovery Manager is intended for archival and recovery of private encryption keys only. Therefore end entities must be using either a browser that supports dual-key generation or a browser that is using Netscape 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, 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

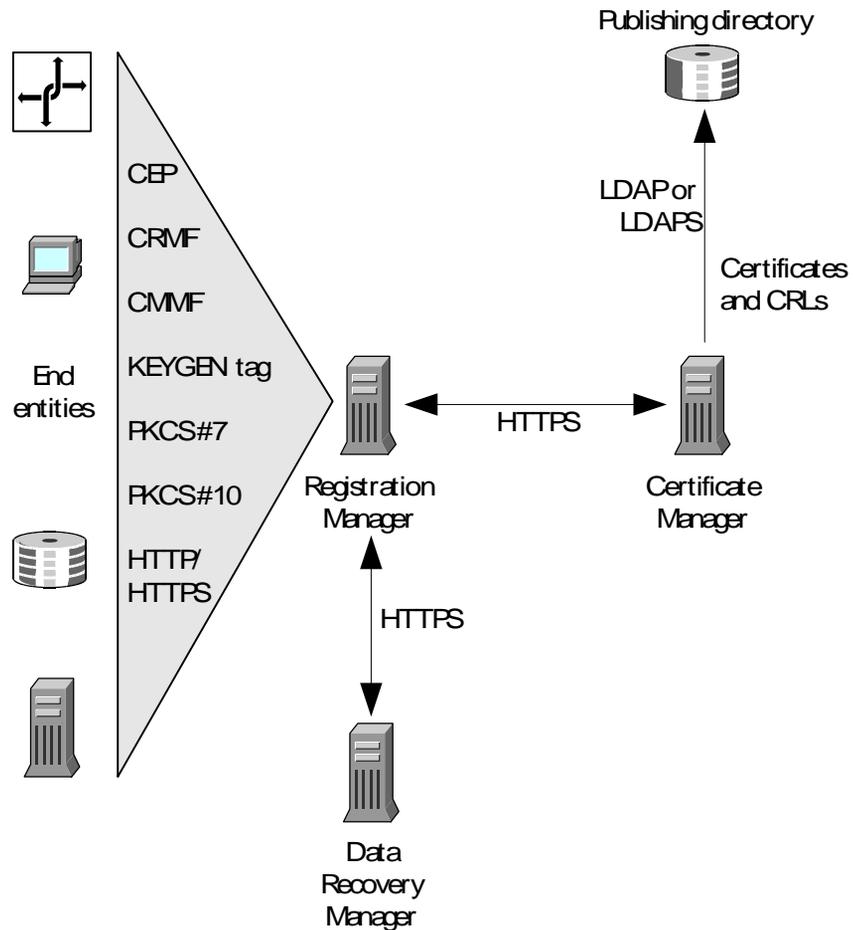
The three CMS subsystems can be deployed in many different relationships. Figure 4-4 illustrates some of the issues involved in deploying all three subsystems by showing the relationships among a single Certificate Manager, a single Registration Manager, and a single Data Recovery Manager, each installed 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.

Only the Certificate Manager can be configured to enable or disable LDAP publishing or to publish to separate directories. The Certificate Manager also has the complete record of issued certificates, so that it can perform the publishing tasks, 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 4-4 Certificate Manager, Registration Manager, and Data Recovery Manager in separate instances



NOTE The current design of 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 182).

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, identified as the *master* 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, but no 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 configuration

(`<server_root>cert-<instance_id>/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
- CA Signing Key Type and Length
- CA Signing Certificate's Validity Period
- Self-Signed Root Versus Subordinate CA
- CAs and Certificate Extensions
- CA Certificate Renewal or Reissuance

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 the Engineering department of a corporation named Siroe Corp: `cn=demoCA, o=Siroe Corp., 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 iPlanet Console*.

CA Signing Key Type and Length

If you wish, you can import the signing key and certificate used in a previous version of CMS installation rather than generating a new signing key pair.

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, check this document: <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 iPlanet 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. If you are using Netscape Communicator as your client, 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 set many of the standard extensions in the certificates the server issues.

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 C, “Certificate and CRL Extensions” of *CMS Plug-Ins Guide*.

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 subject 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 C, “Certificate and CRL Extensions” of *CMS Plug-Ins Guide*.

Cryptographic Token Decisions

As explained in “PKCS #11” on page 75, one or more PKCS #11 modules must be available to any CMS 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; see “Installing Level 2 External Tokens” on page 466.

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 1-10 on page 74, 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. Certificate Management System 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

A Certificate Manager can publish certificates to an LDAP directory and to files, and CRLs to an LDAP directory, files, and the Online Certificate Status Manager.

Publishing to Certificates and CRLs to Files

Any Certificate Manager that publishes certificates or CRLs to files need to specify the location for storing these files. There will be a file for each certificate and CRL, so the specified location must have sufficient disk space for storing these files. For detailed information on publishing certificates and CRLs to files, see Chapter 20, “Publishing Certificates and CRLs to a File.”

Publishing to Certificates and CRLs to a Directory

Any Certificate Manager that publishes certificates or CRLs to a directory must specify the host name and port number for the directory and indicate whether communication should take place over SSL. The Certificate Manager must also specify how it 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 Certificate Manager in the specified manner.

Note that it's not possible to configure the Registration Manager to publish certificates or CRLs. The Certificate Manager has the complete record of issued certificates and that the 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, iPlanet 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 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 or CRL, you will also need an entry for the CA.

If you intend to use SSL authentication, both the directory and the Certificate Manager must be configured appropriately for SSL. For detailed information on LDAP publishing, see Chapter 19, "Setting Up LDAP Publishing."

Publishing CRLs to the Online Certificate Status Manager

Certificate Management System supports the Online Certificate Status Protocol (OCSP) as defined in the PKIX standard RFC 2560 (see <http://www.ietf.org/rfc/rfc2560.txt>). The OCSP protocol enables OCSP-compliant applications to determine the state of a certificate, including the revocation status, without having to directly check a CRL published by a CA to the validation authority. The validation authority, which is also called an *OCSP responder*, does the checking for the application. For more information, see "What's an OCSP-Compliant PKI Setup?" on page 714.

To aid you in the process of setting up a OCSP-compliant PKI setup, Certificate Management System provides two options:

- Use the OCSP-service feature built into the Certificate Manager
- Use the CMS OCSP responder, named Online Certificate Status Manager

Read section "How to Get an OCSP Responder?" on page 716 to decide which method is suitable for your PKI setup.

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, Registration Manager, and Online Certificate Status Manager each requires its own signing certificate, and the Data Recovery Manager needs its own transport certificate and storage key.

For more information about the key pairs and certificates used by the CMS managers, see “Keys and Certificates for the Main Subsystems” on page 450.

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.

In addition to these certificates, the Certificate Manager also generates a few other certificates transparently during installation. For details, see "Certificate Manager's Key Pairs and Certificates" on page 451.

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.

For more information about the key pairs and certificates used by a Registration Manager, see "Registration Manager's Key Pairs and Certificates" on page 459.

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.

NOTE If you want to use hardware tokens for generating and storing Data Recovery Manager's key pairs, you'll need at least two tokens: one exclusively for the storage key pair and the other for the remaining key pairs. Be sure to install (and initialize, if required) these tokens before you start the Data Recovery Manager installation.

For more information about the key pairs and certificates used by a Data Recovery Manager, see "Data Recovery Manager's Key Pairs and Certificates" on page 460.

Online Certificate Status Manager Certificates

Every Online Certificate Status Manager must have a signing certificate whose public key corresponds to the private key the Online Certificate Status Manager uses to sign OCSP responses before sending them to OCSP-compliant clients. The Online Certificate Status Manager's signature provides persistent proof to an OCSP-compliant client that the Online Certificate Status Manager has processed the request.

The Online Certificate Status Manager also requires at least one SSL server certificate. For more information about the key pairs and certificates used by a Online Certificate Status Manager, see “Online Certificate Status Manager’s Key Pairs and Certificates” on page 463.

Authentication Decisions

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 Chapter 2, “Certificate Enrollment and Life-Cycle Management.”

For a detailed overview of authentication management using Certificate Management System, see Chapter 15, “Setting Up End-User Authentication.”

Policy Decisions

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=Siroe Corp.` 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 18, “Setting Up Policies.”

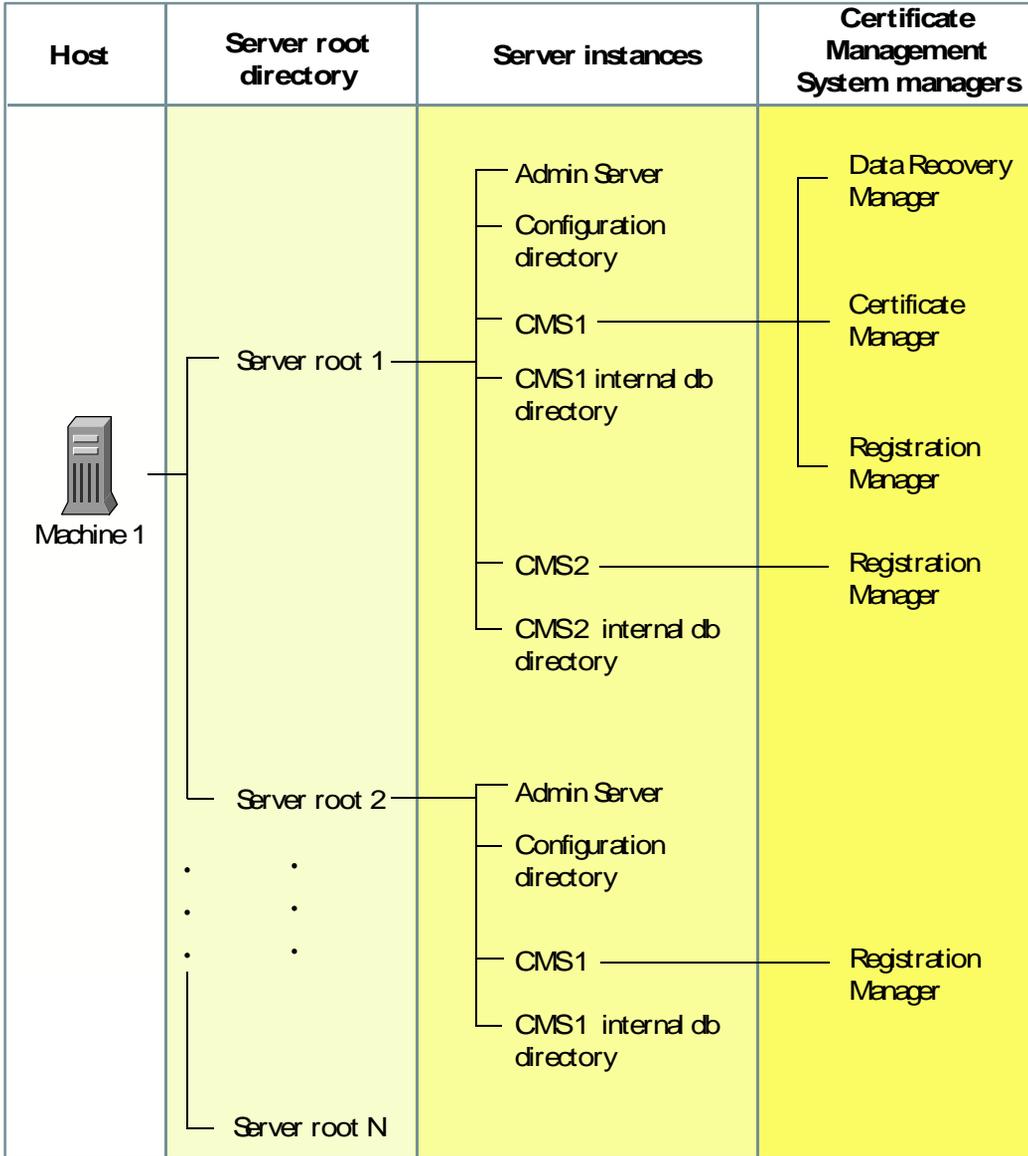
Deployment Strategy and Port Assignments

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, Registration Managers, and Online Certificate Status 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 5, “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 4-5 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 4-5 Deploying servers on a single host



Each server root directory shown in Figure 4-5 has its own Administration Server and iPlanet 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 3-1 on page 110 illustrates the ports used by a single CMS 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 require at least four unique ports:

- Internal database port for communication with internal database
- SSL administration port for communication with iPlanet 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 Directory Server being used as 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 7, “Installing and Uninstalling CMS Instances.”

Installation Worksheet

This chapter provides a worksheet to help you prepare for installing a single instance of iPlanet Certificate Management Server (CMS). 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 4, “Planning Your Deployment” It is designed for easy reference while you are following the procedures described in Chapter 6, “Installing Certificate Management System.”

CAUTION 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 190)
- Information for NT Installation Script (page 193)
- Initial Configuration (page 196)
- Certificate Manager Configuration (page 199)
- Registration Manager Configuration (page 203)
- Data Recovery Manager Configuration (page 205)
- Online Certificate Status Manager Configuration (page 209)
- Cloned Certificate Manager Configuration (page 211)
- SSL Server Certificate Configuration (page 213)
- Single Sign-On Password (page 216)

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 iPlanet Console application and have access to a configuration and user/group directory. For more information on the iPlanet server environment, see *Managing Servers with iPlanet 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/iplanet/servers`.

- Computer name _____

The default should be the fully qualified host name of the machine on which the installation is taking place. For example, `mydirectory.siroe.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 iPlanet 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.siroe.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 iPlanet 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=siroe.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, `siroe.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. For the name, you can use any combination of letters (aA to zZ), digits (0 to 9), an underscore (_), and a hyphen (-); other characters and spaces are not allowed. For example, you can type `pilotCA`, `pilot_CA`, or `pilot-CA` as the instance name, but not `pilot CA`.

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 iPlanet Console application and have access to a configuration and user/group directory. For more information on the iPlanet server environment, see *Managing Servers with iPlanet Console*.

- Installation directory (Server root directory) _____

The default installation directory is `C:\iPlanet\Servers`. 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. 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.
- Suffix _____
If you are creating a new directory, this should be the domain name of the current host. For example, `o=siroe.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, `siroe.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. For the name, you can use any combination of letters (aA to zZ), digits (0 to 9), an underscore (`_`), and a hyphen (`-`); other characters and spaces are not allowed. For example, you can type `cmsdemo`, `cms_demo`, or `cms-demo` as the instance name, but not `cms demo`.

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 iPlanet 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. Certificate Management System uses LDAP 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 iPlanet 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 to install any individual manager, Certificate Manager and Data Recovery Manager together, or Registration Manager and Data Recovery Manager together. Other combinations are not allowed, for example, you cannot install a Certificate

Manager and Registration Manager together or Certificate Manager and Online Certificate Status 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.

In addition to x.509 certificates, the Certificate Manager can also issue Wireless Transport Layer Security (wTLS)-compliant certificates for wireless applications. If you want this feature, you must choose the appropriate option. If you enable issuance of wTLS certificates, the Certificate Manager generates a wTLS CA signing certificate and installs the appropriate HTML interfaces for users to request certificates for wireless applications.

- Certificate Manager _____
Enable issuance of wTLS certificates? _____
- Registration Manager _____
- Data Recovery Manager _____
- Online Certificate Status 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 186.

- 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).

CA Signing Certificate

When you install the Certificate Manager, 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 176.

- Token for storing the Certificate Manager CA signing certificate and private key _____

Enter either `internal` (if you plan to use the internal/software 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 wizard, you can select from a list of already installed and available tokens. For example, `SmartCard`.

- Token password _____

The password for the token must be at least eight character long.

- Key type _____

RSA or DSA.

- Key length _____

Available key sizes for RSA are 512, 1024, 2048, 4096, or custom. Available key sizes for DSA are 512, 1024, or custom (must be in increments of 64 bit).

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 175.

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 (version 4.6 or below) to crash. For more information about distinguished names, see Appendix A, “Distinguished Names,” in *CMS Plug-Ins Guide*. To locate this guide, see “Where to Go for Related Information” on page 29.

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. The default validity period is two years.

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 and additional/custom extension by pasting its base-64 encoding in the space provided on this screen. For more information about extensions, see Appendix C, “Certificate and CRL Extensions” of *CMS Plug-Ins Guide*.

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

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

- 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 Certificate Management System, you will need to paste the base-64 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, 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 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 wizard, you can select from a list of already 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 key sizes for RSA are 512, 1024, 2048, 4096, or custom. Available key sizes for DSA are 512, 1024, or custom (in increments of 64 bits only).
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 (version 4.6 or below) to crash. For more information about distinguished names, see Appendix A, "Distinguished Names," in *CMS Plug-Ins 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 a Certificate Manager _____

Enter the URL for the end-entity gateway of the Certificate Manager that will issue the Registration Manager'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).

NOTE If you want to use hardware tokens for generating and storing Data Recovery Manager's key pairs, you'll need at least two tokens: one exclusively for the storage key pair and the other for the remaining key pairs. Be sure to install (and initialize, if required) these tokens before you start the Data Recovery Manager installation.

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 176.

- 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 wizard, you can select from a list of already 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 key sizes for RSA are 512, 1024, 2048, 4096, or custom. Available key sizes for DSA are 512, 1024, or custom (in increments of 64 bits only).
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 (version 4.6 or below) to crash. For more information about distinguished names, see Appendix A, "Distinguished Names," in *CMS Plug-Ins 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 C, “Certificate and CRL Extensions” of *CMS Plug-Ins Guide*.

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 Certificate Management System, 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 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, 2048, or 4096.

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 _____

Online Certificate Status Manager Configuration

This section summarizes information required to configure a Online Certificate Status Manager.

Online Certificate Status Manager Signing Certificate Request

When you install a Online Certificate Status Manager, you must supply information for the certificate that the Online Certificate Status Manager will use to sign OCSP responses. 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 Online Certificate Status Manager Signing Certificate

- Token for storing the Online Certificate Status Manager 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 wizard, you can select from a list of already 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 key sizes for RSA are 512, 1024, 2048, 4096, or custom. Available key sizes for DSA are 512, 1024, or custom (in increments of 64 bits only).

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 Online Certificate Status 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 Online Certificate Status Manager signing certificate. For more information about distinguished names, see Appendix A, “Distinguished Names,” in *CMS Plug-Ins Guide*.

Online Certificate Status 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 a Certificate Manager _____

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

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.

CA Signing Certificate

When you install the Certificate Manager, 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 iPlanet Certificate Management Server, 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 wizard, you can select from a list of already 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 iPlanet Certificate Management Server, available settings for RSA are 512, 1024, 2048, 4096, or custom, and available settings for DSA are 512, 1024, or custom (in increments of 64 bits only).
- 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 (version 4.6 or below) to crash. For more information about distinguished names, see Appendix A, "Distinguished Names," in *CMS Plug-Ins 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 the 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 _____ to _____
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 C, “Certificate and CRL Extensions” of *CMS Plug-Ins Guide*.

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)_____
 - 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 Certificate Management System, 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 iPlanet Certificate Management Server 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 _____

Installing Certificate Management System

This chapter describes the procedure for installing a iPlanet Certificate Management Server (CMS) instance. Before you use this chapter to guide you through an installation, you should have read Chapter 1 through Chapter 5 and filled out the worksheet provided by Chapter 5, “Installation Worksheet.”

This chapter contains the following sections:

- Installation Overview (page 218)
- Stage 1. Running the Installation Script (page 221)
- Stage 2. Running the Installation Wizard (page 227)
- Stage 3. Enrolling for Administrator/Agent Certificate (page 277)
- Stage 4. Further Configuration Options (page 283)
- Stage 5. Creating Additional Instances or CA Clones (page 284)

Installation Overview

Before you begin installation, make sure your system meets the requirements listed in the Release Notes for the product version at:

http://docs.sun.com/?p=coll/S1_s1CertificateServer_47

The installation process installs the iPlanet Administration Server, iPlanet Console, and iPlanet Directory Server, as well as 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 iPlanet 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 CMS installation. These procedures are described in “Stage 1. Running the Installation Script” on page 221.
- **Stage 2:** Run the Installation Wizard to set up the initial configuration of the CMS 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. Running the Installation Wizard” on page 227.
- **Stage 3:** Use iPlanet Console to further configure the new Certificate Management System instance, as needed. See “Stage 4. Further Configuration Options” on page 283.
- **Stage 4 (optional):** Use iPlanet 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 5. Creating Additional Instances or CA Clones” on page 284.

Before You Begin the Installation

Before you start installing Certificate Management System, follow these instructions:

- If you're not familiar with Certificate Management System, you might find it useful to run a demo installation first; see Chapter 3, "Default Demo Installation."
- If you want to install the Certificate Manager as a root CA:
 - Read and fill in the information requested in the Certificate Manager installation worksheet; see "Certificate Manager Configuration" on page 199.
 - Decide whether you want to create clones of the CA. If you do, determine the serial number ranges for each CA.
- If you want to install the Certificate Manager as a subordinate CA:
 - Read and fill in the information requested in the Certificate Manager installation worksheet; see "Certificate Manager Configuration" on page 199.
 - Identify the CA to which you'll submit the subordinate CA's *CA signing certificate* and *SSL server certificate* requests. Make sure the CA is running and, if required, identify the forms you'll use to submit these requests.
 - If the CA is a third-party CA, familiarize with the enrollment interface of that CA and check how long does the CA take to send you the certificates.
 - Decide whether you want to create clones of the CA. If you do, determine the serial number ranges for each CA.
- If you want to install a standalone Registration Manager, do this:
 - Read and fill in the information requested in the Registration Manager installation worksheet; see "Registration Manager Configuration" on page 203.
 - Identify the CA to which you'll submit the Registration Manager's *signing certificate* and *SSL server certificate* requests. Make sure the CA is running and, if required, identify the forms you'll use to submit these requests.
- If you want to install a standalone Data Recovery Manager:
 - Read and fill in the information requested in the Data Recovery Manager installation worksheet; see "Data Recovery Manager Configuration" on page 205.

- Identify the CA to which you'll submit the Data Recovery Manager's *transport certificate* and *SSL server certificate* requests. Make sure the CA is running and, if required, identify the forms you'll use to submit these requests.
- If you plan to use hardware tokens for generating and storing Data Recovery Manager's key pairs, you'll need at least two tokens: one exclusively for the storage key pair and the other for the remaining key pairs. Be sure to install (and initialize, if required) these tokens before you start the Data Recovery Manager installation. For installation instructions, see "Installing Level 2 External Tokens" on page 466.
- If you want to install a standalone Online Certificate Status Manager:
 - Read and fill in the information requested in the Online Certificate Status Manager installation worksheet; see "Online Certificate Status Manager Configuration" on page 209.
 - Identify the CA to which you'll submit the Online Certificate Status Manager's *signing certificate* and *SSL server certificate* requests. Make sure the CA is running and, if required, identify the forms you'll use to submit these requests. For Online Certificate Status Manager's signing certificate to work properly, it must contain the following extensions:

OCSPNoCheck extension—Presence of this extension indicates that an OCSP client should not use OCSP to check the revocation status of the OCSP responder certificate, because the certificate is only used to identify the responder that does the checking. (This extension is required to avoid a circular reference.) For details about this extension, see section "OCSPNoCheckExt Plug-in Module" of *CMS Plug-Ins Guide*.

OCSPSigning extension—This is an *Extended Key Usage* extension with a unique value, OCSPSigning. Presence of this extension indicates that the key pair that corresponds to the certificate used by the OCSP responder can be used for signing OCSP responses. For details about this extension, see section "OCSPSigningExt Rule" of *CMS Plug-Ins Guide*.

Make sure the Certificate Manager to which you'll submit the Online Certificate Status Manager's signing certificate request has these policies enabled.

- If you want to install two subsystems in a CMS instance, for example, a Certificate Manager along with a Data Recovery Manager, collect the information for both the subsystems.

Stage 1. Running the Installation Script

The `setup` program extracts files for the Administration Server, Directory Server, iPlanet 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 iPlanet 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 5, “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 iPlanet servers.
4. **Install location [/usr/iplanet/servers]:** 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 iPlanet 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. iPlanet configuration directory server administrator ID [admin]:** Enter the name and password of the user who will authenticate to iPlanet 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. Running the Installation Wizard” on page 227.

Running the Installation Script on Windows NT

The `setup.exe` program extracts files for the Administration Server, Directory Server, iPlanet 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 iPlanet 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 5, “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.** “iPlanet Servers” is selected by default. Click Next to accept the default selection.
4. **Choose Installation Directory.** The default installation directory is `C:\iPlanet\Servers`. 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:
 - iPlanet Server Products Core Components.
 - iPlanet Directory Suite
 - Administration Services
 - iPlanet Certificate Management System

You don't need to select the fifth component, Directory Server Synch Service, unless you want to set up the Directory Server Synchronization Service. Click Next to accept the default selection.

6. **Directory Server 4.13.** “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 4.13.** “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 4.13 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 5.1 iPlanet 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 4.13 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 4.13 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.** A randomly selected port number will be shown. 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. **iPlanet 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 iPlanet 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. Running the Installation Wizard” on page 227.

Stage 2. Running 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—you use the wizard to get the initial certificates and set the initial configuration for this instance of Certificate Management System. The Installation Wizard is the same for both UNIX and Windows NT.

In the last step of the installation script, you were given an opportunity to specify whether to launch iPlanet Console. If you chose to launch iPlanet Console, you’ll see iPlanet Console open automatically. Otherwise, you need to open it manually. To bring up iPlanet Console and launch the Installation Wizard, follow these steps:

1. Start iPlanet Console:

On a Windows NT system, click Start, and then choose Programs, iPlanet, and iPlanet Console, in that order. Alternatively, click the corresponding shortcut in the iPlanet Server Products directory window displayed after setup completes.

On a UNIX system, open a command shell, change to the directory `/usr/ipplanet/servers`, 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 iPlanet Console appears.

3. In the navigation tree at the left, open your computer, then open Server Group.
4. Select the CMS instance that you named while running the installation script.

5. In the Certificate Management System panel at the right, click Open.

After a few moments, the Introduction screen for the Installation Wizard appears.

Click Next to continue. The Internal Database screen appears.

6. In the Internal Database screen, specify the Directory Server instance that Certificate Management System should use as its internal database—you may choose to create a new Directory Server instance or use an existing Directory Server instance. The Directory Server instance you choose will be used as a database to store information (such as certificates and certificate requests) used by all the subsystems you will be installing in this CMS instance. It's recommended that you do not use this Directory Server instance for any other purposes; the directory schema will be configured for storing CMS data.

Click Next to continue. The wizard sets up the new internal database, which takes some time.

(If you have previously installed an internal database for this instance, the Recreate Internal Database screen appears. In the 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.

A special screen, Internal Database password, 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.)

7. In the Administrator screen, type the ID, name and password for the CMS administrator. This is the administrator who can access the CMS window and control all CMS settings.

Click Next to continue.

The “Subsystems” screen appears. This screen enables you to choose a subsystem or the permitted combinations of subsystems you want to install. Depending on what you want to install, follow the appropriate instructions.

- Installing the Certificate Manager as a Root CA
- Installing the Certificate Manager as a Subordinate CA
- Installing a Standalone Registration Manager
- Installing a Standalone Data Recovery Manager
- Installing an Online Certificate Status Manager

Installing the Certificate Manager as a Root CA

To install the Certificate Manager as a root CA:

1. **Subsystems.** Select Certificate Manager. If you want the Certificate Manager to issue certificates for wireless applications, select the “In addition to X.509 v3 certificates, do you want the Certificate Manager to support issuance of Wireless Transport Layer Support (wTLS)-compliant certificates” option. Otherwise, leave the option unchecked. (If you select the option, the end-entity interface will include two forms for requesting certificates for wireless applications and an option for downloading the wireless CA certificate.)

Click Next to continue.

2. **Remote Data Recovery Manager.** Select the appropriate options:
 - Select No, if you don’t want to connect the Certificate Manager to a remote Data Recovery Manager.
 - If you have already installed a remote Data Recovery Manager that you want the Certificate Manager to use for archiving end users’ encryption private keys, select Yes. Then, enter the remote Data Recovery Manager’s host name and agent SSL port number in the associated fields.

Click Next to continue.

3. **Network Configuration.** Type the port numbers for the ports to be used by the CMS instance. If you want to enable the non-SSL end-entity port, be sure to check the “Enable” checkbox.

Click Next to continue.

4. **CA’s Serial Number Range.** Specify range for the serial numbers. In the “Starting serial number” field, type the lowest serial number the CA should assign to a certificate. If you plan to only use one CA server, you can leave the “Ending serial number” field blank to indicate no upper limit. If you plan to clone the CA to distribute load, you must specify an upper limit. (For cloned CAs, you must make sure that the range of serial numbers does not overlap with any other CA server.)

Click Next to continue.

5. **CA Signing Certificate.** Select the “Create self-signed CA certificate” option.

Click Next to continue.

- 6. Key-Pair Information for Certificate Manager CA Signing Certificate.** Select the token to store the root CA signing certificate and key pair. If you have not previously initialized the token's password, you must do so in this screen. Also specify the key type and length.

Click Next to continue.

- 7. Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature. The choices are: MD2, MD5, or SHA-1.

Click Next to continue.

- 8. Subject Name for Certificate Manager CA Signing Certificate.** Type values for the subject DN components; these values identify the root CA signing certificate.

Click Next to continue.

- 9. Validity Period for Certificate Manager CA Signing Certificate.** Select the validity period for the CA signing certificate. The default validity is two years. The validity period determines how soon you will have to renew the certificate, which can be a complex procedure.

Click Next to continue.

- 10. Certificate Extensions for Certificate Manager CA Signing Certificate.** Select the required extensions. 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.

Certificate Management System provides command-line tools for generating extensions to include in CA and other certificate requests. For details about these tools, check this directory: `<server_root>/bin/cert/tools`

Note that 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 in the `tools` directory. For details on using the `ExtJoiner` program, see Chapter 5, "Extension Joiner Tool" of *CMS Command-Line Tools Guide*.

Click Next to continue.

- 11. Certificate Manager CA Signing Certificate Creation.** Click Next to generate and install the certificate.

- 12. SSL Server Certificate.** Select the “Sign SSL certificate with my CA signing certificate” option. This option enables the wizard to generate an SSL Server Certificate signed with the local CA signing certificate, the root Certificate Manager’s CA signing certificate you just created.

Click Next to continue.

- 13. Key-Pair Information for SSL Server Certificate.** Select the token 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. Also specify the key type and length.

Click Next to continue.

- 14. Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature. The choices are: SHA-1, MD2, or MD5.

Click Next to continue.

- 15. Subject Name for SSL Server Certificate.** Type the values for the subject DN components; these values identify the root CA’s SSL server certificate. The CN must be the fully-qualified host name of the machine on which you’re installing the Certificate Manager.

Click Next to continue.

- 16. Validity Period for SSL Server Certificate.** Select the validity period for the SSL server certificate. The validity period determines how soon you will have to renew the certificate.

Click Next to continue.

- 17. Certificate Extensions for SSL Server Certificate.** Select the required extensions. 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 (see Step 10).

Click Next to continue.

- 18. 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.

19. Create Single Signon Password. Type the single signon password.

The single signon password simplifies the way you subsequently sign on to Certificate Management System by storing the passwords for the internal database, tokens, publishing directory, and so on. Each time you log on, you're only required to enter this single password. (For details, see "Required Start-up Information" on page 322.)

Click Next to continue.

20. Configuration Status. This screen should indicate that your configuration has been successful.

Click Done to exit the Installation Wizard.

21. Proceed to the next step, "Stage 3. Enrolling for Administrator/Agent Certificate" on page 277, to create the first *agent* user for the Certificate Manager.

Installing the Certificate Manager as a Subordinate CA

To install the Certificate Manager as a subordinate CA:

- 1. Subsystems.** Select Certificate Manager. If you want the Certificate Manager to issue certificates for wireless applications, select the "In addition to X.509 v3 certificates, do you want the Certificate Manager to support issuance of Wireless Transport Layer Support (wTLS)-compliant certificates" option. Otherwise, leave the option unchecked. (If you select the option, the end-entity interface will include two forms for requesting certificates for wireless applications and an option for downloading the wireless CA certificate.)

Click Next to continue.

- 2. Remote Data Recovery Manager.** Select the appropriate options:

- If you don't want to connect the Certificate Manager to a remote Data Recovery Manager, select No.
- If you have already installed a remote Data Recovery Manager that you want the Certificate Manager to use for archiving end users' encryption private keys, select Yes. Then, enter the remote Data Recovery Manager's host name and agent SSL port number in the associated fields.

Click Next to continue.

- 3. Network Configuration.** Type the port numbers for the ports to be used by the CMS instance. If you want to enable the non-SSL end-entity port, be sure to check the “Enable” checkbox.

Click Next to continue.

- 4. CA’s serial number range.** Specify range for the serial numbers. In the “Starting serial number” field, type the lowest serial number the CA should assign to a certificate. If you only use one CA server, you can leave the “Ending serial number” field blank to indicate no upper limit. If you plan to clone the CA to distribute load, you must specify an upper limit. (For cloned CAs, you must make sure that the range of serial numbers does not overlap with any other CA server.)

Click Next to continue.

- 5. CA Signing Certificate.** Select the “Create subordinate CA certificate request” option.

Click Next to continue.

- 6. Key-Pair Information for Certificate Manager CA signing certificate.** Select the token 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. Also specify the key type and length.

Click Next to continue.

- 7. Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature. The choices are: MD2, MD5, or SHA-1.

Click Next to continue.

- 8. Subject Name for Certificate Manager CA Signing Certificate.** Type values for the subject DN components; these values identify the subordinate CA signing certificate.

Click Next to continue.

- 9. Validity Period for Certificate Manager CA Signing Certificate.** Select the validity period for the subordinate CA signing certificate. The default validity is two years. The validity period determines how soon you will have to renew the certificate, which can be a complex procedure.

Click Next to continue.

- 10. Certificate Extensions for Certificate Manager CA Signing Certificate.** Select the required extensions. 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.

Certificate Management System provides command-line tools for generating extensions to include in CA and other certificate requests. For details about these tools, check this directory: `<server_root>/bin/cert/tools`

Note that 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 in the `tools` directory. For details about using the `ExtJoiner` program, see Chapter 5, “Extension Joiner Tool” of *CMS Command-Line Tools Guide*.

Click Next to continue.

- 11. Certificate Manager CA Signing Certificate Creation.** This is an informational screen that tells you that the wizard has all the information required to generate the key pair and certificate request. In the previous screen, if you chose to include the Subject Key Identifier extension in the certificate, you’ll be given the choice to select the format for the certificate request. Otherwise, the request format will be PKCS #10.
 - If you want the wizard to generate the certificate request in PKCS #10 format, select the “Generate PKCS10 request” option.
 - If you want the wizard to generate the certificate request in CMC format, select the “Generate CMC full enrollment request” option.

Click Next to generate the request. The wizard creates a certificate request that you must submit to another CA.

- 12. Submission of Request.** Select whether you want to submit the request manually or send the request to a remote Certificate Manager automatically.

To automatically submit the request to a remote Certificate Manager (or for automatic enrollment), follow these steps:

- a. Select the “Send the request to a remote CMS now” option.
- b. Enter the host name and end-entity port number of the remote Certificate Manager, and select whether this end-entity port is SSL enabled.

- c. Click Next to submit the request.

The Certificate Request Result screen appears, confirming that the request has been submitted. Note the request ID provided in the response message. (You can use it later to retrieve the certificate, once it has been issued, from the end-entity port.)

Note that the request you submitted gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you should wait for the other agent to approve your request and issue the certificate.

- d. Open a web browser window.
- e. Enter the URL for the remote Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- f. Select List Requests, then click Show Pending Requests and click Find. The pending request list is displayed.
- g. Locate your request, click Details to see it, and make any changes. Then, scroll down to the bottom of the form and click Do It.
- h. After the certificate is generated, click Show Certificate.
- i. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard screen next. So, once you've copied the certificate, go back to the wizard screen (Step 13).

To submit your certificate request manually to a remote Certificate Manager, follow these steps:

- a. Open a web browser window.
- b. Go to the end-entity URL for the remote 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.

- c. In the left-hand frame of the Enrollment tab, choose the form appropriate for the request type:

If the request is in the PKCS #10 format, under Server, click Certificate Manager. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information.

If the request is in the CMC format, click CMC Enrollment. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information. Be sure to select *CA Signing Certificate* as the certificate type.

- d. Click Submit.
- e. The request gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you'll have to wait till the remote Certificate Manager's agent approves your request.
- f. In the web browser window, enter the URL for the remote Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- g. Select List Requests, then click Show Pending Requests and click Find.
- h. In the pending request list, locate your request, click Details to see the request, and make any changes. Then, scroll down to the bottom of the form, and click Do It.
- i. After the certificate is generated, click Show Certificate.
- j. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 13).

To submit your certificate request manually to a third-party CA, follow these steps:

- a. Make sure that the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) is highlighted, and click the Copy to Clipboard button.

This action copies the certificate request 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.

- b. Submit your certificate request to a third-party CA, following the instructions provided by that CA.

Click Next when you are ready to proceed.

13. CA Signing Certificate Installation. Depending on whether you have the certificate ready for pasting into the Installation Wizard screen, click Yes or No.

- o If you have submitted your request to a third-party CA or to a remote Certificate Manager for which you do not have agent privileges, 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. The default selection is No.
- o Select Yes, only if you have the certificate ready in its base-64 encoded format.

Click Next to continue.

- o If you selected No, you will be presented with the "SSL Server Certificate" screen (Step 17).
- o If you selected Yes, the "Location of Certificate" screen appears (Step 14).

14. Location of Certificate. Specify the location of the certificate. You can use any of these options:

- o If you copied the encoded certificate to a file, select the "The certificate is located in this file" option and then type the file path, including the filename, in the text field.
- o If you copied the certificate to the clipboard, select the "The certificate is located in the text area below" option and then paste in a base-64 encoded certificate (including the header and footer) in the text area provided.

- If you noted the request ID of your request and know the host name and end-entity port number of the remote Certificate Manager that issued the certificate, select the “The certificate is at the CMS server where the request was sent” option and then specify the required details.

Click Next to continue.

- 15. Certificate Details.** This is an informational screen that shows the certificate so you can inspect its contents. Notice the nickname assigned to the certificate and verify that you’re installing the correct certificate.

Click Next to continue.

- 16. Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. If the CA that issued the certificate is a Certificate Manager, follow these steps:

- a. Go to the end-entity URL for the Certificate Manager that issued the subordinate CA’s signing certificate.
- b. Select the Retrieval tab, and then choose Import CA Certificate Chain.
- c. Select the “Display the CA certificate chain in PKCS#7 for importing into a server” option, and then click Submit.
- d. Copy the certificate chain to the clipboard.
- e. Return to the Installation Wizard.
- f. Paste the certificate chain into the text box.

Click Next to continue.

- 17. SSL Server Certificate.** Select the appropriate option:

- If you want to get the SSL server certificate signed by the subordinate CA itself, select the “Sign SSL certificate with my CA signing certificate” option.
- If you want to submit the SSL server certificate request to another CA, for example to the CA that signed the subordinate CA’s signing certificate, select the “Create request for submission to another CA” option.

Click Next to continue.

18. Key-Pair Information for SSL Server Certificate. Select the token 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. Also specify the key type and length.

Click Next to continue.

19. Message Digest Algorithm. Select the algorithm to use for computing the certificate signature. The choices are: SHA-1, MD2, or MD5.

Click Next to continue.

20. Subject Name for SSL Server Certificate. Type the values for the subject DN components; these values identify the subordinate CA's SSL server certificate. The CN must be the fully-qualified host name of the machine on which you're installing the Certificate Manager.

Click Next to continue.

21. Certificate Extensions for SSL Server Certificate. Select the required extensions. 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 details, see Step 10 of this section.)

Click Next to continue.

22. SSL Server Certificate Request Creation. This is an informational screen that tells you that the wizard has all the information required to generate the key pair and certificate request. In the previous screens, if you chose to generate a certificate request and include the Subject Key Identifier extension in the certificate, you'll be given the choice to select the format for the certificate request. Otherwise, the request format will be PKCS #10.

- If you want the wizard to generate the certificate request in PKCS #10 format, select the "Generate PKCS10 request" option.
- If you want the wizard to generate the certificate request in CMC format, select the "Generate CMC full enrollment request" option.

Click Next to generate the certificate or the request:

- If you chose to get the certificate signed by the subordinate CA itself, the wizard generates the SSL server certificate. You'll be presented with the "Create Single Signon Password" screen (Step 28).
- If you chose to generate a request for submission to another CA, the wizard generates an SSL server certificate request that you must submit to another CA. You'll be presented with the "Submission of Request" screen (Step 23).

- 23. Submission of Request.** Select whether you want to submit the request manually or send the request automatically to a remote Certificate Manager.

To automatically submit the request to a remote Certificate Manager (or for automatic enrollment), follow these steps:

- a. Select the “Send the request to a remote CMS now” option.
- b. Enter the host name and end-entity port number of the remote Certificate Manager, and specify whether the end-entity port is SSL enabled.
- c. Click Next to submit the request.

The Certificate Request Result screen appears, confirming that the request has been submitted. Note the request ID provided in the response message. (You can use it later to retrieve the certificate, once it has been issued, from the end-entity port.)

Note that the request gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager’s agent. If you’ve permission to access that Certificate Manager’s Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you should wait for the remote Certificate Manager’s agent to approve your request and issue the certificate.

- d. Open a web browser window.
- e. Enter the URL for the remote Certificate Manager’s Agent Services page. (You must use the same computer where you got your agent certificate.)
- f. Select List Requests, click Show Pending Requests, and then click Find.
- g. In the pending request list, locate your request, click Details to see the request, and make any changes. Then, scroll down to the bottom of the form, and click Do It.
- h. After the certificate is generated, click Show Certificate.
- i. 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 or to a text file.

Be sure to not make any changes to the certificate. You’re required to paste the encoded certificate into the Installation Wizard next. So, once you’ve copied the certificate, go back to the wizard screen (Step 24).

To submit your certificate request manually to a remote Certificate Manager, follow these steps:

- a. Open a web browser window.
- b. Go to the end-entity URL for the remote Certificate Manager that will issue the subordinate CA's SSL server 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.

- c. In the left-hand frame of the Enrollment tab, choose the form appropriate for the request type:

If the request is in the PKCS #10 format, under Server, click SSL Server. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information.

If the request is in the CMC format, click CMC Enrollment. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information. Be sure to select `Server SSL Certificate` as the certificate type.

- d. Click Submit.
- e. The request gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you'll have to wait till the remote Certificate Manager's agent approves your request.
- f. In the web browser window, enter the URL for the remote Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- g. Select List Requests, click Show Pending Requests, and click Find.
- h. In the pending request list, locate your request, click Details to see it, and make any changes. Then, scroll down to the bottom of the form and click Do It.
- i. After the certificate is generated, click Show Certificate.

- j. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 24 below).

To submit your certificate request manually to a third-party CA, follow these steps:

- a. Make sure that the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) is highlighted, and click the Copy to Clipboard button.

This action copies the certificate request 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 SSL server certificate.

- b. Submit your certificate request to a third-party CA, following the instructions provided by that CA.

Click Next when you are ready to proceed to the next screen.

24. SSL Server Certificate Installation. Depending on whether you have the certificate ready for pasting into the Installation Wizard screen, click Yes or No.

- o If you have submitted your request to a third-party CA or to a remote Certificate Manager for which you do not have agent privileges, 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. The default is No. If you selected No, you will be presented with the "Create Single Signon Password" screen.
- o Select Yes, only if you have the certificate ready in its base-64 encoded format.

Click Next to continue.

- o If you selected Yes, the "Location of Certificate" screen appears (Step 25).
- o If you selected No, you will be presented with the "Create Single Signon Password" screen (Step 28).

- 25. Location of Certificate.** Specify the location of the certificate. You can use any of these options:
- If you copied the encoded certificate to a file, select the “The certificate is located in this file” option and then type the file path, including the filename, in the text field.
 - If you copied the certificate to the clipboard, select the “The certificate is located in the text area below” option and then paste in a base-64 encoded certificate (including the header and footer) in the text area provided.
 - If you noted the request ID of your request and know the host name and end-entity port number of the remote Certificate Manager that issued the certificate, select the “The certificate is at the CMS server where the request was sent” option and then specify the required details.

Click Next to continue.

- 26. Certificate Details.** This is an informational screen that displays the certificate so you can inspect its contents. Notice the nickname assigned to the certificate and verify that you’re installing the correct certificate.

Click Next to continue.

- 27. Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. If the CA that issued the certificate is a Certificate Manager, follow these steps:
- a. Go to the end-entity URL for the remote Certificate Manager that issued the SSL server certificate.
 - b. Select the Retrieval tab, and then in the left-hand frame, click Import CA Certificate Chain.
 - c. Select the “Display the CA certificate chain in PKCS#7 for importing into a server” option, and then click Submit.
 - d. In the resulting form, locate the CA certificate chain, in its base-64 encoded format, to the clipboard.
 - e. Return to the Installation Wizard.
 - f. Paste the certificate chain into the text box.

Click Next to continue.

28. **Create Single Signon Password.** Type the single signon password. The single signon password simplifies the way you subsequently sign on to Certificate Management System by storing the passwords for the internal database, tokens, publishing directory, and so on. Each time you log on, you're only required to enter this single password. (For details, see "Required Start-up Information" on page 322.)

Click Next to continue.

29. **Configuration Status.** This screen should indicate that your configuration has been successful.

Click Done to exit the Installation Wizard.

30. Proceed to the next step, "Stage 3. Enrolling for Administrator/Agent Certificate" on page 277, to create the first *agent* user for the Certificate Manager.

Installing a Standalone Registration Manager

To install a standalone Registration Manager:

1. **Subsystems.** Select Registration Manager.

Click Next to continue.

2. **Remote Certificate Manager.** Type the host name and agent port number of the remote Certificate Manager to which you want to connect this Registration Manager.

Click Next to continue.

3. **Remote Data Recovery Manager.** Select the appropriate options:

- Select No, if you don't want to connect the Registration Manager to a remote Data Recovery Manager.
- If you have already installed a remote Data Recovery Manager that you want the Registration Manager to use for archiving end users' encryption private keys, select Yes. Then, enter the remote Data Recovery Manager's host name and agent port number in the associated fields.

Click Next to continue.

4. **Network Configuration.** Type the numbers for the ports to be used by the CMS instance. If you want to enable the non-SSL end-entity port, be sure to check the “Enable” checkbox.

Click Next to continue.

5. **Key-Pair Information for Registration Manager Signing Certificate.** Select the token 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. Also specify the key type and length.

Click Next to continue.

6. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature. The choices are: SHA-1, MD2, or MD5.

Click Next to continue.

7. **Subject Name for Registration Manager Signing Certificate.** Type the values for the subject DN components; these values identify the Registration Manager’s signing certificate.

Click Next to continue.

8. **Certificate Extensions for Registration Manager Signing Certificate.** Select the required extensions. 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.

Certificate Management System provides command-line tools for generating extensions to include in CA and other certificate requests. For details about these tools, check this directory: `<server_root>/bin/cert/tools`

Note that 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 in the `tools` directory. For details on using the `ExtJoiner` program, see Chapter 5, “Extension Joiner Tool” of *CMS Command-Line Tools Guide*.

Click Next to continue.

9. **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. In the previous screen, if you chose to include the Subject Key Identifier extension in the certificate, you’ll be given the choice to select the format for the certificate request. Otherwise, the request format will be PKCS #10.

- If you want the wizard to generate the certificate request in PKCS #10 format, select the “Generate PKCS10 request” option.
- If you want the wizard to generate the certificate request in CMC format, select the “Generate CMC full enrollment request” option. (This option is available only if you selected to add the Subject Key Identifier extension to the certificate in the previous.)

Click Next. The wizard creates a certificate request that you must submit to a CA, which could be a remote Certificate Manager or a third-party CA.

- 10. Submission of Request.** Select whether you want to submit the request manually or send the request automatically to a remote Certificate Manager.

To automatically submit the request to a remote Certificate Manager (or for automatic enrollment), follow these steps:

- a. Select the “Send the request to a remote CMS now” option.
- b. Enter the host name and end-entity port number of the remote Certificate Manager, and specify whether the end-entity port is SSL enabled.
- c. Click Next to submit the request.

The Certificate Request Result screen appears, confirming that the request has been submitted. Note the request ID provided in the response message. (You can use it later to retrieve the certificate, once it has been issued, from the end-entity port.)

Note that your request gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager’s agent. If you’ve permission to access that Certificate Manager’s Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you should wait for the remote Certificate Manager’s agent to approve your request.

- d. Open a web browser window.
- e. Enter the URL for the remote Certificate Manager’s Agent Services page. (You must use the same computer where you got your agent certificate.)
- f. Select List Requests, click Show Pending Requests, and click Find.
- g. In the pending request list, locate your request, click Details to see the request, and make any changes. Then, scroll down to the bottom of the form and click Do It.
- h. After the certificate is generated, click Show Certificate.

- i. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 11).

To submit your certificate request manually to a remote Certificate Manager, follow these steps:

- a. Open a web browser window.
- b. Go to the end-entity URL of the remote Certificate Manager that will issue the Registration Manager'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.

- c. In the left-hand frame of the Enrollment tab, choose the form appropriate for the request type:

If the request is in the PKCS #10 format, under Server, click Registration Manager. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information.

If the request is in the CMC format, click CMC Enrollment. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information. Be sure to select RA Signing Certificate as the certificate type.

- d. Click Submit.
- e. The request gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you'll have to wait till the remote Certificate Manager's agent approves your request.
- f. In the web browser window, enter the URL for the remote Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- g. Select List Requests, click Show Pending Requests, and click Find.

- h. In the pending request list, locate your request, click Details to see it. After checking the certificate request and making required changes, scroll down to the last section, labeled Privileges.
- i. 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.)
- j. 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 iPlanet Console, such as `RMEng`.
- k. Scroll to the bottom and click Do It.
- l. After the certificate is generated, click Show Certificate.
- m. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 11).

To submit your certificate request manually to a third-party CA, follow these steps:

- a. Make sure that the certificate request (including `-----BEGIN NEW CERTIFICATE REQUEST -----` and `-----END NEW CERTIFICATE REQUEST -----`) is highlighted, and click the Copy to Clipboard button.

This action copies the certificate request 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 Registration Manager's signing certificate.

- b. Submit your certificate request to a third-party CA, following the instructions provided by that CA.

Click Next when you are ready to proceed.

- 11. Registration Manager Signing Certificate Installation.** Depending on whether you have the certificate ready for pasting into the Installation Wizard screen, click Yes or No.

- If you have submitted your request to a third-party CA or to a remote Certificate Manager for which you do not have agent privileges, 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. The default selection is No.
- Select Yes, only if you have the certificate ready in its base-64 encoded format.

Click Next to continue.

- If you selected Yes, the “Location of Certificate” screen appears (Step 12).
- If you selected No, you will be presented with the “Key-Pair Information for SSL Server Certificate” screen (Step 15).

12. Location of Certificate. Specify the location of the certificate. You can use any of these options:

- If you copied the encoded certificate to a file, select the “The certificate is located in this file” option and then type the file path, including the filename, in the text field.
- If you copied the certificate to the clipboard, select the “The certificate is located in the text area below” option and then paste in a base-64 encoded certificate (including the header and footer) in the text area provided.
- If you noted the request ID of your request and know the host name and end-entity port number of the remote Certificate Manager that issued the certificate, select the “The certificate is at the CMS server where the request was sent” option and then specify the required details.

Click Next to continue.

13. Certificate Details. This is an informational screen that displays the certificate so you can inspect its contents. Notice the nickname assigned to the certificate and verify that you’re installing the correct certificate.

Click Next to continue.

14. Import Certificate Chain. This screen appears only if you need to import the CA certificate chain. If the CA that issued the certificate is a Certificate Manager, follow these steps:

- a. Go to the end-entity URL for the remote Certificate Manager that issued the Registration Manager’s signing certificate.
- b. Select the Retrieval tab, and in the left-hand frame, click Import CA Certificate Chain.

- c. In the resulting form, select the “Display the CA certificate chain in PKCS#7 for importing into a server” option, and then click Submit.
- d. In the resulting page, locate the CA certificate chain in its base-64 encoded format, and copy the certificate chain to the clipboard.
- e. Return to the Installation Wizard.
- f. Paste the certificate chain into the text box.

Click Next to continue.

- 15. Key-Pair Information for SSL Server Certificate.** Select the token 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. Also specify the key type and length.

Click Next to continue.

- 16. Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature. The choices are: SHA-1, MD2, or MD5.

Click Next to continue.

- 17. Subject Name for SSL Server Certificate.** Type the values for the subject DN components; these values identify the Registration Manager’s SSL server certificate. The CN must be the fully-qualified host name of the machine on which you’re installing the Registration Manager.

Click Next to continue.

- 18. Certificate Extensions for SSL Server Certificate.** Select the required extensions. 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 details, see Step 8 of this section.)

Click Next to continue.

- 19. SSL Server Certificate Request Creation.** This is an informational screen that tells you that the wizard has all the information required to generate the key pair and certificate request. In the previous screen, if you chose to include the Subject Key Identifier extension in the certificate, you’ll be given the choice to select the format for the certificate request. Otherwise, the request format will be PKCS #10.

- o If you want the wizard to generate the certificate request in PKCS #10 format, select the “Generate PKCS10 request” option.

- o If you want the wizard to generate the certificate request in CMC format, select the “Generate CMC full enrollment request” option.

Click Next. The wizard creates the certificate request that you must submit to another CA.

- 20. Submission of Request.** Select whether you want to submit the request manually or send the request automatically to a remote Certificate Manager.

To automatically submit the request to a remote Certificate Manager (or for automatic enrollment), follow these steps:

- a. Select the “Send the request to a remote CMS now” option.
- b. Enter the host name and end-entity port number of the remote Certificate Manager, and select whether the end-entity port is SSL enabled.
- c. Click Next to submit the request.

The Certificate Request Result screen appears, confirming that the request has been submitted. Note the request ID provided in the response message. (You can use it later to retrieve the certificate, once it has been issued, from the end-entity port.)

Note that your request gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager’s agent. If you’ve permission to access that Certificate Manager’s Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you should wait for the remote Certificate Manager’s agent to approve your request.

- d. In the web browser window, enter the URL for the remote Certificate Manager’s Agent Services page. (You must use the same computer where you got your agent certificate.)
- e. Select List Requests, click Show Pending Requests, and click Find.
- f. In the pending request list, locate your request, click Details to see it, and make any changes. Then, scroll down to the bottom of the form, and click Do It.
- g. After the certificate is generated, click Show Certificate.

- h.** 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 21).

To submit your certificate request manually to a remote Certificate Manager, follow these steps:

- a.** Open a web browser window.
- b.** Go to the end-entity URL for the remote Certificate Manager that will issue the SSL server 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.

- c.** In the left-hand frame of the Enrollment tab, choose the form appropriate for the request type:

If the request is in the PKCS #10 format, under Server, click **SSL Server**. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information.

If the request is in the CMC format, click **CMC Enrollment**. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information. Be sure to select **Server SSL Certificate** as the certificate type.

- d.** Click **Submit**.
- e.** The request gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you'll have to wait till the remote Certificate Manager's agent approves your request.
- f.** In the web browser window, enter the URL for the remote Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- g.** Select **List Requests**, click **Show Pending Requests**, and click **Find**. The pending request list is displayed.

- h. Locate your request, click Details to see it, and make any changes. Then, scroll down to the bottom of the form and click Do It.
- i. After the certificate is generated, click Show Certificate.
- j. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 21).

To submit your certificate request manually to a third-party CA, follow these steps:

- a. Make sure that the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) is highlighted, and click the Copy to Clipboard button.

This action copies the certificate request 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 Registration Manager's SSL server certificate.

- b. Submit your certificate request to a third-party CA, following the instructions provided by that CA.

Click Next when you are ready to proceed to the next screen.

21. SSL Server Certificate Installation. Depending on whether you have the certificate ready for pasting into the Installation Wizard screen, click Yes or No.

- o If you have submitted your request to a third-party CA or to a remote Certificate Manager for which you do not have agent privileges, 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. The default is No.
- o Select Yes, only if you have the certificate ready in its base-64 encoded format.

Click Next to continue.

- o If you selected Yes, the "Location of Certificate" screen appears (Step 22).

- If you selected No, you will be presented with the “Create Single Signon Password” screen (Step 25).

22. Location of Certificate. Specify the location of the certificate. You can use any of these options:

- If you copied the encoded certificate to a file, select the “The certificate is located in this file” option and then type the file path, including the filename, in the text field.
- If you copied the certificate to the clipboard, select the “The certificate is located in the text area below” option and then paste in a base-64 encoded certificate (including the header and footer) in the text area provided.
- If you noted the request ID of your request and know the host name and end-entity port number of the Certificate Manager that issued the certificate, select the “The certificate is at the CMS server where the request was sent” option and then specify the required details.

Click Next to continue.

23. Certificate Details. This is an informational screen that displays the certificate so you can inspect its contents. Notice the nickname assigned to the certificate and verify that you’re installing the correct certificate.

Click Next to continue.

24. Import Certificate Chain. This screen appears only if you need to import the CA certificate chain again; for example, if you requested the SSL certificate from a different CA than the one from which you requested the signing certificate.

Follow these steps to import the remote Certificate Manager’s CA chain:

- a. Go to the web browser window.
- b. Enter the end-entity URL for the remote Certificate Manager that issued the SSL server certificate.
- c. Select the Retrieval tab, and in the left-hand frame, click Import CA Certificate Chain.
- d. Select the “Display the CA certificate chain in PKCS#7 for importing into a server” option, and then click Submit.
- e. In the resulting page, locate the CA certificate chain in its base-64 encoded format, and copy it to the clipboard.
- f. Return to the Installation Wizard.

- g. Paste the CA certificate chain into the text box.

Click Next to continue.

- 25. **Create Single Signon Password.** Type the single signon password. The single signon password simplifies the way you subsequently sign on to Certificate Management System by storing the passwords for the internal database, tokens, and so on. Each time you log on, you're only required to enter this single password. (For details, see "Required Start-up Information" on page 322.)

Click Next to continue.

- 26. **Configuration Status.** This screen should indicate that your configuration has been successful.

Click Done to exit the Installation Wizard.

- 27. Proceed to the next step, "Stage 3. Enrolling for Administrator/Agent Certificate" on page 277, to create the first *agent* user for the Registration Manager.

Installing a Standalone Data Recovery Manager

To install a standalone Data Recovery Manager:

- 1. **Subsystems.** Select Data Recovery Manager.

Click Next to continue.

- 2. **Network Configuration.** Type the numbers for the ports to be used by the CMS instance. If you want to enable the non-SSL end-entity port, be sure to check the "Enable" checkbox.

Click Next to continue.

- 3. **Key-Pair Information for Data Recovery Manager Transport Certificate.** Select the token 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. Also specify the key type and length.

Click Next to continue.

- 4. **Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature. The choices are: SHA-1, MD2, or MD5.

Click Next to continue.

- 5. Subject Name for Data Recovery Manager Transport Certificate.** Type the values for the subject DN components; these values identify the transport certificate.

Click Next to continue.

- 6. Certificate Extensions for Data Recovery Manager Transport Certificate.** Select the required extensions. 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.

Certificate Management System provides command-line tools for generating extensions to include in CA and other certificate requests. For details about these tools, check this directory: `<server_root>/bin/cert/tools`

Note that 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 in the `tools` directory. For details on using the `ExtJoiner` program, see Chapter 5, “Extension Joiner Tool” of *CMS Command-Line Tools Guide*.

Click Next to continue.

- 7. 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. In the previous screen, if you chose to include the Subject Key Identifier extension in the certificate, you’ll be given the choice to select the format for the certificate request. Otherwise, the request format will be PKCS #10.

- If you want the wizard to generate the certificate request in PKCS #10 format, select the “Generate PKCS10 request” option.
- If you want the wizard to generate the certificate request in CMC format, select the “Generate CMC full enrollment request” option.

Click Next. The wizard generates the certificate request that you must submit to a CA, which could be a remote Certificate Manager or a third-party CA.

- 8. Submission of Request.** Specify whether you want to submit the request automatically or manually.

To automatically submit the request to a remote Certificate Manager (or for automatic enrollment), follow these steps:

- Select the “Send the request to a remote CMS now” option.
- Enter the host name and end-entity port number, and specify whether the end-entity port is SSL enabled.

- c. Click Next to submit the request.

The Certificate Request Result screen appears, confirming that the request has been submitted. Note the request ID provided in the response message. (You can use it later to retrieve the certificate, once it has been issued, from the end-entity port.)

Note that your request gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you should wait for the remote Certificate Manager's agent to approve your request.

- d. Open a web browser window.
- e. Enter the URL for the remote Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- f. Select List Requests, click Show Pending Requests, and click Find.
- g. In the pending request list, locate your request, click Details to see the request, and make any changes. Then, scroll down to the bottom of the form, and click Do It.
- h. After the certificate is generated, click Show Certificate.
- i. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 9).

To submit your certificate request manually to a remote Certificate Manager, follow these steps:

- a. Open a web browser window.
- b. Go to the end-entity URL for the remote Certificate Manager that will issue the Data Recovery Manager's transport 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.

- c. In the left-hand frame of the Enrollment tab, choose the form appropriate for the request type:

If the request is in the PKCS #10 format, under Server, click SSL Server. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information.

If the request is in the CMC format, click CMC Enrollment. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information. Be sure to select `Server SSL Certificate` as the certificate type.

- d. Click Submit.
- e. The request gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate.
- f. In the web browser window, enter the URL for the remote Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- g. Select List Requests, click Show Pending Requests, and click Find.
- h. In the pending request list, locate your request, then click Details to see the request. After checking the rest of the certificate request, scroll down to the end of the form and click Do It.
- i. After the certificate is generated, click Show Certificate.
- j. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 9).

To submit the transport certificate request manually to a third-party CA, follow these steps:

- a. Make sure that the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) is highlighted, and click the Copy to Clipboard button.

This action copies the certificate request 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 Data Recovery Manager's transport certificate.

- b. Submit your certificate request to a third-party CA, following the instructions provided by that CA.

Click Next when you are ready to proceed.

- 9. **Data Recovery Manager Transport Certificate Installation.** Depending on whether you have the certificate ready for pasting into the Installation Wizard screen, click Yes or No.

- o If you have submitted your request to a third-party CA or to a remote Certificate Manager for which you do not have agent privileges, 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. The default is No.
- o Select Yes, only if you have the certificate ready in its base-64 encoded format.

Click Next to continue.

- o If you selected Yes, the "Location of Certificate" screen appears (Step 10).
- o If you selected No, you will be presented with the "Storage Key Creation for Data Recovery Manager" screen (Step 13).

- 10. **Location of Certificate.** Specify the location of the certificate. You can use any of these options:

- o If you copied the encoded certificate to a file, select the "The certificate is located in this file" option and then type the file path, including the filename, in the text field.
- o If you copied the certificate to the clipboard, select the "The certificate is located in the text area below" option and then paste in a base-64 encoded certificate (including the header and footer) in the text area provided.

- If you noted the request ID of your request and know the host name and end-entity port number of the remote Certificate Manager that issued the certificate, select the “The certificate is at the CMS server where the request was sent” option and then specify the required details.

Click Next to continue.

11. **Certificate Details.** This informational screen displays the certificate so you can inspect its contents. Notice the nickname assigned to the certificate and verify that you’re installing the correct certificate.

Click Next to continue.

12. **Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. Follow these steps to import the CA chain of the remote Certificate Manager:

- a. Go to the web browser window.
- b. Enter the end-entity URL for the remote Certificate Manager that issued the transport certificate.
- c. Select the Retrieval tab, and then in the left-hand frame, click Import CA Certificate Chain.
- d. In the resulting form, select the “Display the CA certificate chain in PKCS#7 for importing into a server” option, and click Submit.
- e. In the resulting page, locate the CA certificate chain in its base-64 encoded format, and copy it to the clipboard.
- f. Return to the Installation Wizard.
- g. Paste the CA certificate chain into the text box.

Click Next to continue.

The screens that follow let you configure the storage key and recovery schemes for the Data Recovery Manager.

13. **Storage Key Creation for Data Recovery Manager.** Select the length you have decided on for your storage key.

Click Next to continue.

14. **Data Recovery Key Scheme - 1.** Type the both the required number of recovery agents and the total number of recovery agents.

Click Next to continue.

- 15. Data Recovery Key Scheme - 2.** The number of table rows correspond to the total number of agents you specified in the previous screen. Type the user ID and password for each agent in the table.

Click Next to continue. The screens that follow let you request an SSL server certificate for the Data Recovery Manager.

- 16. Key-Pair Information for SSL Server Certificate.** Select the token 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. Also specify the key type and length.

Click Next to continue.

- 17. Message Digest Algorithm.** Select the algorithm to use for computing the certificate signature. The choices are: SHA-1, MD2, or MD5.

Click Next to continue.

- 18. Subject Name for SSL Server Certificate.** Type the values for the subject DN components; these values the Data Recovery Manager's SSL server certificate. The CN must be the fully-qualified host name of the machine on which you're installing the Data Recovery Manager.

Click Next to continue.

- 19. Certificate Extensions for SSL Server Certificate.** Select the required extensions. 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 details, see Step 6 of this section.)

Click Next to continue.

- 20. SSL Server Certificate Request Creation.** This is an informational screen that tells you that the wizard has all the information required to generate the key pair and certificate request. In the previous screen, if you chose to include the Subject Key Identifier extension in the certificate, you'll be given the choice to select the format for the certificate request. Otherwise, the request format will be PKCS #10.

- If you want the wizard to generate the certificate request in PKCS #10 format, select the "Generate PKCS10 request" option.
- If you want the wizard to generate the certificate request in CMC format, select the "Generate CMC full enrollment request" option.

Click Next. The wizard generates a certificate request that you must submit to a CA.

21. Submission of Request. Select whether you want to submit the request manually or send the request automatically to a remote Certificate Manager.

To automatically submit the request to a remote Certificate Manager (or for automatic enrollment), follow these steps:

- a. Select the “Send the request to a remote CMS now” option.
- b. Enter the host name and end-entity port number of the remote Certificate Manager, and specify whether the end-entity port is SSL enabled.
- c. Click Next to submit the request.

The Certificate Request Result screen appears, confirming that the request has been submitted. Note the request ID provided in the response message. (You can use it later to retrieve the certificate, once it has been issued, from the end-entity port.)

Note that your request gets added to the agent queue of the remote Certificate Manager for approval by that Certificate Manager’s agent. If you’ve permission to access that Certificate Manager’s Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you should wait for the remote Certificate Manager’s agent to approve your request and issue the certificate.

- d. In the web browser window, enter the URL for the remote Certificate Manager’s Agent Services page. (You must use the same computer where you got your agent certificate.)
- e. Select List Requests, click Show Pending Requests, and click Find.
- f. In the pending request list, locate your request, click Details to see the request, and make any changes. Then, scroll down to the bottom of the form and click Do It.
- g. After the certificate is generated, click Show Certificate.
- h. 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 or to a text file.

Be sure to not make any changes to the certificate. You’re required to paste the encoded certificate into the Installation Wizard next. So, once you’ve copied the certificate, go back to the wizard screen (Step 22).

To submit your certificate request manually to a Certificate Manager, follow these steps:

- a. Open a web browser window.
- b. Go to the end-entity URL for the Certificate Manager that will issue the SSL server 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.

- c. In the left-hand frame of the Enrollment tab, choose the form appropriate for the request type:

If the request is in the PKCS #10 format, under Server, click SSL Server. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information.

If the request is in the CMC format, click CMC Enrollment. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information. Be sure to select `Server SSL Certificate` as the certificate type.

- d. Click Submit.

The request gets added to the agent queue of that Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate.

- e. In the web browser window, enter the URL for the Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- f. Select List Requests, then click Show Pending Requests and click Find. The pending request list is displayed.
- g. Locate your request, click Details to see it, and make any changes. Then, scroll down to the bottom of the form and click Do It.
- h. After the certificate is generated, click Show Certificate.

- i. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 22).

To submit your certificate request manually to a third-party CA, follow these steps:

- a. Make sure that the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) is highlighted, and click the Copy to Clipboard button.

This action copies the certificate request 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.

- b. Submit your certificate request to a third-party CA, following the instructions provided by that CA.

Click Next when you are ready to proceed to the next screen.

22. SSL Server Certificate Installation. Depending on whether you have the certificate ready for pasting into the Installation Wizard screen, click Yes or No.

- o If you have submitted your request to a third-party CA or to a remote Certificate Manager for which you do not have agent privileges, 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. The default is No.
- o Select Yes, only if you have the certificate ready in its base-64 encoded format.

Click Next to continue.

- o If you selected Yes, the "Location of Certificate" screen appears (Step 23).
- o If you selected No, you will be presented with the "Create Single Signon Password" screen (Step 26).

- 23. Location of Certificate.** Specify the location of the certificate. You can use any of these options:
- If you copied the encoded certificate to a file, select the “The certificate is located in this file” option and then type the file path, including the filename, in the text field.
 - If you copied the certificate to the clipboard, select the “The certificate is located in the text area below” option and then paste in a base-64 encoded certificate (including the header and footer) in the text area provided.
 - If you noted the request ID of your request and know the host name and end-entity port number of the remote Certificate Manager that issued the certificate, select the “The certificate is at the CMS server where the request was sent” option and then specify the required details.

Click Next to continue.

- 24. Certificate Details.** This is an informational screen that displays the certificate so you can inspect its contents. Notice the nickname assigned to the certificate and verify that you’re installing the correct certificate.

Click Next to continue.

- 25. Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain again. Follow these steps to import the CA chain of a remote Certificate Manager:
- a. Go to the web browser window.
 - b. Enter the end-entity URL for the remote Certificate Manager that issued the SSL server certificate.
 - c. Select the Retrieval tab, and then in the left-hand frame, select Import CA Certificate Chain.
 - d. In the resulting form, select the “Display the CA certificate chain in PKCS#7 for importing into a server” option, and click Submit.
 - e. In the resulting page, locate the CA certificate chain in its base-64 encoded format and copy it to the clipboard.
 - f. Return to the Installation Wizard.
 - g. Paste the CA certificate chain into the text box.

Click Next to continue.

- 26. Create Single Signon Password.** Type the single signon password. The single signon password simplifies the way you subsequently sign on to Certificate Management System by storing the passwords for the internal database, tokens, and so on. Each time you log on, you're only required to enter this single password. (For details, see "Required Start-up Information" on page 322.)

Click Next to continue.

- 27. Configuration Status.** This screen should indicate that your configuration has been successful.

Click Done to exit the Installation Wizard.

- 28.** Proceed to the next step, "Stage 3. Enrolling for Administrator/Agent Certificate" on page 277, to create the first agent for the Data Recovery Manager.

Installing an Online Certificate Status Manager

To install a standalone Online Certificate Status Manager:

- 1. Subsystems.** Select Online Certificate Status Manager.

Click Next to continue.

- 2. Network Configuration.** Type the numbers for the ports to be used by the CMS instance. Be sure to leave the "Enable" checkbox for the non-SSL end-entity port selected. The OCSP-compliant clients will use this port to communicate with the Online Certificate Status Manager.

Click Next to continue.

- 3. Key-Pair Information for Online Certificate Status Manager Signing Certificate.** Select the token to store the Online Certificate Status Manager signing certificate and key pair. If you have not previously initialized the token's password, you must do so in this screen. Also specify the key type and length.

Click Next to continue.

- 4. Subject Name for Online Certificate Status Manager Signing Certificate.** Type the values for the subject DN components; these values identify the Online Certificate Status Manager's signing certificate.

Click Next to continue.

5. Online Certificate Status 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.

6. Submission of Request. Select whether you want to submit the request manually or send the request to a remote CMS manager (Certificate Manager or Registration Manager) automatically. The wizard creates a certificate request that you must submit to a CA.

To automatically submit the request to a remote Certificate Manager (or for automatic enrollment), follow these steps:

- a. Select the “Send the request to a remote CMS now” option.
- b. Enter the host name (for example, `host.domain.com`) and end-entity port number of the Certificate Manager, then specify whether this end-entity port uses SSL.
- c. Click Next to submit the request.

The Certificate Request Result screen appears, confirming that the request has been submitted. Note the request ID provided in the response message. (You can use it later to retrieve the certificate, once the certificate has been issued, from the end-entity port.)

Note that your request gets added to the agent queue of the Certificate Manager for approval by that Certificate Manager’s agent. If you’ve permission to access that Certificate Manager’s Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you should wait for the other agent to approve the request you submitted and issue the certificate.

- d. Open a web browser window.
- e. Enter the URL for the Certificate Manager’s Agent Services page. (You must use the same computer where you got your agent certificate.)
- f. Select List Requests, then click Show Pending Requests and click Find. The pending request list is displayed.
- g. Locate your request, click Details to see it, and make any changes. Then, scroll down to the bottom of the form and click Do It.
- h. After the certificate is generated, click Show Certificate.

- i. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 7).

Also note that you might be required to paste the CA certificate chain in the Installation Wizard. So, keep the browser window open.

To submit your certificate request manually to a Certificate Manager, follow these steps:

- a. Open a web browser window.
- b. Go to the end-entity URL for the Certificate Manager that will issue the Online Certificate Status Manager's signing certificate.

For example, if you assigned the port number 17006 to the non-SSL end-entity port for your CA, you would go to the URL `http://hostname.17006` to bring up the Certificate Manager page for end entities.

- c. In the left-hand frame, under Server, click OCSP Responder.
- d. In the OCSP Responder Enrollment page that appears, paste the request from the clipboard into the field labeled PKCS #10 Request and fill in any other required information.
- e. Click Submit.
- f. If the request contains all the required information, you'll get a notification of request being successfully added to the agent queue of that Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate.
- g. In the web browser window, enter the URL for the Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- h. Select List Requests, then click Show Pending Requests and click Find. The pending request list is displayed.
- i. Locate your request, click Details to see it.

- j. After checking the rest of the certificate request and making any changes, scroll to the bottom, and click Do It.
- k. After the certificate is generated, click Show Certificate.
- l. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 7).

Also note that you might be required to paste the CA certificate chain in the Installation Wizard. So, keep the browser window open.

To submit your certificate request manually to a third-party CA, follow these steps:

- a. Make sure that the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) is highlighted, and click the Copy to Clipboard button.

This action copies the certificate request 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 Online Certificate Status Manager's signing certificate.

- b. Submit your certificate request to a third-party CA, following the instructions provided by that CA.

Click Next when you are ready to proceed.

7. Online Certificate Status Manager Signing Certificate Installation.

Depending on whether you have the certificate ready for pasting into the Installation Wizard screen, click Yes or No.

- o If you have submitted your request to a third-party CA or to a remote Certificate Manager for which you do not have agent privileges, 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. The default selection is No.
- o Select Yes, only if you have the certificate ready in its base-64 encoded format.

Click Next to continue.

- If you selected Yes, the “Location of Certificate” screen appears (Step 8).
- If you selected No, you will be presented with the “Key-Pair Information for SSL Server Certificate” screen (Step 11).

8. Location of Certificate. Specify the location of the certificate. You can use one of these options:

- If you noted the file path to the file that contains the certificate (in its base 64-encoded format), select the “The certificate is located in this file” option and type the file path, including the filename, in the text field.
- If you copied the certificate (in its base 64-encoded format) to the clipboard, select the “The certificate is located in the text area below” option and paste the certificate (including the header and footer) in the text area provided.
- If you want the wizard to retrieve the certificate from the remote CMS manager to which you submitted the request, select the “The certificate is at the CMS where the request was sent” option and supply the host name, end-entity port number, and request ID.

Click Next to continue.

9. Certificate Details. This is an informational screen that displays the certificate so you can inspect its contents. Notice the nickname assigned to the certificate and verify that you’re installing the correct certificate.

Click Next to continue.

10. Import Certificate Chain. This screen appears only if you need to import the CA certificate chain. Follow these steps to import the CA chain of a Certificate Manager:

- a. Go back to the web browser window from which you copied the Online Certificate Status Manager’s signing certificate (in its base-64 encoded format).
- b. Scroll down to the part that says “Base 64 encoded certificate with CA certificate chain in pkcs7 format” and shows the CA certificate chain in its PKCS#7 format.
- c. Highlight all the encoded blob (including -----BEGIN CERTIFICATE ----- and -----END CERTIFICATE-----), and copy it to the clipboard or to a text file.

Be sure to not make any changes to the certificate. You’re required to paste the encoded certificate into the Installation Wizard next. So, once you’ve copied the certificate, go back to the wizard screen.

- d. Paste the certificate chain into the text box.
- e. Click Next to continue.

If you closed the end-entity interface, you can get the CA certificate chain this way:

- a. Open a web browser window.
 - b. Go to the end-entity URL for the Certificate Manager that issued the Online Certificate Status Manager's signing certificate.
 - c. Select the Retrieval tab, and then choose Import CA Certificate Chain.
 - d. Select the "Display the CA certificate chain in PKCS#7 for importing into a server" option, and then click Submit.
 - e. Copy the certificate chain to the clipboard.
 - f. Return to the Installation Wizard.
 - g. Paste the certificate chain into the text box.
 - h. Click Next to continue.
- 11. Key-Pair Information for SSL Server Certificate.** Select the token 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. Also specify the key type and length.

Click Next to continue.

- 12. Subject Name for SSL Server Certificate.** Type the values for the subject DN components; these values identify the Online Certificate Status Manager's SSL server certificate. The CN must be the fully-qualified host name of the machine on which you're installing the Online Certificate Status Manager.

Click Next to continue.

- 13. Certificate Extensions for SSL Server Certificate.** Select the required extensions. 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.

Certificate Management System provides command-line tools for generating extensions to include in CA and other certificate requests. For details about these tools, check this directory: `<server_root>/bin/cert/tools`

Note that 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 in the `tools` directory. For details on using the `ExtJoiner` program, see Chapter 5, “Extension Joiner Tool” of *CMS Command-Line Tools Guide*.

Click Next to continue.

14. **SSL Server Certificate Request Creation.** This is an informational screen that tells you that the wizard has all the information required to generate the key pair and certificate request. In the previous screen, if you chose to include the Subject Key Identifier extension in the certificate, you’ll be given the choice to select the format for the certificate request. Otherwise, the request format will be PKCS #10.

- o If you want the wizard to generate the certificate request in PKCS #10 format, select the “Generate PKCS10 request” option.
- o If you want the wizard to generate the certificate request in CMC format, select the “Generate CMC full enrollment request” option.

Click Next. The wizard generates the certificate request that you must submit to a CA.

15. **Submission of Request.** Select whether you want to submit the request manually or send the request to a remote CMS server (Certificate Manager or Registration Manager) automatically.

To automatically submit the request to a remote Certificate Manager (or for automatic enrollment), follow these steps:

- a. Select the “Send the request to a remote CMS now” option.
- b. Enter the host name and end-entity port number of the Certificate Manager, and select whether this end-entity port uses SSL.

- c. Click Next to submit the request.

The Certificate Request Result screen appears, confirming that the request has been submitted. Note the request ID provided in the response message. (You can use it later to retrieve the certificate, once it has been issued, from the end-entity port.)

Note that your request gets added to the agent queue of the Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you should wait for the other agent to approve your request and issue the certificate.

- d. In the web browser window, enter the URL for the Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- e. Select List Requests, then click Show Pending Requests and click Find. The pending request list is displayed.
- f. Locate your request, click Details to see it, and make any changes. Then scroll down to the bottom of the form and click Do It.
- g. After the certificate is generated, click Show Certificate.
- h. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 16).

To submit your certificate request manually to a Certificate Manager, follow these steps:

- a. Open a web browser window.
- b. Go to the end-entity URL for the Certificate Manager that will issue the SSL server 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.

- c. In the left-hand frame of the Enrollment tab, choose the form appropriate for the request type:

If the request is in the PKCS #10 format, under Server, click SSL Server. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information.

If the request is in the CMC format, click CMC Enrollment. In the resulting form, paste the request from the clipboard into the text area and fill in any other required information. Be sure to select `Server SSL Certificate` as the certificate type.

- d. Click Submit.
- e. The request gets added to the agent queue of that Certificate Manager for approval by that Certificate Manager's agent. If you've permission to access that Certificate Manager's Agent interface, you can follow the instructions below to issue the certificate. Otherwise, you'll have to wait for the Certificate Manager's agent to approve your request and issue the certificate.
- f. In the web browser window, enter the URL for the Certificate Manager's Agent Services page. (You must use the same computer where you got your agent certificate.)
- g. Select List Requests, then click Show Pending Requests and click Find. The pending request list is displayed.
- h. Locate your request, click Details to see it, and make any changes. Then, scroll down to the bottom of the form and click Do It.
- i. After the certificate is generated, click Show Certificate.
- j. 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 or to a text file.

Be sure to not make any changes to the certificate. You're required to paste the encoded certificate into the Installation Wizard next. So, once you've copied the certificate, go back to the wizard screen (Step 16).

To submit your certificate request manually to a third-party CA, follow these steps:

- a. Make sure that the certificate request (including -----BEGIN NEW CERTIFICATE REQUEST ----- and -----END NEW CERTIFICATE REQUEST -----) is highlighted, and click the Copy to Clipboard button.

This action copies the certificate request 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.

- b. Submit your certificate request to a third-party CA, following the instructions provided by that CA.

Click Next when you are ready to proceed to the next screen.

16. **SSL Server Certificate Installation.** Depending on whether you have the certificate ready for pasting into the Installation Wizard screen, click Yes or No.

- If you have submitted your request to a third-party CA or to a remote Certificate Manager for which you do not have agent privileges, 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. The default is No.
- Select Yes, only if you have the certificate ready in its base-64 encoded format.

Click Next to continue.

- If you selected Yes, the “Location of Certificate” screen appears (Step 17).
- If you selected No, you will be presented with the “Create Single Signon Password” screen (Step 20).

17. **Location of Certificate.** Specify the location of the certificate. You can use one of these options:

- If you copied the encoded certificate to a file, select the “The certificate is located in this file” option and type the file path, including the filename, in the text field.
- If you copied the certificate to the clipboard, select the “The certificate is located in the text area below” option and then paste in a base-64 encoded certificate (including the header and footer) in the text area provided.

- If you know the request ID of your request and the host name and end-entity port number of the Certificate Manager that issued the SSL server certificate, select the “The certificate is at the CMS server where the request was sent” option and then specify the required details.

Click Next to continue.

- 18. Certificate Details.** This is an informational screen that displays the certificate so you can inspect its contents. Notice the nickname assigned to the certificate and verify that you’re installing the correct certificate.

Click Next to continue.

- 19. Import Certificate Chain.** This screen appears only if you need to import the CA certificate chain. Follow these steps to import the CA chain of a Certificate Manager:

- a. Go to the web browser window.
- b. Enter the end-entity URL for the Certificate Manager that issued the SSL server certificate.
- c. Select the Retrieval tab, and then choose Import CA Certificate Chain.
- d. Select the “Display the CA certificate chain in PKCS#7 for importing into a server” option, and then click Submit.
- e. Copy the certificate chain to the clipboard.
- f. Return to the Installation Wizard.
- g. Paste the certificate chain into the text box.
- h. Click Next to continue.

- 20. Create Single Signon Password.** Type the single signon password. The single signon password simplifies the way you subsequently sign on to Certificate Management System by storing the passwords for the internal database, tokens, and so on. Each time you log on, you’re only required to enter this single password. (For details, see “Required Start-up Information” on page 322.)

Click Next to continue.

- 21. Configuration Status.** This screen should indicate that your configuration has been successful and that you need to create an agent for the Online Certificate Status Manager.

Click Done to exit the Installation Wizard.

22. Proceed to the next step, “Stage 3. Enrolling for Administrator/Agent Certificate” on page 277, to create an *agent* user for the Online Certificate Status Manager.

Stage 3. Enrolling for Administrator/Agent Certificate

Immediately after installing any CMS instance, the administrator must enroll for the initial administrator/agent certificate. This is the first user (agent) 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.

Follow the appropriate procedure for the subsystem you installed:

- Agent Certificate for a Certificate Manager
- Agent Certificate for Other CMS Managers

For more information about setting up and managing agents, see “Agents” on page 397.

Agent Certificate for a Certificate Manager

If the CMS instance you installed contains a Certificate Manager, a special enrollment form, Administrator/Agent Certificate Enrollment Form, allows you to get this first certificate automatically. After you submit this initial 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>`

- o For `<hostname>`, provide the fully qualified name of the machine on which Certificate Management System is installed; for example, `CAmachine.siroe.com`.
- o The `<agent_port>` 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. Type the ID you entered for the CMS administrator during installation.

Password. Type 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. Type the name of administrator/agent.

Login name. Type the user ID of administrator/agent.

Email address. Type the email address of administrator/agent.

Organization unit. Type the name of the organization unit to which the administrator/agent belongs.

Organization. Type the name of the company or organization the administrator/agent works for.

Country. Type the two-letter code for the administrator/agent's country.

User's Key Length Information

Key Length. Type 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 iPlanet Console, you use the 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. Here's how you can do this:

1. In the left frame of iPlanet 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 this directory: `<server_root>/cert-<instance_id>/config`

4. Open the configuration file (`CMS.cfg`) in a text editor.
5. Locate the following line: `agentGateway.enableAdminEnroll=false`
6. Change `false` to `true`, and save the file.
7. 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.
8. The next time you access the SSL agent port, the Administrator/Agent Certificate Enrollment form will be available again.

Agent Certificate for Other CMS Managers

If the CMS instance you installed doesn't include a Certificate Manager—for example, if it's a standalone Registration Manager, Data Recovery Manager, or Online Certificate Status Manager—you need to manually submit a client certificate request to the CA and then install the certificate in the certificate database of the CMS instance. Alternatively, if you have agent privileges to any of the CMS managers, for example to a Certificate Manager, you can use the same agent certificate for performing agent tasks of another CMS manager. This you can do by storing a copy of the agent's SSL client certificate in the internal database of the newly-installed CMS manager.

The instructions below assume that you already have a client certificate for a Certificate Manager (whether the original administrator/agent certificate or a new agent certificate) and want to use the same certificate for agent operations of another CMS manager.

1. Make sure the Certificate Manager is started.
2. Open a web browser window.
3. Go to the end-entity interface of the Certificate Manager that issued the certificate you want to use.
4. Select the Retrieval tab
5. Click List Certificates or Search for Certificates and locate the certificate (check the subject name of the certificate).
6. Copy the certificate in its base-64 encoded format (or keep the browser window open so that you can copy the certificate later in this procedure).
7. Log in to iPlanet Console (see “Logging In to iPlanet Console” on page 344).

8. In the navigation tree, locate the CMS instance for which you want to create the agent user, and double-click the icon.

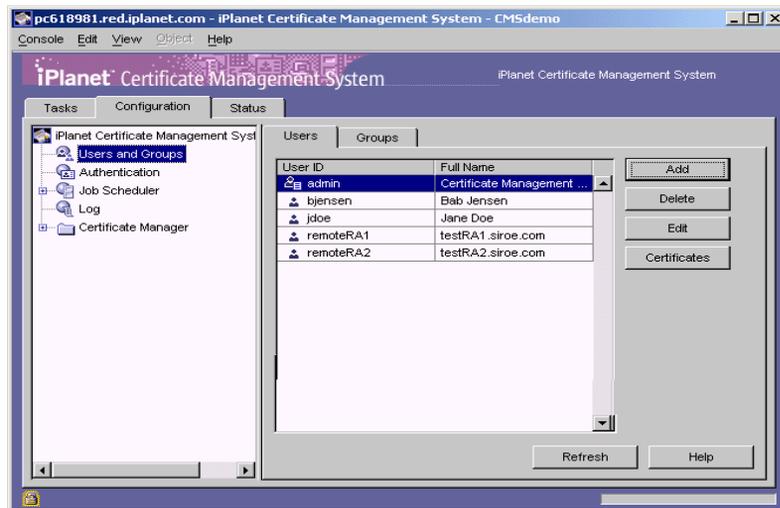
The login screen for the CMS window appears.

9. Enter your administrator ID and password.

The CMS window for the subsystem opens.

10. In the navigation tree, select Users and Groups.

The Users tab appears.



11. Select the user ID for the administrator, the one created during installation, and click Certificates.

The Manage User Certificates window appears.

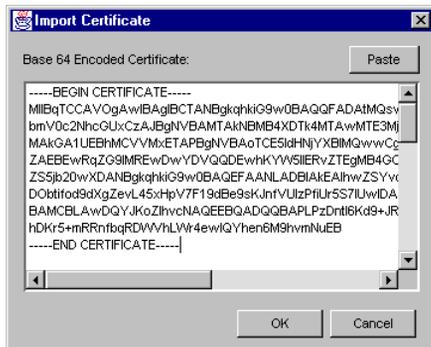


12. Click Import.

The Import Certificate window appears.

13. Click inside the text area, and paste the agent's certificate in base-64 encoded form. (If you haven't copied the certificate, go back to the browser window, copy the certificate, and then paste the certificate here.)

Be sure to include the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines.



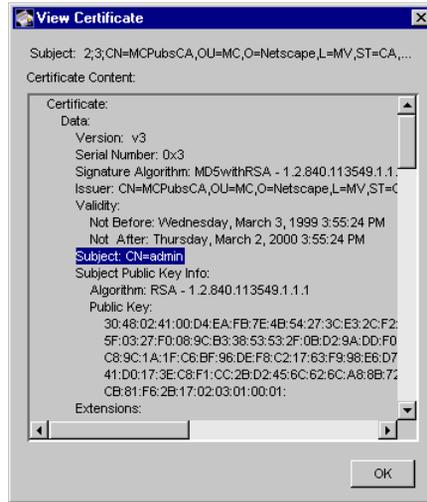
14. Click OK.

You are returned to the Manage User Certificates window. The certificate you imported should now be listed in this window.



15. To view the certificate you imported, select it and click View.

The certificate information appears.



16. Click Done.

You are returned to the Users tab.

17. Click Refresh to view the updated configuration.

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 “Agents” on page 397.

Stage 4. 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 iPlanet 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. If you installed a Data Recovery Manager, you may want to configure your Certificate Manager or Registration Manager to archive end users’ encryption private keys with the Data Recovery Manager.

For detailed information about the many CMS configuration options available, check the chapters in Part 3, “Configuration.” You might find it useful to read “Road Map to Configuring Subsystems” on page 376.

Stage 5. Creating Additional Instances or CA Clones

After the initial installation, you can use iPlanet Console to create additional instances of Certificate Management System in the same server root directory. Once you have a new instance, you can use the Installation Wizard and CMS window to configure any new instances.

- For instructions to create an additional instance of Certificate Management System, see “Installing Multiple CMS Instances” on page 286.
- For instructions to clone a Certificate Manager (CA), see “Cloning a Certificate Manager” on page 288.

Installing and Uninstalling CMS Instances

After the initial installation of iPlanet Certificate Management Server (CMS), you may need to install additional instances, remove unwanted instances, or duplicate configuration in multiple instances. This chapter describes how to manage these tasks by using iPlanet Console, the single, unified administration interface for your network.

You can use iPlanet Console only when iPlanet Administration Server is running. During CMS installation, you specified a port number for the Administration Server instance you will use to administer Certificate Management System. If Administration Server is shut down, be sure to start it at this port. To minimize security risks, shut down the Administration Server when you have finished using iPlanet Console. For more information about iPlanet Console, see , “Administration Tasks and Tools.”

The chapter has the following sections:

- Installing Multiple CMS Instances (page 286)
- Cloning a Certificate Manager (page 288)
- Viewing Instance Information (page 306)
- Changing the Name of an Instance (page 308)
- Removing an Instance From a System (page 309)
- Uninstalling Certificate Management System (page 311)

Installing Multiple CMS Instances

Multiple instances of Certificate Management System can run on the same machine. You might, for example, install multiple Registration Managers, all reporting to the same Certificate Manager, to handle requests from different types of users (end users, servers, and routers) or from users from different domains. For example deployment scenarios, see Chapter 4, “Planning Your Deployment.”

Once Certificate Management System is installed on a machine, you can use that CMS installation to create multiple instances of the server on the same machine. Administration Server contains all the files necessary to install another instance of Certificate Management System on the same machine; you don't have to run the complete installation (`setup`) program again. However, you do need to run the CMS installation wizard each time you create an instance, so that you can configure the server and generate the required certificates. So, before attempting to create another instance of Certificate Management System, be sure to read about the installation wizard explained in Chapter 6, “Installing Certificate Management System.”

When you install additional CMS instances on the same machine, you are required to specify different ports for each CMS instance to listen on. For example, you will have to set up one server to listen on port 443 and another server to listen on port 4430. However, if you install multiple CMS instances on a machine that has been set up with more than one IP addresses, you can configure each instance to listen to a specific IP address—this enables you to use same the port numbers for different CMS instances installed on the same machine.

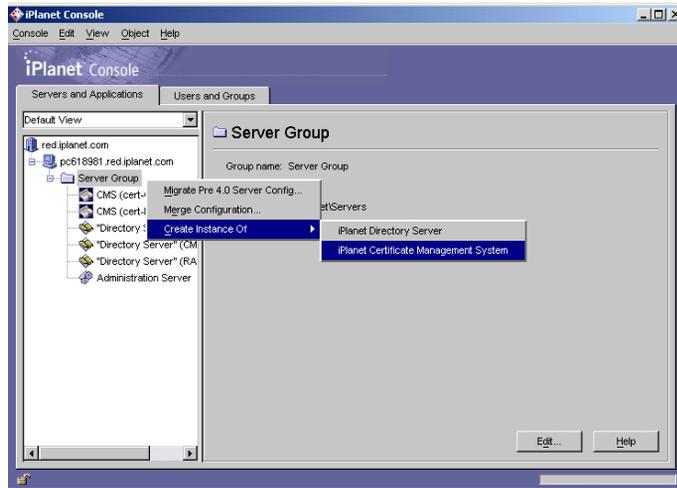
Keep in mind that when you create a new instance, you'll be required to specify different port numbers; the installation wizard allows you to specify the port numbers only, not IP addresses. After you have successfully created the instance, you can edit the CMS configuration file to specify the IP addresses for individual CMS ports and then change the port numbers. For details on editing the configuration file to include the IP addresses, see “Step 2: Specify IP Addresses” on page 387. For details on changing the port numbers, see “Configuring Port Numbers” on page 384.

To create another instance of Certificate Management System with a separate configuration file (and certificates):

1. Log in to iPlanet Console (see “Logging In to iPlanet Console” on page 344).
2. In the navigation tree at the left, expand the icon for your computer, then select the server group that contains the CMS instance you want to use as your source.

- From the Object menu, choose the Create Instance Of option and, in the pop-up menu that appears, choose Certificate Management System.

As shown in this figure, you can also right-click to choose this option from the pop-up menu.



The Create Server Instance dialog box appears.



- Type a unique name or identifier for the new instance.

For the name, you can use any combination of letters (aA to zZ), digits (0 to 9), an underscore (_), and a hyphen (-); other characters and spaces are not allowed. For example, you can type `Siroe_root-CA` as the instance name, but not `Siroe root CA`.

- Click OK.

The instance you created appears in the navigation tree. Note that the instance name appears in the CMS (cert-`<instance_id>`) form, where `<instance_id>` is the name you specified for the new CMS instance. For example, if you named the instance `Marketing_CA`, the instance name in the navigation tree will be CMS (cert-`Marketing_CA`).

6. To start the installation wizard, double-click the new instance in the navigation tree, and then use the installation wizard to finish configuring the new instance.
7. Create the first agent for the new 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 “Agent Certificate for a Certificate Manager” on page 277 as you did for the first instance in the server root. If the new instance does not include a Certificate Manager—that is, if it contains a Registration Manager, Data Recovery Manager, Online Certificate Status Manager, Registration Manager and Data Recovery Manager, or Online Certificate Status Manager and Data Recovery Manager—you will need to use the CMS window to create a new agent. This is described in section “Agent Certificate for Other CMS Managers” on page 280.

Cloning a Certificate Manager

Cloning a Certificate Manager refers to the process of creating two server processes performing the same CA functions: you create another instance of a Certificate Manager and configure it to use the same CA signing key and certificate and issue certificates with serial numbers that do not conflict or overlap with the serial numbers of the Certificate Manager that’s being cloned or with the serial numbers of any other clones. The Certificate Manager that’s being cloned is called the *master Certificate Manager* or *master CA* in this document.

You can use the cloning feature for CA scalability and for setting up a PKI with CAs organized in a flat structure as opposed to a hierarchical structure. For example, if you don’t want your PKI to be a CA hierarchy comprising root and subordinate CAs, you can create multiple clones of a Certificate Manager and configure each clone to issue certificates that fall within a distinct range of serial numbers. Because clone CAs use the same CA signing key and certificate (as that of the master CA) to sign the certificates they issue, the *issuer name* in all the certificates in your PKI setup would be the same, as if they’ve been issued by a single CA.

The other advantage of cloning is that when you setup a clone Certificate Manager, it automatically sends the revocation status of the certificates it has issued to the master Certificate Manager. The clone Certificate Manager uses the master Certificate Manager’s agent port to communicate this information; the

communication is SSL-client authenticated. This way, the master Certificate Manager has the complete list of certificates revoked by all clone Certificate Managers and is able to generate a consolidated list of revoked certificates or a complete CRL.

Because the master Certificate Manager has the complete CRL, if you enable the OCSP-service feature built into the Certificate Manager, it can function as a full-fledged OCSP responder for your PKI—that is, irrespective of which clone Certificate Manager has issued the certificate, OCSP-compliant clients can directly query the master Certificate Manager for the revocation status of a certificate. (For information on enabling a Certificate Manager’s OCSP service, see “Setting Up a Certificate Manager with OCSP Service” on page 719.) So, CAs organized in a flat structure using the cloning method eliminate the need for you to install the standalone OCSP responder, the Online Certificate Status Manager, and configure each Certificate Manager to publish its CRL to the Online Certificate Status Manager.

To setup a clone a Certificate Manager (or a CA), follow these steps:

- Step 1. Before You Begin
- Step 2. Create Instances for Clone CAs
- Step 3. Shutdown the Master CA
- Step 4. Copy Master CA’s Certificate and Key Database
- Step 5. Start the Master CA
- Step 6. Configure the Clone CA
- Step 8. Establish Trust Between Master CA and Clone CAs
- Step 9. Test Clone-Master Connection
- Step 10. Use Master CA’s Agent Certificate in Clone CAs

Step 1. Before You Begin

Before you start cloning a Certificate Manager:

- Verify that the master Certificate Manager is installed and configured properly, and is started.

- Check the master Certificate Manager’s serial number range. The “Next serial number” field should be set to the next serial number of the certificate the CA will issue and the “Last serial number” field must be blank. To locate the panel that enables you to do this, see “Enabling End-Entity Interaction with a Certificate Manager” on page 563.
- If the master Certificate Manager’s keys and certificates are stored in a hardware token, check the token vendor’s documentation for copying the keys and certificates (from the original token to a new token accessible to the clone Certificate Manager). Keep the relevant instructions handy.
- Decide how many clone CAs you need to deploy, and note the following for each clone CA.
 - CA’s serial number range—Each clone Certificate Manager must be configured to issue certificates with unique serial numbers. Which means, when you configure a clone Certificate Manager, you must specify upper and lower bounds for the serial numbers and make sure that the serial-number range does not overlap with the one specified for another clone Certificate Manager.

When specifying the serial number range for the first clone Certificate Manager, it’s recommended that you start with, say, 0x100, as the starting/lowest serial number. This will ensure that the master Certificate Manager has sufficient serial numbers for its own certificates, such as the CA signing certificate, SSL server certificate, agent’s certificate, and so on. The master Certificate Manager will also need distinct serial numbers in the future, for example, when you renew its certificates in the future. Any subsequent clone Certificate Manager does not need to make such a provision; its serial numbers only need to not overlap with the ones assigned to the previous clones.

- CA’s signing key and certificate—You must use the master Certificate Manager’s signing key and certificate. If you do not use the master Certificate Manager’s key and certificate databases, the clone Certificate Manager will need to generate a new signing key and certificate; consequently, it will not be a clone.
- CA’s SSL server key and certificate—This depends on the hostname of the clone Certificate Manager. If the clone Certificate Manager uses the same hostname as that of the master Certificate Manager, you can use the same SSL server certificate and key copied from the master Certificate Manager. If the hostnames are different, you must generate a new SSL server certificate for the clone Certificate Manager; 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.

During the cloning process, the master Certificate Manager's SSL server certificate is automatically copied to the certificate database of the clone Certificate Manager. The clone Certificate Manager uses this certificate for SSL-client-authenticated communication with the master Certificate Manager. Don't be alarmed when you see the certificate in clone Certificate Managers' certificate databases. Also, be sure not to remove them from the master and clone Certificate Managers' databases.

If you want to note the details such as the serial numbers and ports used by the clone CAs, use the section "Cloned Certificate Manager Configuration" in Chapter 5, "Installation Worksheet."

Step 2. Create Instances for Clone CAs

Depending on how many clone CAs you want to deploy, follow the appropriate instructions in this step to create that many CMS instances.

Depending on your master Certificate Manager's installation, there are three possible scenarios to install a clone Certificate Manager:

- **Installing Clone CA in Master CA's Server Group**—In this case, you install the clone Certificate Manager in the same server group in which the master Certificate Manager is installed.
- **Installing Clone CA in a Different Server Group**—In this case, you install the clone Certificate Manager on the same host on which the master Certificate Manager is installed, but in a different server group.
- **Installing Clone CA on a Separate Host**—In this case, you install the clone Certificate Manager on a different host than the one on which the master Certificate Manager is installed.

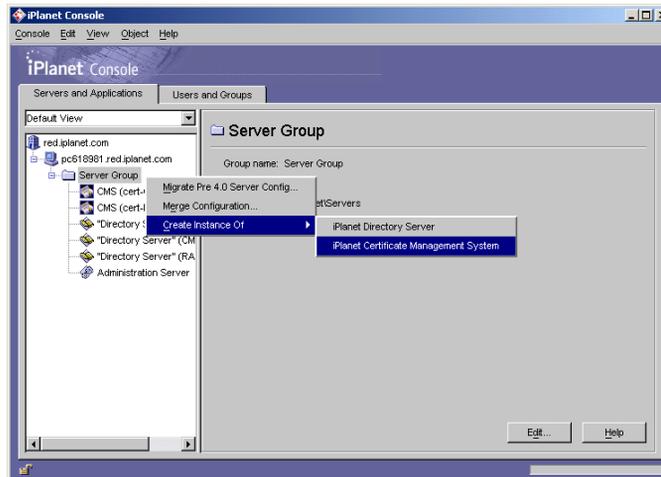
Installing Clone CA in Master CA's Server Group

If you want to install your clone Certificate Manager in the same server group as that of the Certificate Manager:

1. Log in to iPlanet Console of the master Certificate Manager (see "Logging In to iPlanet Console" on page 344).
2. In the navigation tree at the left, expand the icon for your computer, then select the server group that contains the CMS instance you want to use as your master.

- From the Object menu, choose the Create Instance Of option and, in the pop-up menu that appears, choose Certificate Management System.

As shown in this figure, you can also right-click to choose this option from the pop-up menu.



The Create Server Instance dialog box appears.

- Type a unique name or identifier for the new instance.

For the name, you can use any combination of letters (aA to zZ), digits (0 to 9), an underscore (_), and a hyphen (-); other characters and spaces are not allowed. For example, you can type `Clone1_of_root-CA` as the instance name, but not `Clone1 of root-CA`.

- Click OK.

The instance you created appears in the navigation tree. Note that the instance name appears in the `CMS (cert-<instance_id>)` form, where `<instance_id>` is the name you specified for the new CMS instance. For example, if you named the instance `Clone1_of_root-CA`, the instance name in the navigation tree will be `CMS (cert-Clone1_of_root-CA)`.

Installing Clone CA in a Different Server Group

In a Windows NT installation of Certificate Management System, you cannot create more than one server group. In a Unix installation, you can create multiple server groups. For more information, see section "The Administration Server" of *Managing Servers with iPlanet Console*.

If you want to install your clone Certificate Manager on the same host on which the master Certificate Manager is installed, but in a different server group:

1. In the master Certificate Manager host machine, go to the directory that contains the `CMS setup` program.
2. Run the `setup` program. For instructions, see “Stage 1. Running the Installation Script” on page 221. Be sure to follow these guidelines:
 - When prompted to chose a *server root* or the location for the installation, specify a different directory/folder (than the one where the master Certificate Manager is installed). For example, if your master Certificate Manager is installed at `/u/iplanet/servers`, you can specify `/u/iPlanet/server4_clone1` as the server root for the clone instance.
 - When prompted to specify a *configuration directory*, select the option for an existing directory and specify the host name and port number of the Directory Server instance used by the master Certificate Manager.
 - When prompted to specify a port number for the Administration Server, be sure to specify distinct port number; each server group is managed by a separate Administration Server. Note the port number as you will need this later to log in to iPlanet Console.

Installing Clone CA on a Separate Host

If you want to install your clone Certificate Manager on a different host than the one on which the master Certificate Manager is installed, you should run the `CMS setup` program on that host. For instructions, see “Stage 1. Running the Installation Script” on page 221. (Note that you only need to complete Stage 1 and then proceed to “Step 3. Shutdown the Master CA” below.)

Step 3. Shutdown the Master CA

Stop the master Certificate Manager. If you need instructions, see “Stopping Certificate Management System” on page 330.

Step 4. Copy Master CA's Certificate and Key Database

Because you want the clone Certificate Manager to own the same keys and certificates as that of the master Certificate Manager, you need to make available the keys and certificates used by the master Certificate Manager to each clone Certificate Manager.

- If the master Certificate Manager's keys and certificates are stored in the internal/software token, you need to copy the `key3.db` and `cert7.db` files from the `config` directory of the master Certificate Manager to the `config` directory of each clone Certificate Manager. Here's how you do this:
 - a. In the master Certificate Manager's host machine, go to this directory:
`<server_root>/cert-<instance_id>/config`
 - b. Locate files named `cert7.db` and `key3.db`.
 - c. In the clone Certificate Manager's host machine, go to this directory:
`<server_root>/cert-<instance_id>/config`
 - d. Copy the `cert7.db` and `key3.db` files from master Certificate Manager to the clone.
 - e. Repeat steps c and d to copy the master Certificate Manager's `key3.db` and `cert7.db` files to the `config` directory of each clone Certificate Manager.
- If the master Certificate Manager's keys and certificates are stored in the hardware token, you need to copy the keys and certificates following the instructions provided by the hardware-token vendor.

Step 5. Start the Master CA

Start the master Certificate Manager. If you need instructions, see "Starting Certificate Management System" on page 322.

Step 6. Configure the Clone CA

Depending on how many CMS instances you've created for clone Certificate Managers, you should repeat the instructions in this step to configure each clone Certificate Manager.

To configure a clone Certificate Manager:

1. Log in to or go to iPlanet Console that shows the clone Certificate Manager instance.
2. In the navigation tree, locate the instance ID for the clone you created, and double-click the instance.

The CMS Installation Wizard starts.

3. Follow the on-screen instructions to finish configuring the clone CA. During configuration, be sure to follow these:
 - **Clone key and certificate materials**—On this screen, click Yes to reuse the certificate and key material in the database files you copied from the master Certificate Manager. In the Instance Name field enter the instance ID of the master Certificate Manager. Select the token name where the keys and certificate are stored and enter the token’s password, if required.
 - **Clone key and certificate materials**—On this screen, you choose whether to reuse the master Certificate Manager’s SSL server certificate or create a new one. If you created the clone Certificate Manager on the same host as the master Certificate Manager, you can reuse the SSL server certificate. To reuse the SSL server certificate, select Yes, enter the instance ID of the master Certificate Manager, select a token, and enter the token password. If you do not or cannot reuse the SSL server certificate, select No and follow the screens that enable you to generate a new SSL server certificate.
 - **CA’s serial number range**—On this screen, specify the lowest serial number the CA should assign to certificates it creates in the “Starting serial number” field. In the “Ending serial number” field, specify the highest serial number available for this CA. For both the fields, you can enter the number in decimal or hexadecimal (0xnn).
4. Repeat steps 1 through 3 for each clone Certificate Manager.

Step 8. Establish Trust Between Master CA and Clone CAs

For the master Certificate Manager to trust the clone Certificate Manager, you associate the clone Certificate Manager as a *trusted manager* to the master Certificate Manager. For details about trusted managers, see “Trusted Managers” on page 405.

The setup process involves the following steps:

- Step A. Locate the Master CA's SSL Server Certificate
- Step B. Create a Privileged-User Entry for Clone CAs

Step A. Locate the Master CA's SSL Server Certificate

Depending on which CA issued/signed the master Certificate Manager's SSL server certificate, you can locate the certificate in either the internal database or the certificate database (`cert7.db` file).

- If the issuer of the SSL server certificate is the master Certificate Manager itself, you can locate the certificate in the internal database by going to the Retrieval tab of the master Certificate Manager's end-entity interface.
- If the issuer of the SSL server certificate is another CA, for example, a third-party CA, you can locate the certificate in the certificate database by using the `certutil` command-line tool. For more information about this tool, see Chapter 11 , "Certificate Database Tool" of *CMS Command-Line Tools Guide*.

Follow the instructions that's appropriate for you.

To locate the certificate in the Retrieval tab of the end-entity interface:

1. Open web browser window.
2. Go the master Certificate Manager's end-entity interface. The URL is in `https://<hostname>:<SSL_port>` or `http://<hostname>:<port>` format.
3. Select the Retrieval tab, and in the left frame, click List Certificates.
4. In the resulting form, click List.
A list of certificates appear.
5. Locate the Certificate Manager's SSL server certificate by looking at the subject name of the certificate.
Typically, the SSL server certificate would be the second certificate.
6. Click Details.
7. In the resulting page, scroll to the section that says "Base 64 encoded certificate" and shows the certificate in its base-64 encoded format.

8. Copy the base-64 encoded certificate, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines, to the clipboard or a text file. (Alternatively, you can keep the browser window open and copy the certificate later in the procedure.)

The copied information should look similar to the following example:

```
-----BEGIN CERTIFICATE-----
MIICJzCCAZCgAwIBAgIBAzANBgkqhkiG9w0BAQQFADBCMSAwHgYDVQQKEXdOZXRz
Y2FwZSBDb21tdW5pYF0aW9uczngjhnMVQ2VydGhmaWNhdGUgQXV0aG9yaXR5MB4X
DTk4MDgyNzE5MDAwMFoXDTk5MDIyMzE5MDAwMnBjdGngYoxIDAeBgNVBAoTF05ld
HNjYXBlIENvbW11bmljYXRpb25zMQ8wDQYDVQQLEWZQZWR9wGUXFzAVBgoJkiaJk
IsZAEBEwdzdxByaXlhMRcwFQYDVQQDEw5TdXByaXlhIFNoZXR0eTEjMCEGCSqGSI
b3DbnGJARYUC3Vwcm15YUBuZXRzY2FwZS5jb20wXDANBgkqhkiG9w0BAQEFAANL
ADBIAkEAoYiYgthgtbbnjfngjnjgnagwJjaOBgNVHQ8BAf8EBAMCBLAwFAYJYIZI
AYb4QgEBAQHBAQDAgCAMA0GCSqGSIb3DQEBAUAA4GBAFi9FzyJlLmS+kzsue0kT
XawbwamGdYq12w4hIBgdR+jWeLmD4CP4xzmKdvQ6IqD2q8DBs91RQu9
-----END CERTIFICATE-----
```

To locate the SSL server certificate in the master Certificate Manager's certificate database using the `certutil` tool:

1. Open a terminal window in the system that hosts the master Certificate Manager.
2. Go to this directory: `<server_root>/cert-<instance_id>/config`
3. Next, run this command:

```
<server_root>/bin/cert/tools/certutil -L -d <certdir>
-n <certname> cert-<instance_id> -a
```

replacing `<certdir>` with the directory containing a certificate database file, `<certname>` with the nickname of the SSL server certificate, and `<instance_id>` with the ID assigned to the master Certificate Manager instance.

For example, your command might look like this:

```
<server_root>/bin/cert/tools/certutil -L -d . -n Server-Cert
cert-masterCA -a
```

The SSL server certificate appears.

4. View the certificate information. Make sure that the certificate you are looking at is the correct one; the certificate shows the DN that was specified for the certificate during the installation.

5. Copy the base-64 encoded certificate, including the marker lines -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----, to the clipboard or to a text file. The copied information should look like the example below:

```
-----BEGIN CERTIFICATE-----
MIICJzCCAZCgAwIBAgIBAzANBgkqhkiG9w0BAQQFADBCMSAwHgYDVQQKEXdOZXRz
Y2FwZSBDb21tdW5pYF0aW9uczngjhnMVQ2VydG1maWNhdGUGQXV0aG9yaXR5MB4X
DTk4MDgyNzE5MDAwMFoXDtk5MDIyMzE5MDAwMnBjdGngYoxIDAeBgNVBAoTF05ld
HNjYXB1IENvbW11bmljYXRpb25zMQ8wDQYDVQQLewZQZW9wbGUxZzAVBgoJKiaJk
IsZAEBEwdzdXByaXlhMRcwFQYDVQQDEw5TdXByaXlhIFNoZXR0eTEjMCEGCSqGSI
b3DbnmgJARYUc3Vwcm15YUBuZXRzY2FwZS5jb20wXDANBgkqhkiG9w0BAQEFAANL
ADBIAKEAoYiYgthgtbbnjfngjnjgnagwJjAOBgNVHQ8BAf8EBAMCBLAwFAYJYIZI
AYb4QgEBAQHBAQDAgCAMA0GCSqGSIb3DQEBAUAA4GBAFi9FzyJlLmS+kzsue0kT
XawbwamGdYql2w4hIBgdR+jWeLmD4CP4xzmKdvQ6IqD2q8DBS9lRQu9JYg129o
-----END CERTIFICATE-----
```

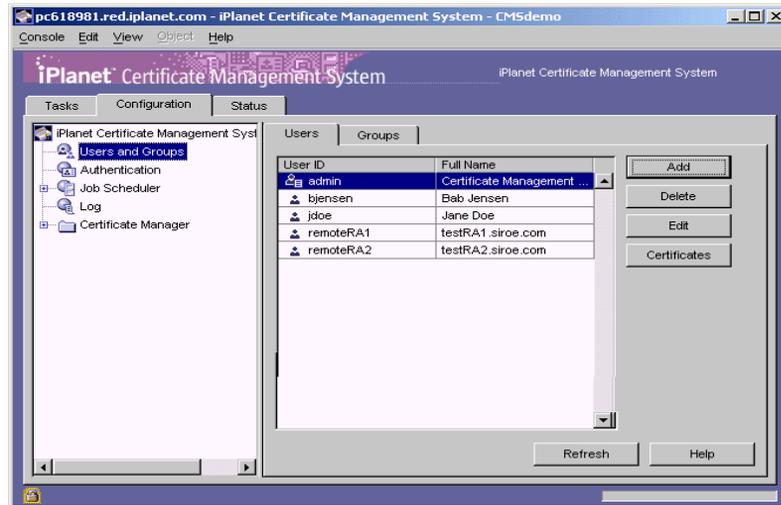
Step B. Create a Privileged-User Entry for Clone CAs

In this step, you create a privileged-user entry for all clone Certificate Managers in the internal database of the master Certificate Manager. As a part of creating this entry, you also add the new user entry to the `Trusted Managers` group in order to give the entry access privileges to the agent port of the master Certificate Manager.

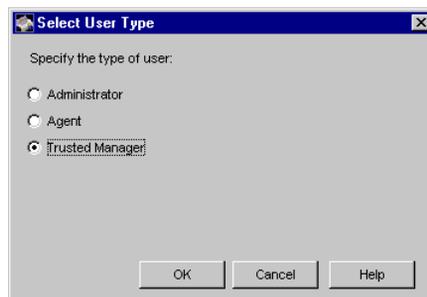
To create a user entry with appropriate access privileges:

1. Log in to or go to the CMS window for the master Certificate Manager (see “Logging In to the CMS Window” on page 351).

- In the navigation tree, select Users and Groups.
The Users tab appears in the right pane.



- Click Add.
The Select User Type window appears.



4. Select Trusted Manager and click OK.

The Edit User Information window appears.



5. Specify information as appropriate.

The information you enter here is to help you keep track of the clone Certificate Managers. The master Certificate Manager relies solely on its SSL server certificate (which you will add in Step 3) for authentication.

User ID. Type an ID that will help you identify this user in the list of privileged users. The ID can be an alphanumeric string of up to 255 characters.

Description. Type a description to identify that the user entry is for the clone Certificate Managers. The description can be an alphanumeric string of up to 255 characters.

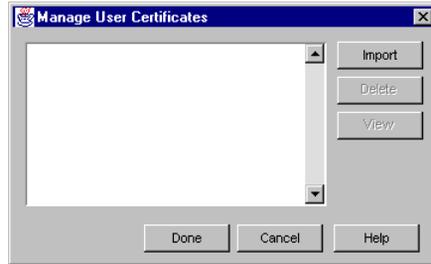
Group. Select Trusted Managers. For more information about this group, see “Group for Trusted Managers” on page 412.

6. Click OK.

You are returned to the Users tab. The user you just added is displayed in the list of users.

7. Select the user entry you just added for the clone Certificate Managers and click Certificates.

The Manage User Certificates window appears.

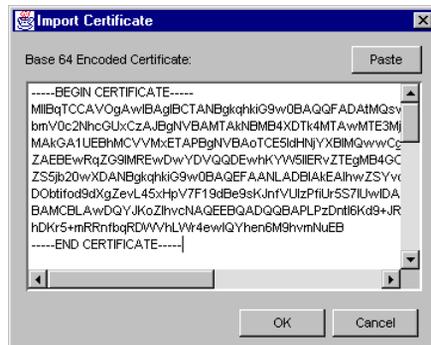


8. Click Import.

The Import Certificate window appears.

9. Click inside the text area, and paste the master Certificate Manager's SSL server certificate in its base-64 encoded form.

Be sure to include the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines.



10. Click OK.

You are returned to the Manage User Certificates window. The certificate you imported should now be listed in this window.

11. To view the certificate you imported, select it and click View.

The certificate information appears. Verify that the certificate you added is the correct one.

12. Click Done.

You are returned to the Users tab.

13. Click Refresh.

Step 9. Test Clone-Master Connection

To test whether your clone-master CA setup is complete and functional, repeat these steps for each clone Certificate Manager.

- Step A. Request a Certificate from the Clone CA
- Step B. Approve the Request
- Step C. Download the Certificate to the Browser
- Step D. Revoke the Certificate
- Step E. Check Master CA's CRL for the Revoked Certificate

Step A. Request a Certificate from the Clone CA

The steps outlined below explain how to request a client certificate from the clone Certificate Manager using the *manual* enrollment method. If you've configured the clone Certificate Manager for automated certificate issuance, for example for directory-based enrollment, you may use the appropriate form and request a certificate.

To request a client or personal certificate from the clone Certificate Manager:

1. Open a web browser window.
2. Go to the end-entity interface of the Certificate Manager you configured (or to the Registration Manager that's connected to this Certificate Manager).

The URL is in this form: `https://<hostname>:<end_entity_HTTPS_port>` or `http://<hostname>:<end_entity_HTTP_port>`

3. In the left frame, under Browser, click Manual.

This opens the manual enrollment form.

4. Fill in all the values and submit the request.

The client prompts you to enter the password for your key database.

5. When you enter the correct password, the client generates the key pairs.
Do not interrupt the key-generation process.

Step B. Approve the Request

Skip this step if you requested the certificate using any of the automated enrollment methods. Complete this step if you used the manual enrollment form for requesting the certificate; the request you submitted is waiting in the agent queue for approval by an agent.

To approve the request:

1. Go to the clone Certificate Manager's Agent Services interface.
The URL is in this format: `https://<hostname>:<agent_port>`
2. In the left frame, click List Requests.
3. In the form that appears, select the "Show pending requests" option and click Find.
4. In the list of pending requests, identify the request you submitted and click Details.
5. Check the request to make sure that it has all the required attributes of a client certificate.
6. Scroll to the bottom of the request form, and approve the request.

You should see a confirmation page indicating that the certificate has been issued. If you're using the same browser for requesting the certificate and for agent operations, don't close the page until after you complete the next step.

Step C. Download the Certificate to the Browser

If you're using the same browser, to download the certificate into the certificate database of the browser:

1. In the confirmation page, scroll down to the section that says "Installing this certificate in a client."
2. Check the certificate details for the required extensions.
3. Follow the on-screen instructions and download the certificate to your browser's certificate database.

If you're using a different browser, to download the certificate:

1. Go to the end-entity interface of the Certificate Manager that issued the certificate.
2. Select the Retrieval tab.
3. Search for the certificate; it will be the last certificate.
4. Click Details.
5. Scroll to down to the section that enables you to download the certificate to the browser, and download the certificate.

Step D. Revoke the Certificate

To revoke the certificate you issued:

1. Go to the end-entity interface for the Certificate Manager.
2. Select the Revocation tab.
3. In the left frame, click User Certificate.

The User Certificate Revocation form appears.

4. In the Revocation Reason section, select Unspecified and click Submit.

The browser shows the “Select a Certificate” dialog box and prompts you to choose the certificate you want to revoke.

5. Select the certificate you downloaded and click OK.

The clone Certificate Manager revokes the certificate, updates the certificate status in its internal database, and sends details about the revoked certificate to the master Certificate Manager.

Step E. Check Master CA's CRL for the Revoked Certificate

To verify that the revoked certificate has been included in the master Certificate Manager's CRL:

1. Go to the master Certificate Manager's Agent Services interface.

The URL is in this format: `https://<hostname>:<agent_port>`

2. In the left frame, click Update Certificate Revocation List.

3. In the form that appears, select the algorithm that you want to use to sign the new CRL.
 - MD5 with RSA generates a 128-bit message digest. Most existing software applications that handle certificates support only MD5. This is the default algorithm.
 - SHA-1 with RSA generates a 160-bit message digest. Before choosing SHA-1 with RSA, make sure your applications support it. Netscape Navigator 3.0 (or later) and Enterprise Server 2.01 (or later) support SHA-1.
 - SHA-1 with DSA generates a 160-bit message digest. Before choosing SHA-1 with DSA, make sure your applications support it. Communicator 4.0 (or later) and iPlanet server products with a version number greater than 4.0 support it.

Before selecting an algorithm, make sure that Certificate Manager has the algorithm enabled.

4. Click Display to examine the CRL (before updating it).

The CRL appears in the browser window. In the list, look for the certificate revoked by the clone Certificate Manager. If you don't see the certificate, check logs to resolve the problem.

5. If you want to update the CRL with the latest certificate revocation information, use the browser's Back button to return to the previous page and click Update.

Step 10. Use Master CA's Agent Certificate in Clone CAs

This step is optional.

The procedure below explains how to use the master Certificate Manager's agent certificate for a clone Certificate Manager (instead of creating a new agent certificates for clone CAs).

1. Go to the configuration directory of a cloned CA:
`<server_root>/cert-<instance_id>config`
2. Open the configuration file (`CMS.cfg`) in a text editor.
3. Locate this line: `agentgateway.enableAdminEnroll=true`

4. Change the value to `false`: `agentgateway.enableAdminEnroll=false`
This configures the cloned CA in to a mode where it expects a certificate (that was already issued and chains properly) to be presented when you access its agent interface.
5. Restart the clone CA.
6. Use iPlanet Console and open the CMS window for the clone CA instance.
7. Go to the “Users and Groups” section, create a new agent user, and add the master CA’s agent certificate to the clone CA’s certificate database.

To add the correct certificate, check the serial number of the master CA’s agent certificate; this certificate should already exist in one of the browsers that you use to access the master CA’s agent interface. Use the serial number to search for the certificate in the master CA’s certificate repository. Once you locate the certificate, look for its base-64 encoded form, copy it, and then paste it as the agent certificate in the clone CA.

For step-by-step instructions to create an agent user, see “Setting up Agents Using the Manual Process” on page 417.

8. After creating the agent entry for the clone CA, go to `https://<cloneCA hostname>:<agent_port>` to verify that you can access its agent interface successfully.
9. Repeat the above steps for other clone CAs.

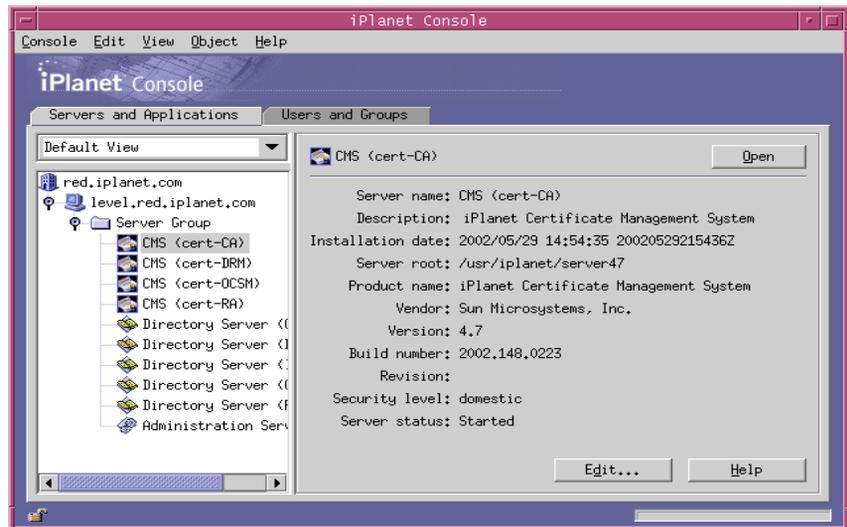
Viewing Instance Information

In iPlanet Console, you can view some of the basic information—the name and version number of the server, the directory in which it’s installed, and date it was installed—about a CMS instance.

To view information pertaining to a specific CMS instance:

1. Log in to iPlanet Console (see “Logging In to iPlanet Console” on page 344).
2. In the Console tab, double-click the server group that contains the CMS instance you want to view.

- In the list of server instances, select the CMS instance you want to view. The right pane shows information about the selected CMS instance.



The information displayed includes the following:

Server Name. A descriptive name of the CMS instance. You can change this name; see “Changing the Name of an Instance” on page 308).

Description. Additional information that helps you identify the CMS instance. You can change this description; see “Changing the Name of an Instance” on page 308.

Installation Date. The date the server was installed.

Server Root. The directory that holds all the files for the selected CMS instance, the files of its Administration Server, and the files of any other iPlanet servers in the same server group (that is, administered by that Administration Server). A host typically has only one server root, but more than one is possible, especially if different version numbers of the same server are installed on a single host.

The default server root in Unix is `usr/iplanet/servers/` and in Windows NT is `C:\iplanet\Servers`.

Product Name. The complete product name.

Vendor. The name of the vendor.

Version. The version number.

Build Number. The number that identifies the build that was used for this installation.

Security Level. The server's security level—whether the server is meant for use in the United States and Canada (domestic) or any other part of the world (export). (See “Configuring the Server's Security Preferences” on page 500.)

Server Status. The server's status—whether it is `started`, `stopped`, or `unknown`; normally, `unknown` indicates that the server hasn't been configured properly.

Changing the Name of an Instance

Following installation, the name of a CMS instance is in the form:

```
CMS (cert-<instance_id>)
```

`<instance_id>` is the ID for this instance of Certificate Management System. You first specified this when you installed this server.

For example, if you installed an instance of Certificate Management System with an ID of `testCA`, the instance name will be `cert-testCA`.

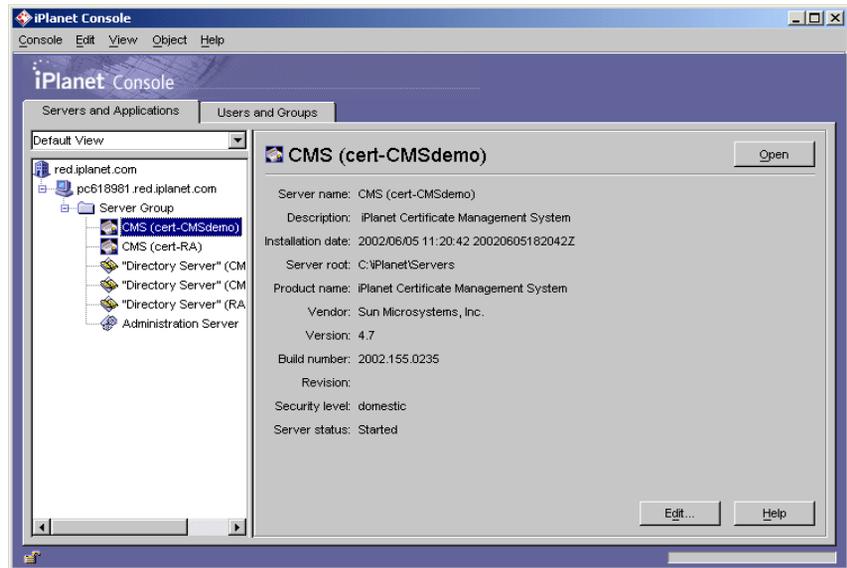
You can change the instance name to a more descriptive one later. If you change the instance name, Certificate Management System uses the new name as a descriptive nickname for the instance. It shows the new name in iPlanet Console only; it does not change the original instance ID in the configuration.

To change the name of a particular CMS instance:

1. Log in to iPlanet Console (see “Logging In to iPlanet Console” on page 344).
2. In the Console tab, select the CMS instance you want to rename.

3. Click Edit.

Details about the selected CMS instance appear in the right pane.



Specify the appropriate information:

Server Name. Type a descriptive name for the server.

Description. Type any additional description for the server. For example, you may want to type information that will help you identify this instance of Certificate Management System.

4. Click OK.

You are returned to the previous screen. The new name appears in the right pane.

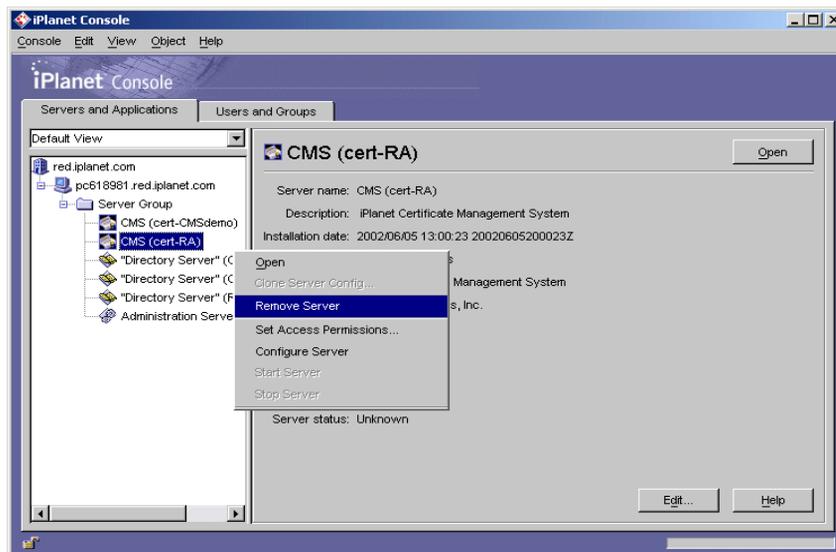
Removing an Instance From a System

If you are sure you won't need a particular CMS instance anymore, you can use iPlanet Console to remove the server instance from your machine. Removing a CMS instance is not the same as uninstalling Certificate Management System; when you uninstall Certificate Management System, its program files are deleted from the host machine. (For instructions, see "Uninstalling Certificate Management System" on page 311.)

To remove a CMS instance from your machine:

1. Log in to iPlanet Console (see “Logging In to iPlanet Console” on page 344).
2. In the Console tab, select the CMS instance you want to remove.
3. From the Object menu, choose Stop; you can also right-click to choose this option from the pop-up menu (see the figure below).
4. When the server has stopped, from the Object menu, choose Remove Server.

As shown in the figure below, you can also right-click to choose this option from the pop-up menu.



5. When prompted, confirm that you want to remove the server instance.

The selected CMS instance is removed. The corresponding internal database is not removed. If you want to remove it, select the instance, and repeat steps 3 through 5.

The Directory Server (configuration directory) and Administration Server binaries are also not removed; you require these to administer the remaining servers installed in the same server group.

Uninstalling Certificate Management System

To remove files pertaining to Certificate Management System from a host system, run the uninstallation program. Uninstalling Certificate Management System removes all the corresponding CMS instances from the navigation tree of iPlanet Console. To remove a specific CMS instance, follow the instructions provided in “Removing an Instance From a System” on page 309.

You can uninstall Certificate Management System in two ways:

- From the command line (locally only)
- On a Windows NT system, by using the Windows NT Add/Remove Programs Utility

Uninstalling From the Command Line

To uninstall Certificate Management System from the command line:

1. Open a terminal window to your server.
2. In a Unix system, log in either as `root` or using the server’s user account (if that is how you started the server).
3. At the command-line prompt, enter the following line:

On Windows NT, enter `<server_root>\uninst.`

On Unix, enter `<server_root>/uninstall.`

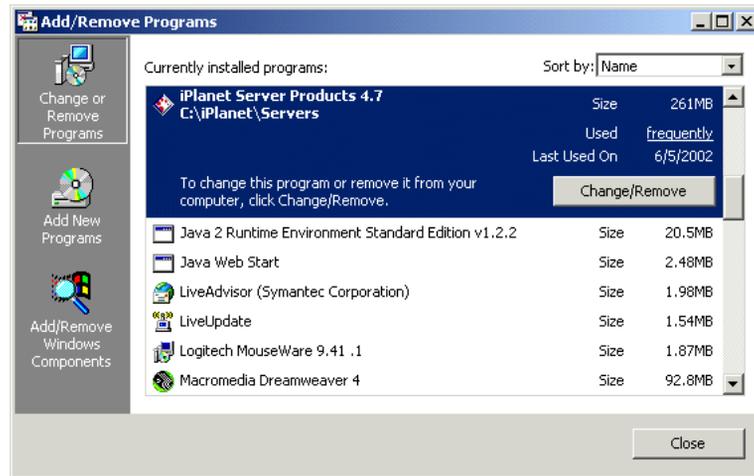
The uninstallation program starts.

Uninstalling by Using the Windows NT Add/Remove Programs Utility

To remove Certificate Management System by using the Windows NT Add/Remove Programs utility:

1. From the Start menu, choose Settings, then Control Panel.
2. In the Control Panel, choose Add/Remove Programs.

3. In the Add/Remove Programs Properties window, choose iPlanet Server Products 4.7 <server_root>, and click Add/Remove.



4. In the iPlanet Uninstall window, make sure all the components are selected, and click Uninstall.



The uninstallation program starts.

If you want to install a separate, stand-alone version of iPlanet Console for any reason, you can download it from this site:

<http://www.iplanet.com/downloads/patches/>

Upgrading From Version 4.2 SP2 to Version 4.7

The only direct migration path to Certificate Server 4.7 is from version 4.2, Service Pack 2 (SP2). If you have an existing installation of Certificate Management System version 4.2 or earlier, you must first upgrade to version 4.2SP2. Follow the instructions in the section “Upgrading to Version 4.2 SP2” on page 316.

If you already have an existing installation of Certificate Server 4.2 SP2, use the following instructions.

The CMS Migration Tool

Certificate Server 4.7 provides a utility that migrates certificates, keys, CRLs, and related user information contained in the Internal DB directories. The tool migrates only Certificate Server instances, and only on a single host; it does not span multiple machines. There are two versions of the migration utility, one for Unix and one for Windows. All steps listed in the migration tool documentation are performed.

Log files containing migration details can be found in the following directories:

Solaris:

```
/47_binaries_location/migration_MM-DD-YYYY-HH_MM_SS.log for
migration details
```

Windows:

```
\47_binaries_location\migration-MMDDYYYY.log
```

Known Issues and Workarounds

- If you're upgrading a Windows NT or 2000 installation:
 - If you use a third-party tool such as MKS toolkit, then the PATH should have the MKS toolkit as the last entry. Otherwise, the perl script will fail to execute.
 - During the migration and after Certificate Server 4.2SP2 is uninstalled, if this message displays: "There are files marked for deletion upon next Reboot, Do you wish to Reboot now?" do not reboot; enter "No."

Before You Begin

You should address the following issues before running the migration tool:

- Migration to Certificate Server 4.7 can be performed only on the same machine where Certificate Server 4.2SP2 is installed.
- Back up the your CMS 4.2SP2 installation in case you need to recover your data.
- Gather information about the servers running on the system. For example, you should know the server names, ports, and plug-ins being used, and so on.
- Know the passwords for the iPlanet Console administrator, the Certificate Server administrator, the LDAP data bases, the single sign-on for each instance.
- Close the 4.2SP2 Certificate Server window, and close any instances of iPlanet servers before starting the migration process.
- Process all 4.2SP2 pending certificate requests.
- Close all sessions to Certificate Server 4.2SP2. Do not access Certificate Server until after the migration process is completed.
- On Unix, you should run the migration tool as the same user who installed CMS 4.2SP2.
- Know the location of the unzipped Certificate Server 4.7 binaries.
- If a Certificate Server instance has been deleted from the installation, then delete its corresponding internal database instance.
- The migration tool needs ample space to backup the Certificate Server instances. Be sure that there is sufficient disk space in /var/tmp and the new 4.7 installation directory for Unix/Solaris, or in the default system drive for Windows NT/2000.
- On Windows only:
 - Prior to running the migration tool, you should install PERL on a directory other than the one that contains the PERL that comes with Certificate Server 4.2SP2. If the PATH to PERL is set to the PERL that comes with Certificate Server 4.2SP2, the PATH will be invalid after uninstallation.
 - If possible, take a system image of your machine for recovery purposes.
 - You should run the migration tool as Administrator.

Running the Migration Tool on Unix

The Unix version is a bourne shell script and is supported on Solaris.

1. Identify the Certificate Server 4.2SP2 instance that you want to upgrade and note the corresponding server root and instance ID.
2. Extract files from the Certificate Server archive; you can get the archive from the product CD or from the iPlanet download site (at <http://www.sun.com/software/download/>).
3. In the list of extracted files, locate this file: `/dist/MigrationSolaris`.
4. Run the following command:


```
cd <extracted root>/dist
```
5. Run the migration tool:


```
./MigrationSolaris
```
6. The script prompts you to provide the following information:
 - a. **Installed location of CMS 4.2 SP2:**
 - b. **Do you want the Migration script to run the cmsbackup tool for each CMS instance?**
 - c. **Install location for CMS 4.7:**
 - d. **Location of extracted CMS 4.7 distribution:**
 - e. **Please make sure that at least <X> space is available in the respective partitions. Continue?**
 - f. **Please verify the Admin password for the Administration Server:**
 - g. **Please enter the password for the configuration Directory Manager:**
 - h. **Please enter the Single Sign-On password:**
7. After the script has finished running, verify that the new installation works and that your data has been successfully migrated.
8. Manually uninstall Certificate Server 4.2 SP2.

Running the Migration Tool on Windows

This is PERL script using PERL 5.005 or higher.

1. Identify the Certificate Server 4.2SP2 instance that you want to upgrade and note the corresponding server root and instance ID.

2. Extract files from the Certificate Server archive; you can get the archive from the product CD or from the iPlanet download site (at <http://www.sun.com/software/download/>).
3. In the list of extracted files, locate this file: `MigrationNt.pl`
4. Run the migration tool:

```
MigrationNt.pl
```

You can invoke the script with the `-v` option to see debug messages.
5. The script will prompt you to provide the following information:
 - a. **The absolute pathname of the 4.2 SP2 Certificate Server Root Directory:**
 - b. **The absolute pathname for the new 4.7 Certificate Server Root Directory:**
 - c. **The absolute pathname of the CMS 4.7 binaries:**
 - d. **Please verify the Admin password for the Administration Server:**
 - e. **What is the Directory Manager Password?**
 - f. **Do you want to delete the temporary backup files?**
6. After the script has completed, reboot the computer system.
7. Verify that the new installation works and that your data has been successfully migrated.

The migration tool for Windows automatically uninstalls the 4.2SP2 installation. It is a good practice to check the old installation directory and to delete any remaining files.

Upgrading to Version 4.2 SP2

If you have an existing installation of Certificate Management System version 4.2, you can upgrade to Certificate Management System 4.2, Service Pack 2 (SP2), by installing CMS 4.2-SP2 into the same server root as that of CMS 4.2. When prompted to specify the instance name, you must enter the name of the CMS instance that you want to upgrade. If the specified CMS instance exists in the selected server root, the installation program recognizes the instance and updates the existing configuration automatically. Note that during installation, you must

specify the same port numbers that are in use by existing services, such as the Administration Server port for iPlanet Console. If you have multiple CMS instances under the same server root, you must run the installation program for each instance.

Note the following:

- Prior to attempting any upgrade from CMS 4.2 to CMS 4.2-SP2, always remember to shutdown all instances of the Console, as these clients are never shutdown by the upgrade process, and therefore, may not be correctly updated.
- Only the CMS instance you specify in the upgrade panel is updated, although all of the global files are restored upon updating each and every instance. Therefore, you must perform the upgrade process for each and every CMS instance individually.
- All CMS instances are shutdown each time so that any global CMS information may be updated.
- Backup copies of the following list of original files and directories are saved with an underscore and timestamp appended (for example, `classes/` becomes `classes_20000506035603Z/`) before the new file or directory replaces them:
 - Instance Specific:
 - `classes/` (directory)
 - `emails/` (directory)
 - `restart-cert[.bat]` (file)
 - `start-cert[.bat]` (file)
 - `stop-cert[.bat]` (file)
 - `web/` (directory)
 - Globally Specific: (backed up for each instance)
 - `bin/cert/classes/` (directory)
- Since the start and restart scripts are always replaced on a per instance basis, any changes you have made will be backed up, and you must manually edit these scripts to put back your customizations.
- All CMS instances are restarted. During restart, you are prompted for the single sign-on password, if automatic restart has not already been configured/reconfigured ahead of time.
- There should be no reason to configure an updated instance (by running the CMS Installation Wizard), as there is on a first-time installation.
- You may upgrade an instance numerous times to restore any corrupted global binaries. However, backup copies of the files and directories detailed above will be created upon each update.

- Occasionally, upgrade does not allow you to specify the port number for the console to be reused. It is okay to use a different port number for the console, however, you must remember to use this new port number the next time that you login to iPlanet Console.
- Since upgrades must be done on a per instance basis, if you have two instances, and if you upgrade the first and then launch the console, you may notice that the instance that has not been upgraded will be displayed out of alignment within the GUI. You can consider this as a visual clue that you need to upgrade this instance as well. After upgrading this, the instance will appear aligned correctly within the GUI.

Follow the procedure below to upgrade a CMS instance:

1. Identify the CMS 4.2 instance that you want to upgrade and note the corresponding server root and instance ID.
2. Prior to attempting any upgrade, shutdown all instances of the Console, as these clients are never shutdown by the upgrade process, and therefore, may not be correctly updated.
3. Extract files from the CMS archive; you can get the archive from the product CD or from the iPlanet download site (at <http://www.sun.com>).
4. In the list of extracted files, locate this file: `ns-certsrv4_2SP2.conf`
5. Go to the file structure of the CMS instance that you want upgrade, and navigate to the configuration directory of the internal database (the Directory Server instance used for storing CMS data):
`<server_root>/slapd-<cms_instance_id>-db/config`
6. Copy the `ns-certsrv4_2SP2.conf` file to this directory.
7. Run the `setup` program.

When you run the installation program for upgrading a CMS 4.2 instance, you will be presented with a series of screens or panels; the example below lists the panels on UNIX. Follow the on-screen prompts and complete the upgrade process.

- o Welcome
- o License
- o Product Selection

- Location (You must specify the same server root or installation directory that was used for your 4.2 Console.)
 - Server Products Components
 - Core Components
 - Directory Suite Components
 - Administration Service Components
 - Certificate Server Components
- Fully qualified domain name of machine
- User and Group
 - System User
 - System Group
- Configuration Directory Server Administration Identifier
 - Administrator ID
 - Password
- Administration Server Port for Console (You must enter the same port number that was used for your 4.2 Console.)
 - Administration Server User
 - Certificate Server Identifier (The CMS instance name that you enter in this panel must exist.)

Starting and Stopping CMS Instances

This chapter describes how to start, stop, and restart iPlanet Certificate Management Server (CMS) and how to check its current status. The chapter also explains the CMS watchdog process, a native bootstrapping program that enables Certificate Management System to start up with a single password instead of multiple ones.

The chapter has the following sections:

- Starting Certificate Management System (page 322)
- Stopping Certificate Management System (page 330)
- Restarting Certificate Management System (page 332)
- Checking System Status (page 334)
- Attending to an Unresponsive Server (page 335)
- CMS Watchdog Process (page 335)
- Password-Quality Checker (page 337)

NOTE You can use the CMS window only when the appropriate Administration Server is running. Be sure to start Administration Server at the port you specified during CMS installation. To minimize security risks, shut down Administration Server when you have finished using iPlanet Console. For instructions on starting and shutting down Administration Server, see “iPlanet Administration Server” on page 342.

Starting Certificate Management System

Once Certificate Management System is installed, it runs constantly, listening for and accepting requests. You can start Certificate Management System in several ways:

- From iPlanet Console (locally and remotely)
- From the command line (locally only)
- On a Windows NT system, from the Windows NT Services panel

Required Start-up Information

When you start Certificate Management System, you are prompted to enter the *single sign-on* password you specified during installation. This password enables the CMS watchdog (see “CMS Watchdog Process” on page 335) to retrieve all the passwords required by the server to start. These include the following:

- Passwords for the internal or external (if any are currently installed) tokens; these tokens contain certificates and corresponding public and private key pairs for the server.
- The bind password used by Certificate Management System to access and update the internal database.
- The bind password used by Certificate Management System to access and remove PINs from the authentication directory, if you’ve configured Certificate Management System to remove PINs from the authentication directory. (See the description for the `ldap.ldapauthbindDN` and `ldap.ldapauth.bindPWPrompt` parameters of the `UidPwdPinDirAuth` plug-in module, which explained in the *CMS Plug-Ins Guide*).
- The bind password used by Certificate Management System to access and create/modify user entries in the directory used for portal registration, if you’ve configured Certificate Management System for portal enrollment. (See the description for the `ldap.ldapauthbindDN` and `ldap.ldapauth.bindPWPrompt` parameters of the `PortalEnroll` plug-in module, which explained in the *CMS Plug-Ins Guide*).
- The bind password used by Certificate Management System to access and update the LDAP directory; this is required only if you have configured Certificate Management System for publishing certificates and CRLs to an LDAP-compliant directory.

You first specified these passwords when you installed Certificate Management System. Keep in mind that the passwords you provide for the tokens unlock a combination of the following private keys:

- If you have installed a Certificate Manager in the currently selected CMS instance, the token password unlocks the private keys for the Certificate Manager's *CA signing* and *SSL server* certificates.
- If you have installed a Registration Manager in the currently selected CMS instance, the token password unlocks the private keys for the Registration Manager's *signing* and *SSL server* certificates.
- If you have installed a Data Recovery Manager in the currently selected CMS instance, the token password unlocks the private keys for the Data Recovery Manager's *storage* keys and *transport* and *SSL server* certificates.
- If you have installed an Online Certificate Status Manager in the currently selected CMS instance, the token password unlocks the private keys for the Online Certificate Status Manager's *signing* and *SSL server* certificates.

For more information about the CMS keys and certificates, see Chapter 14, “Managing CMS Keys and Certificates.”

Note that during CMS installation, the watchdog stores all the passwords, required by the server for starting up, in a password cache. The cache is maintained in a file encrypted using the single sign-on password you specify during installation. When you change any of the required passwords or provide new passwords, you must start the server from the command-line (see “Starting From the Command Line” on page 328) so that the watchdog can prompt you for the new passwords in order to update the cache.

The single sign-on password eliminates the need for you to enter the various password when starting up Certificate Management System. As a security measure, you should consider changing the single sign-on password periodically. For instructions, see “Password Cache” on page 336.

Also note that all passwords used in Certificate Management System are checked by a built-in password-quality checker; for details, see “Password-Quality Checker” on page 337.

Configuring the Server to Start Without the Single Sign-On Password

If you prefer to start up Certificate Management System by entering all the required passwords, instead of just the single sign-on password, you can do so by either deleting or renaming the password cache file, `pwcache.p12` (notice the `.p12` extension).

Here's how you can do it:

1. Go to this directory: `<server_root>/cert-<instance_id>/config`
2. Locate the `pwcache.p12` file.
3. Either rename or delete the file.
4. Start the server from the command line; see “Starting From the Command Line” on page 328.

You are prompted for all the required passwords.

Later, if you want to revert back to starting the server using the single sign-on password:

1. Create a password cache and then create entries for all the required passwords.

For instructions on creating a new password cache and adding new entries to the password cache, see Chapter 2, “Password Cache Utility” of *CMS Command-Line Tools Guide*.

2. Copy the file to the `<server_root>/cert-<instance_id>/config` directory.
3. Start the server from the command line; see “Starting From the Command Line” on page 328.

You are prompted for the single sign-on password.

Configuring the Server to Read the Single Sign-on Password From a File

Every time you start Certificate Management System, you are required to enter either the single sign-on password or all the passwords required by the server to startup (see “Required Start-up Information” on page 322 and “Configuring the Server to Start Without the Single Sign-On Password” on page 323). By default, there is no way to start Certificate Management System non-interactively; the `startcert` script requires you to enter a password.

If it is inconvenient for you to start the server this way, you can store the single sign-on password in a file and edit the `startcert` script to use this file to start without any prompts. Configuring the server this way eliminates the need for you to enter the single sign-on password every time you start the server—it can be useful on UNIX machines to have the server automatically started if the machine reboots for any reason, and it can be useful on Windows NT to be able to start the server without having to be at the system console. Note that there is no way to make the edited `startcert` script a service that is automatically started by Windows NT at startup, so if the machine reboots due to a power failure, Certificate Management System will not start automatically.

CAUTION The instructions that follow explain how to configure Certificate Management System to start by reading the single sign-on password from a file. Note that the password is stored in a plain text file and you must use your operating system's security feature to secure this file. Failing to do so poses a security risk, as anyone who has access to the host system will be able to get hold of the single sign-on password.

To configure the server to start by reading the single sign-on password from a file:

1. Create a file named `pwfile`.
2. Put the single sign-on password in the file.
3. Copy the file to this directory: `<server_root>/cert-<instance_id>/config`
4. Edit the `start-cert` script.

To edit the `start-cert` script in Unix, follow these steps:

- a. Open a command-line window.
- b. Go to the CMS-instance directory.

For example, `/usr/iplanet/servers/cert-testCA`.

- c. Enter the following line at the prompt:

```
cat start-cert
```

You should see something similar to this:

```
#!/bin/sh

/usr/iplanet/servers/bin/cert/admin/bin/start -i testCA
-r /usr/iplanet/servers -e -classpath

/usr/iplanet/servers/bin/cert/classes:/usr/iplanet/
servers/bin/cert/jars/jss.jar:/usr/iplanet/servers/bin/
cert/jars/certsrv.jar:/usr/iplanet/servers/java/
ldapjdk.jar:/usr/iplanet/servers/bin/cert/jre/lib/
rt.jar:/usr/iplanet/servers/bin/cert/jre/lib/i18n.jar:/
usr/iplanet/servers/bin/cert/jars/jssjdk12.jar
```

- d. Edit the script to include the file path to the `pwfile` file: to the end of the line that begins `$NETSITE_ROOT/bin/cert/admin/bin/cert add -f $NETSITE_ROOT/cert-<instance_id>/config/pwfile`.

Be sure to include the file path as shown (in bold) in the example.

To edit the `start-cert.bat` script in Windows NT, follow these steps:

```
#!/bin/sh

/usr/iplanet/servers/bin/cert/admin/bin/start -i testCA
-f /usr/iplanet/servers/cert-testCA/config/pwfile -r
/usr/iplanet/servers -e -classpath

/usr/iplanet/servers/cert-testCA/classes/:/usr/iplanet/server
s/bin/cert/classes/:/usr/iplanet/servers/bin/cert/jars/jss.ja
r:/usr/iplanet/servers/bin/cert/jars/certsrv.jar:/usr/iplanet
/servers/java/ldapjdk.jar:/usr/iplanet/servers/bin/cert/jre/l
ib/rt.jar:/usr/iplanet/servers/bin/cert/jre/lib/il8n.jar:/usr
/iplanet/servers/bin/cert/jars/jssjdk12.jar
```

- a. Open a command-line window.
b. Go to the CMS instance directory.

For example, `C:\iPlanet\server4\cert-testCA`.

- c. Enter the following line at the prompt:

```
type start-cert.bat
```

You should see something similar to this:

- d. `iplanetervers` Edit the script to include the file path to the `pwfile` file: to the end of the line that begins `net start cert-<instance_id> add /fc:%NETSITE_ROOT%\cert-<instance_id>\config\pwfile`.

Be sure to include the file path as shown in the example (shown in bold).

```
net start cert-testCA
/fc:\iplanet\servers\cert-testCA\config\pwfile/c:\iplanet\serve
rs
\cert-testCA\classes\;C:\iplanet\servers\bin\cert\classes\;C:
\iplanet\servers\bin\cert\jars\jss.jar;C:\iplanet\servers\bin
\cert\jars\certsrv.jar;C:\iplanet\servers\java\ldapjdk.jar;C:
\iplanet\servers\bin\cert\jre\lib\rt.jar;C:\iplanet\servers\b
in\cert\jre\lib\il8n.jar;C:\iplanet\server\bin\cert\jars\jssj
dk12.jar;C:\iplanet\servers\java\swingall.jar
```

- e. Save your changes.
5. Use your operating system's security feature to restrict access to the password file.
6. Restart the server from the command line; see "Starting From the Command Line" on page 328.

It should start without prompting for the single sign-on password.

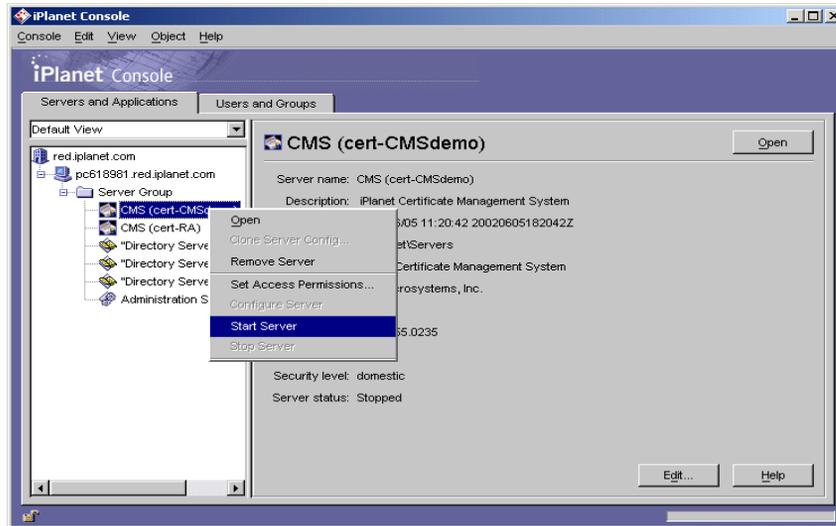
Starting From iPlanet Console

You can use iPlanet Console to start an instance of Certificate Management System running on a local or remote host.

To start Certificate Management System from iPlanet Console:

1. Log in to iPlanet Console (see "Logging In to iPlanet Console" on page 344).
2. In the Console tab, select the Server Group that contains the CMS instance you want to start.
3. In the navigation tree, locate the CMS instance you want to start.

4. Select the instance, right-click, and select the Start Server option from the pop-up menu.



When you start Certificate Management System, you are prompted to supply the single sign-on password for the server.



5. Type the single sign-on password you specified during installation and click OK.

Certificate Management System won't start until you provide this password. For more information, see "Required Start-up Information" on page 322.

Starting From the Command Line

To start Certificate Management System from the command line:

1. Open a terminal window to your server.
2. In a Unix system, log in as `root` if the server runs on ports less than 1024; otherwise, log in either as `root` or with the server's user account.

3. At the command-line prompt, enter the following line:

```
<server_root>/cert-<instance_id>/start-cert[.bat]
```

.bat specifies the file extension; this is required only when running the utility on a Windows NT system.

<server_root> is the directory where the CMS binaries are kept. You first specified this directory during installation.

<instance_id> is the ID for this instance of Certificate Management System. You first specified this when you installed this server.

4. When prompted, enter the single sign-on password.

Certificate Management System won't start until you provide this password. For more information, see "Required Start-up Information" on page 322.

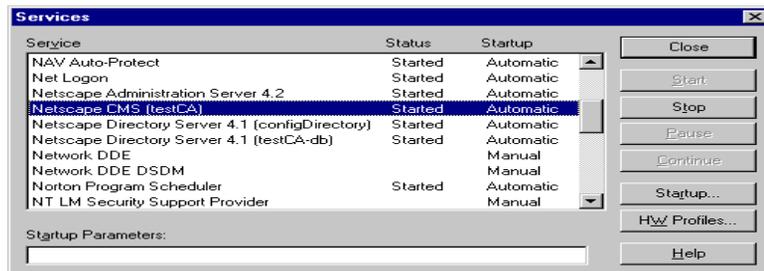
NOTE If Certificate Management System is already running, the start-up command fails. Stop the server first using the `stop-cert` command, then use the `start-cert` command.

Starting From the Windows NT Services Panel

If you have installed Certificate Management System on a Windows NT system, you can start the server (as a service) from the Windows NT Services panel (see Figure 8-1). The CMS service has the following name:

```
iPlanet CMS (<instance_id>)
```

Figure 8-1 CMS service in the Windows NT Services panel



To start Certificate Management System from the Windows NT Services panel:

1. Click the Start button on your desktop.
2. Select Control Panel from Settings.
3. In the Control Panel window that appears, click Services.
4. Select the CMS instance and click Start.

You are prompted to supply the single sign-on password for the server.

5. Enter the single sign-on password you specified during installation and click OK.

Certificate Management System won't start until you provide this password. For more information, see "Required Start-up Information" on page 322.

Stopping Certificate Management System

You can stop Certificate Management System in several ways:

- From iPlanet Console (locally and remotely)
- From the command line (locally only)
- On a Windows NT system, from the Windows NT Services panel

Stopping Certificate Management System shuts down all the subsystems completely, interrupting service until the server is started again. If your machine crashes or is taken offline, the server stops, and any requests it was servicing are lost. You need to start the server again to restore service.

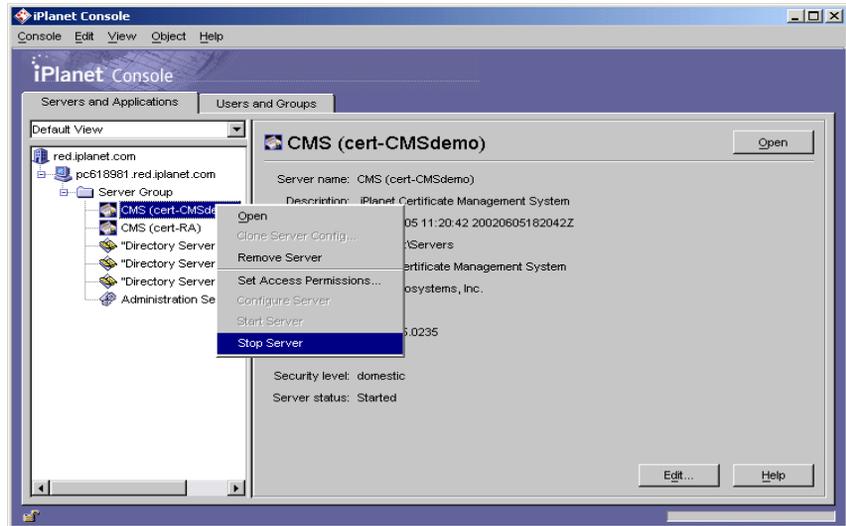
Stopping From iPlanet Console

You can use iPlanet Console to stop an instance of Certificate Management System running on a local or remote host.

To stop Certificate Management System from iPlanet Console:

1. Log in to iPlanet Console (see "Logging In to iPlanet Console" on page 344).
2. In the Console tab, select the Server Group that contains the CMS instance you want to stop.

3. In the navigation tree, select the CMS instance you want to stop, right-click, and select the Stop Server option from the pop-up menu.



The server is stopped.

Stopping From the Command Line

You can stop a CMS instance running on a local host by entering the appropriate command at the command prompt.

To stop a Certificate Management System from the command line:

1. Open a terminal window to your server.
2. In a Unix system, log in either as `root` or using the server's user account (if that is how you started the server).
3. At the command-line prompt, enter the following line:

```
<server_root>/cert-<instance_id>/stop-cert[.bat]
```

`.bat` specifies the file extension; this is required only when running the utility on a Windows NT system.

`<server_root>` is the directory where the CMS binaries are kept. You first specified this directory during installation.

`<instance_id>` is the ID for this instance of Certificate Management System. You first specified this when you installed this server.

Stopping From the Windows NT Services Panel

You can stop a CMS instance running on a local host by stopping the corresponding service; it is identified by the following in the Windows NT Services panel (see Figure 8-1 on page 329):

```
iPlanet Certificate Management Server (cert-<instance_id>)
```

To stop Certificate Management System from the Windows NT Services panel:

1. Click the Start button on your desktop.
2. Select Control Panel from Settings.
3. In the Control Panel window that appears, click Services.
4. Select the CMS instance and click Stop.
5. When prompted, click Yes.

The server is stopped.

Restarting Certificate Management System

Whenever you change the CMS configuration, you must save your changes (by clicking the Save button) for the changes to take effect. Some configuration changes also require that you *restart* the server after you save the changes. If restarting is required, the server prompts you accordingly.

You can restart the server in two ways:

- From the CMS window (locally and remotely)
- From the command line (locally only)

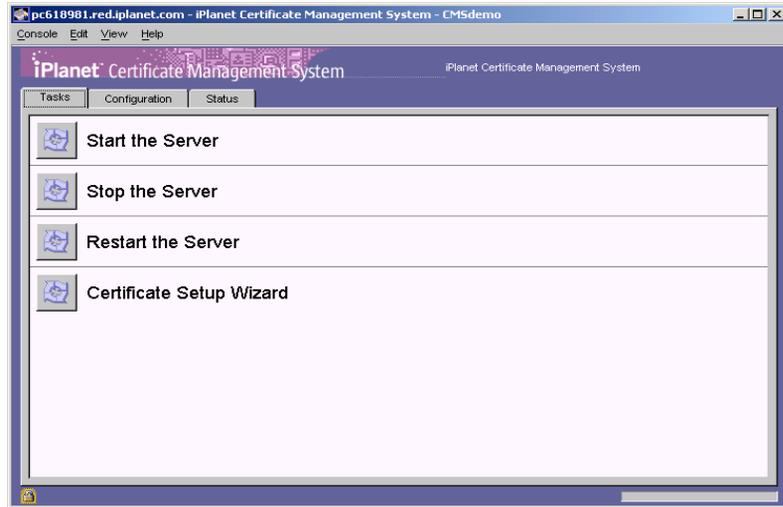
Restarting From the CMS Window

You can use the CMS window to restart an instance of Certificate Management System on a local or remote host.

To restart Certificate Management System from the CMS window:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).

2. In the Tasks tab, click Restart the Server.



When you restart Certificate Management System, you are prompted to supply the single sign-on password for the server.



3. Type the single sign-on password you specified during installation and click OK.

Certificate Management System won't restart until you provide this password. For more information, see "Required Start-up Information" on page 322.

Restarting From the Command Line

To restart Certificate Management System from the command line:

1. Open a terminal window to your server.
2. In a Unix system, log in either as `root` or using the server's user account (if that is how you started the server).

3. At the command-line prompt, enter the following line:

```
<server_root>/cert-<instance_id>/restart-cert[.bat]
```

.bat specifies the file extension; this is required only when running the utility on a Windows NT system.

<server_root> is the directory where the CMS binaries are kept. You first specified this directory during installation.

<instance_id> is the ID for this instance of Certificate Management System. You first specified this when you installed this server.

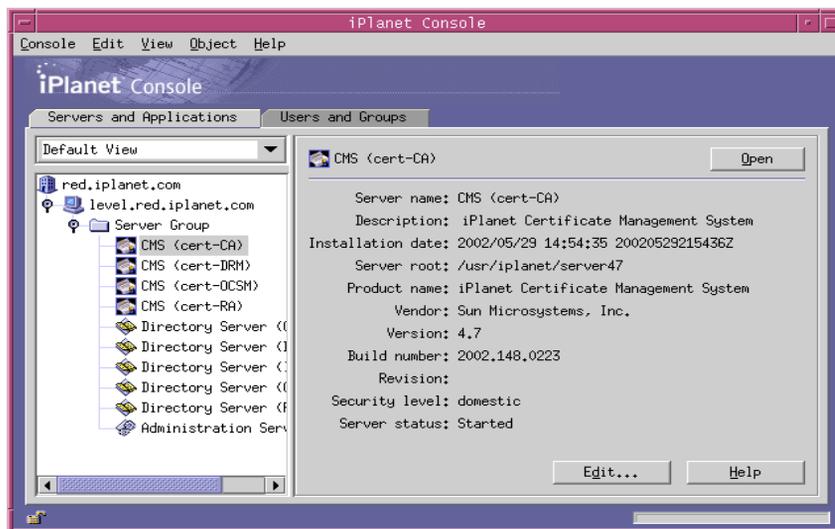
4. When prompted, enter the single sign-on password.

Certificate Management System won't restart until you provide this password. For more information, see "Required Start-up Information" on page 322.

Checking System Status

You can use iPlanet Console to find out whether a particular instance of Certificate Management System is running.

1. Log in to iPlanet Console (see "Logging In to iPlanet Console" on page 344).
2. In the Console tab, select the instance that corresponds to the CMS instance you want to check.



3. In the right pane, check the Server Status field.

If the selected instance of Certificate Management System is running, the status will be *Started*. Otherwise it will be *Stopped* or *Unknown*.

Attending to an Unresponsive Server

If an error causes Certificate Management System to become unresponsive, and all attempts to stop it from iPlanet Console fail, it may be necessary to kill the server processes manually. For this purpose, Certificate Management System provides a command-line tool `killproc`, which is explained in *CMS Command-Line Tools Guide*.

CMS Watchdog Process

The CMS watchdog is a native bootstrapping program that provides specific native functions. It works with Certificate Management System to enable it to start up using a single password—instead of multiple passwords—called the *single sign-on* password. In addition, it manages the start-up, stop, and restart states of Certificate Management System.

The watchdog process (identified as `cms_watchdog`) implements the following operations:

- Starts Certificate Management System and the Virtual Java Machine, or Java VM—the watchdog allows you to start Certificate Management System by using a single password instead of the multiple passwords that would otherwise be required. For details, see “Required Start-up Information” on page 322. (During CMS installation the watchdog stores all the passwords required by the server for starting up in a password cache, which is explained in “Password Cache” on page 336.)
- Stops Certificate Management System.
- Restarts Certificate Management System (after configuration changes).
- Detects Certificate Management System crashes and restarts the server—the watchdog monitors Certificate Management System and the Java VM, restarting the server in the case of a failure.
- In the Unix version of Certificate Management System, the watchdog records the server process ID (`pid`) and sets the user ID (`uid`) of the process.

Password Cache

During CMS installation, the installation program creates a password cache which the CMS watchdog uses to store all the passwords required by the server during start up (see “Required Start-up Information” on page 322). For example, when you specify the cryptographic token password and the bind password for the internal directory during installation, the watchdog adds these passwords into the password cache; similarly, when you configure the server for LDAP publishing from iPlanet Console, the watchdog adds the corresponding password to the cache.

The password cache is maintained in a triple-DES encrypted file named `pwcache.p12`, which is located here:

```
<server_root>/cert-<instance_id>/config
```

The file is protected using the single sign-on password you specify during installation. In the cache, passwords are stored along with a name, a string describing the usage of the password, which is used by Certificate Management System to index into the cache. For example, the contents of the password cache could look like this:

```
----- Password Cache -----
Internal LDAP Database : myIdbPwd
Internal Key Storage Token : myTokenPwd
Authentication : myPinAuthPwd
LDAP Publishing : myLdapPubPwd
```

Note that in the above example

- The string `Internal LDAP Database` is the default value assigned to the `internaldb.ldapauth.bindPWPrompt` parameter in the CMS configuration file; it provides a descriptive usage for the password Certificate Management System uses to bind to the internal database.
- The string `Internal Key Storage Token` is hardcoded and it refers to the Netscape Software Cryptographic Service provider; you cannot change it. You can only change the corresponding password.

Other entries may appear in the password cache. For example, if you set up PIN-based authentication with the remove PIN option, you will see an entry for the password Certificate Management System uses to bind to the authentication directory to remove a PIN after a user successfully authenticates; for details, see `UidPwdPinDirAuth` plug-in module in *CMS Plug-Ins Guide*. Similarly, if you enable LDAP publishing with *basic authentication*, you will also see an entry for the password Certificate Management System will use to bind to the publishing directory; for details, see “Step 5. Identify the Publishing Directory” on page 680.

Except for the string `Internal LDAP Database`, you can change any of the above prompts by modifying the corresponding value in the configuration file and then replacing (delete the old item and add the new item) the current entry in the password cache with the new prompt and the password using the `PasswordCache` utility explained in the *CMS Command-Line Tools Guide*.

When various modules in the server, such as authentication and LDAP publishing, initialize, they query the password cache for the password. The password cache returns the password if it has it, or else it prompts the user for one. Note that this prompting happens only at server startup time, which means whenever you change any of the required passwords or provide new passwords, you must restart the server from the command-line (see “Starting From the Command Line” on page 328) so that the watchdog can prompt you for the new passwords in order to update the cache.

Password-Quality Checker

Certificate Management System comes with a plugin, called *password-quality checker*, to monitor the quality of passwords set within the CMS system. All passwords used in Certificate Management System are checked by the password-quality checker, which by default checks that the length of a password is at least 8 characters long; there are no checks regarding which characters are valid or invalid. If you use a password that doesn't meet the quality rules, you will get an error message indicating that the password didn't meet the password-quality rules.

Note that Certificate Management System enforces password quality on only those passwords that it strictly creates and manages. Passwords you enter for LDAP directory access are not subjected to quality checks. The reason for this is, the password quality is handled by the system that creates and manages the password. In an LDAP directory access, the remote directory that you authenticate to enforces the password quality of the password you use because it is created and managed by the directory.

To enable you to customize password quality, the plugin for the password-quality checker is included in the CMS samples package; for example, you can change the default rule to ensure that all CMS passwords are constructed with certain types of characters such as numbers, symbols, capital letters, and so on. The samples package is located here: `<server_root>/cms_sdk/cms_jdk/samples`

Administration Tasks and Tools

In administering iPlanet Certificate Management Server (CMS), you perform server-specific tasks such as starting, stopping, and restarting the server; changing configuration; configuring certificate issuance and management policies; adding or modifying privileged-user and group information; setting up authentication mechanisms for users who may request services from the server; performing routine server maintenance tasks; monitoring logs; and backing up server data.

To enable system administrators to accomplish these server-specific tasks quickly and easily, Certificate Management System provides a GUI-based administration tool, called the CMS window, within iPlanet Console. This chapter provides an overview of both iPlanet Console and the CMS window.

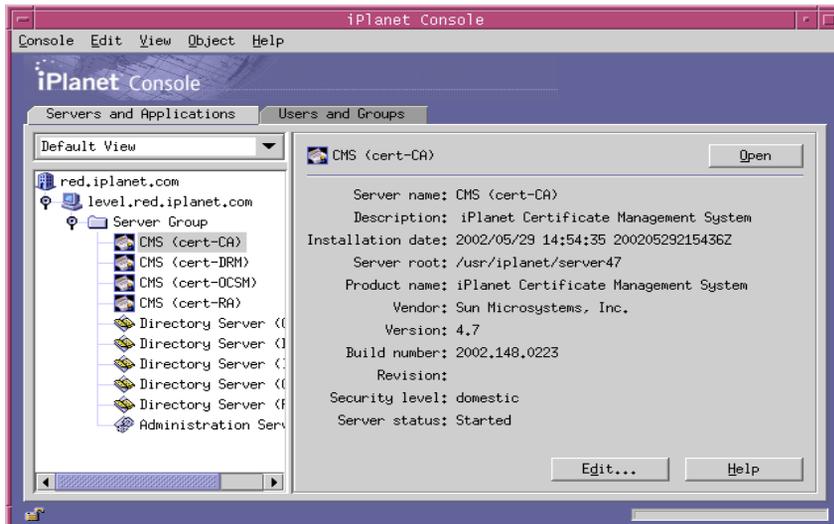
- iPlanet Console (page 340)
- Logging In to iPlanet Console (page 344)
- The CMS Window (page 346)
- Logging In to the CMS Window (page 351)

NOTE You can use iPlanet Console for managing various network resources. However, this chapter's focus is on using iPlanet Console for CMS administration. For complete information about iPlanet Console, see *Managing Servers with iPlanet Console*, which is included with the CMS documentation.

iPlanet Console

iPlanet Console is a stand-alone Java application that provides a GUI-based front end to all network resources registered in an organization's configuration directory. This unified administration interface (shown in Figure 9-1) simplifies network administration by supplying access points to all iPlanet version 4.x server instances installed across a network. Similarly, it simplifies basic user and group management by providing a unified administration interface to the user directory.

Figure 9-1 iPlanet Console window, with a CMS instance selected in the Console tab



Console Tab

For any given instance of iPlanet Console, the limits of the network it can administer are defined by the set of resources whose configuration information is stored in the same configuration directory—that is, the maximum set of hosts and servers that can be monitored from iPlanet Console. The *superadministrator* (the person who manages the configuration directory) can set access permissions on all network resources registered in the configuration directory. Thus, for a given administrator using iPlanet Console, the actual number of visible servers and hosts may be fewer, depending on the access permissions that the administrator has.

The Console tab displays all servers registered in a particular configuration directory, giving you a consolidated view of all the server software and resources under your control. What you control is determined by the access permissions the superadministrator has set up for you.

From this view you can perform tasks across arbitrary groups or a cluster of servers in a single operation. In other words, you can use the Console tab to manage a single server or multiple servers that are installed on different ports on one machine. Also, you can access individual server windows (or administration interfaces) by double-clicking the icons for the corresponding server instance entries (SIEs).

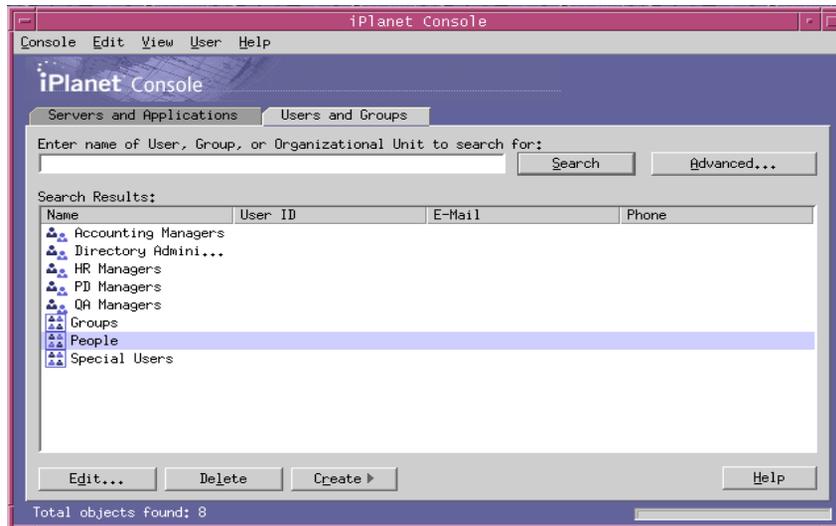
With the exception of Certificate Management System, all server instances displayed on the Console tab store their configuration information in the same configuration directory. For security purposes, Certificate Management System uses file-based configuration which is stored locally on the host system; during installation, the server registers only its SIE in the configuration directory. For details about this file, see “CMS Configuration” on page 355.

You can accomplish various CMS-specific tasks from the Console tab:

- Install multiple instances of Certificate Management System.
- Remove an instance of Certificate Management System from a system or host.
- Clone an instance of Certificate Management System.
- Set access permissions for Certificate Management System.
- Migrate configuration information from one version of Certificate Management System to another.
- Launch the Administration Server window (so that you can configure an Administration Server instance for administering Certificate Management System).
- Launch the CMS window.

Users and Groups Tab

The Users and Groups tab (shown in Figure 9-2) manages user accounts, group lists, and access control information for individual users and groups. All applications registered within the iPlanet Console framework share core user and group information in the user directory, which typically is a global directory for corporate-wide user data.

Figure 9-2 Users and Groups tab of iPlanet Console

From this tab, you can accomplish various user- and group-specific tasks, such as these:

- Add, modify, and delete user and group information in the user directory.
- Search for specific user and group entries in the user directory.

iPlanet Administration Server

iPlanet Administration Server is a web-based (HTTP) server that enables you to configure all your iPlanet servers, including Certificate Management System, through iPlanet Console. Administration Server (and the configuration directory) must be running before you can configure any of these servers. It is included with all iPlanet servers and is installed when you install your first server in a *server group*. A server group refers to servers that are installed in a server root directory and that are managed by a single instance of iPlanet Administration Server.

You access Administration Server by entering its URL in the iPlanet Console login screen. This URL is based on the computer host name and the port number you chose when you installed Certificate Management System. The format for the URL looks like this: `http://<machine_name>.<your_domain>.<domain>:<port>`

Whenever you try to gain access to Administration Server, you will be prompted to authenticate yourself to the configuration directory by entering your user ID and password. These are the *administrator* user name and password that you specified when you installed Certificate Management System (or the first server in the server group) and Administration Server on your computer. Once Administration Server is running, you can use iPlanet Console to administer all servers in that group, including Certificate Management System.

For complete details about iPlanet Administration Server, see *Managing Servers with iPlanet Console*. To locate an online version of this book, go to

`<server_root>/manual/index.html`.

Starting Administration Server

The CMS installation program automatically starts the instance of Administration Server that you identified during installation for monitoring Certificate Management System. If you stopped Administration Server after installation, you must start it before you can administer Certificate Management System from the CMS window.

You can start the server from iPlanet Console, the command line, or the Windows NT Services panel.

- To start Administration Server from iPlanet Console:
 - a. Log in to iPlanet Console (see “Logging In to iPlanet Console” on page 344).
 - b. In the Console tab, locate the Administration Server instance that you want to start, and double-click the corresponding entry.

The Administration Server window appears.

- To start Administration Server from the command line:

At the prompt, enter the following line:

```
<server_root>/admin-<instance_id>/start-admin
```

This command starts Administration Server at the port number you specified during installation. Once the server is running, you can use iPlanet Console to access Certificate Management System.

- Administration Server runs as a service in a Windows NT system. You can use the Windows NT Services panel to start the service directly.

Shutting Down Administration Server

It is good security practice to shut down Administration Server when you are not using it. This minimizes the chances of someone else changing your configuration. You can shut down the server from iPlanet Console, the command line, or the Windows NT Services panel.

- To shut down Administration Server from iPlanet Console:
 - a. Log in to iPlanet Console (see “Logging In to iPlanet Console” on page 344).
 - b. In the Console tab, locate the Administration Server instance that you want to shut down, and double-click the corresponding entry.

The Administration Server window appears.

- c. In the Tasks tab, click Stop the Server.
- To shut down Administration Server from the command line:

At the prompt, enter the following line:

```
<server_root>/admin-<instance_id>/stop-admin
```

- Administration Server runs as a service in a Windows NT system; you can use the Windows NT Services panel to stop the service directly.

Logging In to iPlanet Console

You can launch and use iPlanet Console only when the configuration directory and Administration Server are running. If the servers are not running, go to the command line and start them. For information on starting Administration Server from the command line, see “Starting Administration Server” on page 343. For information on starting the configuration directory, check the iPlanet Directory Server documentation.

When you launch iPlanet Console, it displays a login window. You are required to authenticate to the configuration directory by entering your administrator’s ID, your password, and the URL (including port number) of the Administration Server representing a server group to which you have access. You cannot use iPlanet Console without having login access to at least one server group on your network.

1. Open the iPlanet Console application by using the appropriate option:
 - For local access on a Unix machine, at the command-line prompt, enter the following line: `<server_root>/admin-<instance_id>/start-console`
 - Local access on a Windows NT machine, double-click the iPlanet Console icon on your desktop; this icon was created when you installed your first iPlanet server.



The iPlanet Console window appears.

2. Authenticate yourself to the configuration directory.

User ID. Type the *administrator ID* you specified when you installed Administration Server on your machine. You installed Administration Server either when you installed your first iPlanet server or as a part of CMS installation.

Password. Type the *administrator* password that you specified when you installed Administration Server on your computer during CMS installation.

Administration URL. This field should show the URL to Administration Server. If it doesn't or if it doesn't have the URL of Administration Server that you want, type the URL in this field. The URL is based on the computer host name and the Administration Server port number you chose when you installed Certificate Management System. Use this format:

```
http://<machine_name>.<your_domain>.<domain>:<port_number>
```

For example, if your domain name is `siroe` and you installed Administration Server on a host machine called `myHost` and specified port number `12345`, the URL would look like this: `http://myHost.siroe.com:12345`

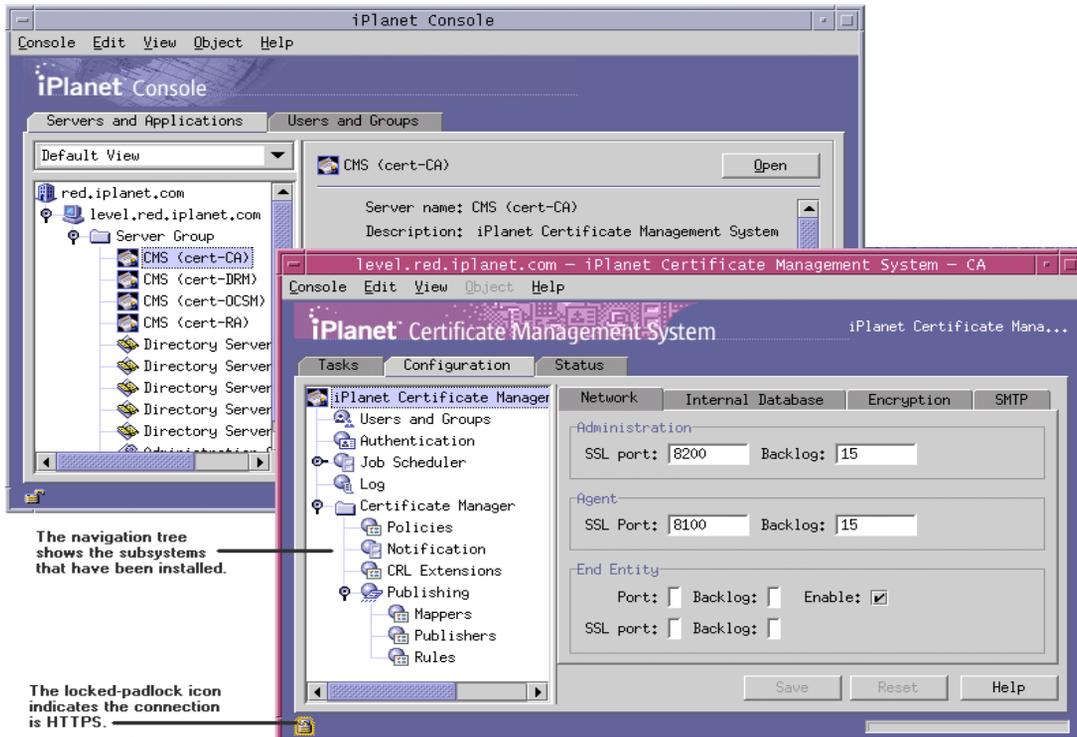
3. Click OK.

iPlanet Console appears with a list of all the servers and resources under your control (see Figure 9-1). To view general information about a specific server, click the corresponding entry. To access the administration interface for a specific server, double-click the corresponding entry.

The CMS Window

The CMS window is a GUI-based administration interface that allows you to perform day-to-day operational and managerial duties for Certificate Management System. You launch the CMS window from within iPlanet Console (Figure 9-3).

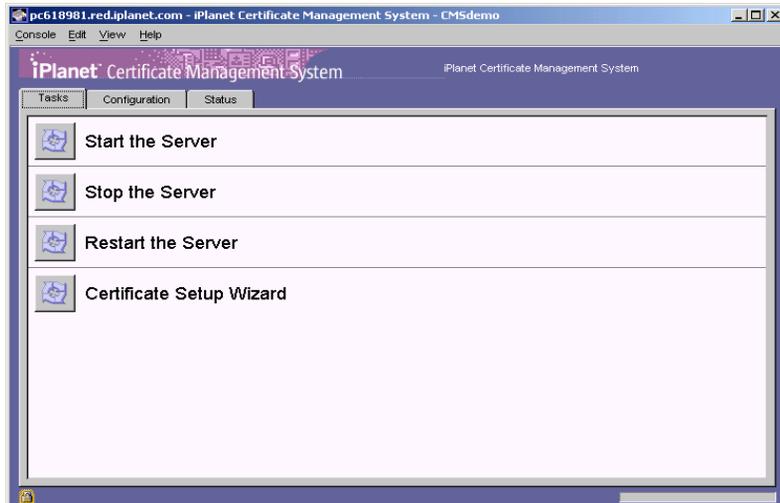
Figure 9-3 Certificate Management System window, launched from iPlanet Console



You can use the CMS window to access the server locally or remotely. The window has three separate tabs—Tasks, Configuration, and Status—each addressing specific administrative areas.

Tasks Tab

The Tasks tab enables you to perform tasks such as starting, stopping, and restarting the server, and running the Certificate Setup Wizard. For details see Chapter 8, “Starting and Stopping CMS Instances” and “Certificate Setup Wizard” on page 478.



Configuration Tab

The Configuration tab enables you to view and modify the configuration.

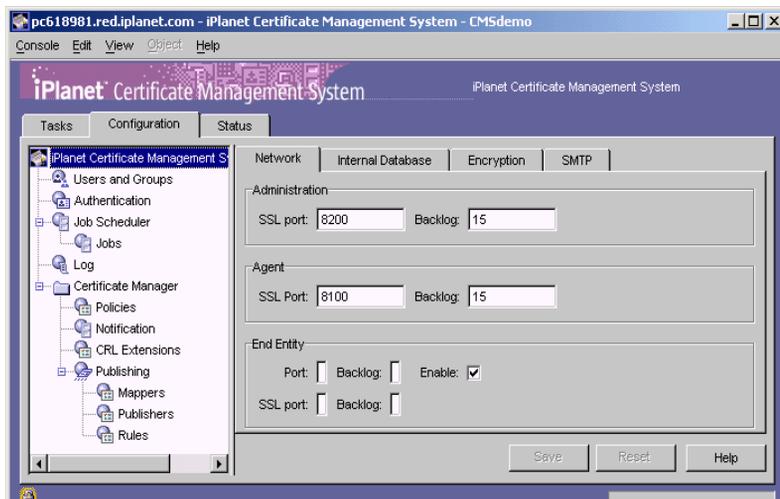


Table 9-1 provides details about the tasks you can accomplish from this tab. You access specific settings by selecting an entry in the navigation tree and working with the tabs that appear in the right pane.

Table 9-1 Tasks you can accomplish from the Configuration tab

Task	Description
Configuring network settings	This involves changing the administration, agent, and end-entity ports of Certificate Management System. For details, see Chapter 11, “Setting Up Ports.”
Configuring the internal database settings	This involves specifying the host name and port number of the Directory Server that Certificate Management System should use for storing data. For details, see Chapter 12, “Setting Up Internal Database.”
Setting up privileged users	<p>This involves operations such as the following:</p> <ul style="list-style-type: none"> • Entering information about privileged users (administrators, agents, and trusted managers) into the CMS internal database. • Modifying user information. • Deleting users from the database. <p>For details, see see Chapter 13, “Managing Privileged Users and Groups.”</p>
Managing CMS keys and certificates	<p>This involves operations such as the following:</p> <ul style="list-style-type: none"> • Managing the CMS certificate database. • Generating new and renewing existing certificates for the Certificate Manager, Registration Manager, Data Recovery Manager, and Online Certificate Status Manager. • Installing new hardware tokens. <p>For details, see Chapter 14, “Managing CMS Keys and Certificates.”</p>
Determining authentication for end users	<p>This involves operations such as the following:</p> <ul style="list-style-type: none"> • Viewing currently registered authentication plug-in modules. • Configuring Certificate Management System to use a specific authentication method to authenticate end users when they enroll for a certificate. • Registering custom authentication plug-in modules. <p>For details, see Chapter 15, “Setting Up End-User Authentication.”</p>

Table 9-1 Tasks you can accomplish from the Configuration tab (*Continued*)

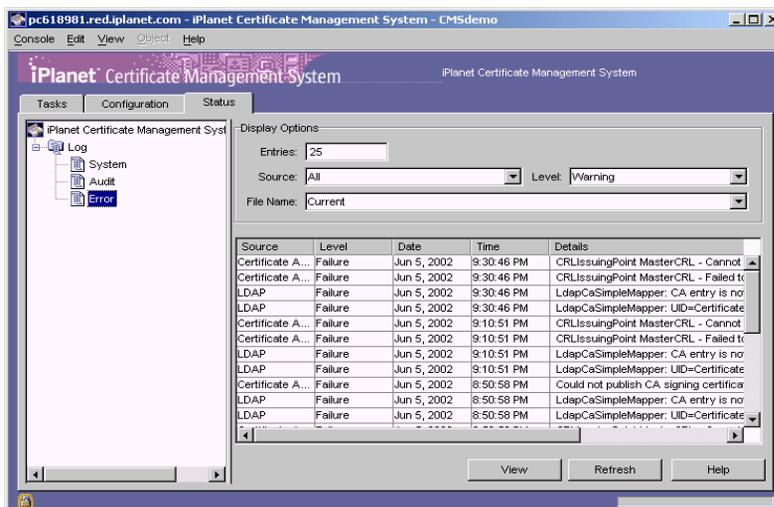
Task	Description
Enabling automated email notifications	<p>This involves operations such as the following:</p> <ul style="list-style-type: none"> • Entering the information required by the server to send automated notifications to one or more agents when a request enters the agent queue. • Entering the information required by the server to send automated certificate-issuance notifications to end entities when the server issues them certificates. • Specifying the host name and port number of the mail server that Certificate Management System should use for sending email notifications. • Customizing the notification message templates to suit your organization's requirements. <p>For details, see Chapter 16, "Setting Up Automated Notifications."</p>
Scheduling automated jobs	<p>This involves operations such as the following:</p> <ul style="list-style-type: none"> • Viewing currently registered plug-in modules for jobs. • Configuring Certificate Management System to execute specific jobs. <p>For details, see Chapter 17, "Scheduling Automated Jobs."</p>
Configuring certificate issuance and management policies	<p>This involves operations such as the following:</p> <ul style="list-style-type: none"> • Viewing currently registered policy plug-in modules for a Certificate Manager or Registration Manager. • Configuring the Certificate Manager or Registration Manager for certificate formulation, issuance, renewal, and revocation policies, and configuring the Data Recovery Manager for the archiving and recovery of end users' encryption private keys. <p>For details, see Chapter 18, "Setting Up Policies."</p>
Publishing certificates and CRLs	<p>This involves operations such as the following:</p> <ul style="list-style-type: none"> • Configuring the Certificate Manager to publish certificates and CRLs to an LDAP-compliant directory, such as iPlanet Directory Server. For details, see Chapter 19, "Setting Up LDAP Publishing." • Configuring the Certificate Manager to publish certificates and CRLs to a flat file for importing into other repositories. For details, see Chapter 20, "Publishing Certificates and CRLs to a File." • Configuring the Certificate Manager to publish CRLs to the CMS OCSP responder, Online Certificate Status Manager. For details, see Chapter 21, "Setting Up an OCSP Responder."

Table 9-1 Tasks you can accomplish from the Configuration tab (*Continued*)

Task	Description
Configuring the Data Recovery Manager	This involves configuring the Data Recovery Manager for archival and recovery of end users' encryption private keys. For details, see Chapter 22, "Setting Up Key Archival and Recovery."
Managing CMS logs	This involves configuring system, error, and audit logs maintained by Certificate Management System and using these logs to monitor the server's activities. For details, see Chapter 23, "Managing CMS Logs."
Backing up and restoring CMS data	This involves operations such as the following: <ul style="list-style-type: none"> Periodically backing up the CMS data. In the event of data loss, using the resulting archives to restore the data. For details, see Chapter 6, "Backing Up and Restoring Data" of <i>CMS Command-Line Tools Guide</i> .

Status Tab

The Status tab allows you to monitor the server by viewing the contents of various logs maintained by Certificate Management System.



You can monitor active as well as rotated System, Error, and Audit log files. For details, see "Monitoring CMS Logs" on page 803.

Logging In to the CMS Window

You access the CMS window from iPlanet Console. For details on iPlanet Console, see “iPlanet Console” on page 340.

The Console tab of iPlanet Console contains a list of network resources that are under your control. In this list you can identify CMS instances by their icons or by server identifiers you specified during installation (for example, you may have named a CMS instance ABC Corp CA).

NOTE Accessing the CMS window is a privileged operation that is restricted to CMS administrators. After you log in for the first time, create at least one user in each of the default groups; see “Groups and Their Privileges” on page 409.

To open the CMS window for a specific CMS instance:

1. Log in to iPlanet Console (see “Logging In to iPlanet Console” on page 344).
2. In the Console tab, select the Server Group that contains the CMS instance you want to use as your source.
3. In the navigation tree, locate the CMS instance you want to administer.
4. Select the instance and click Open or double-click the corresponding entry.

If the selected server is not running, you are asked to start the server first. In that case, start the server, and then repeat steps 2 through 4. For information on starting the server, see “Starting Certificate Management System” on page 322.

If the selected server is running, you are prompted to authenticate to Certificate Management System.



5. Enter the appropriate information:

User ID. If you are logging in for the first time, type the `Certificate Administrator ID`; you specified this user ID during installation (so that you could log in to the CMS window without having to create privileged-user entries). Otherwise, type your privileged-user ID (administrator ID).

Password. If you are logging in for the first time, type the `Certificate Administrator` password; you specified this password during installation (so that you could log in to the CMS window without having to create privileged-user entries). Otherwise, type your privileged-user (administrator) password; see “Administrators” on page 396.

Upon successful authentication, the CMS window appears (Figure 9-3 on page 346).

Configuration

Chapter 9, “Administration Tasks and Tools”

Chapter 10, “CMS Configuration”

Chapter 11, “Setting Up Ports”

Chapter 12, “Setting Up Internal Database”

Chapter 13, “Managing Privileged Users and Groups”

Chapter 14, “Managing CMS Keys and Certificates”

Chapter 15, “Setting Up End-User Authentication”

Chapter 16, “Setting Up Automated Notifications”

Chapter 17, “Scheduling Automated Jobs”

Chapter 18, “Setting Up Policies”

Chapter 19, “Setting Up LDAP Publishing”

Chapter 20, “Publishing Certificates and CRLs to a File”

Chapter 21, “Setting Up an OCSP Responder”

Chapter 22, “Setting Up Key Archival and Recovery”

Chapter 23, “Managing CMS Logs”

CMS Configuration

The runtime properties of iPlanet Certificate Management Server (CMS) are governed by a set of configuration parameters. These parameters are stored in a file that is read by the server during startup.

When you install Certificate Management System, the installer creates an ASCII file, named `CMS.cfg`, and populates it with the appropriate configuration parameters. You can control the way Certificate Management System functions by making the appropriate changes to the configuration information.

This chapter explains how the installation affects the number of configuration files created in your machine and their contents. It also explains ways in which you can modify the configuration and precautions you should take when doing so. The chapter ends with a road map to configuring individual subsystems.

The chapter has the following sections:

- Effects of Installation Type on Configuration (page 355)
- Locating the Configuration File (page 358)
- Modifying the Configuration (page 359)
- Road Map to Configuring Subsystems (page 376)

Effects of Installation Type on Configuration

For each instance of Certificate Management System there is a configuration file, named `CMS.cfg`. The configuration file controls the runtime properties of the corresponding CMS instance.

A CMS instance can include a single subsystem or two subsystems in one of the following combinations:

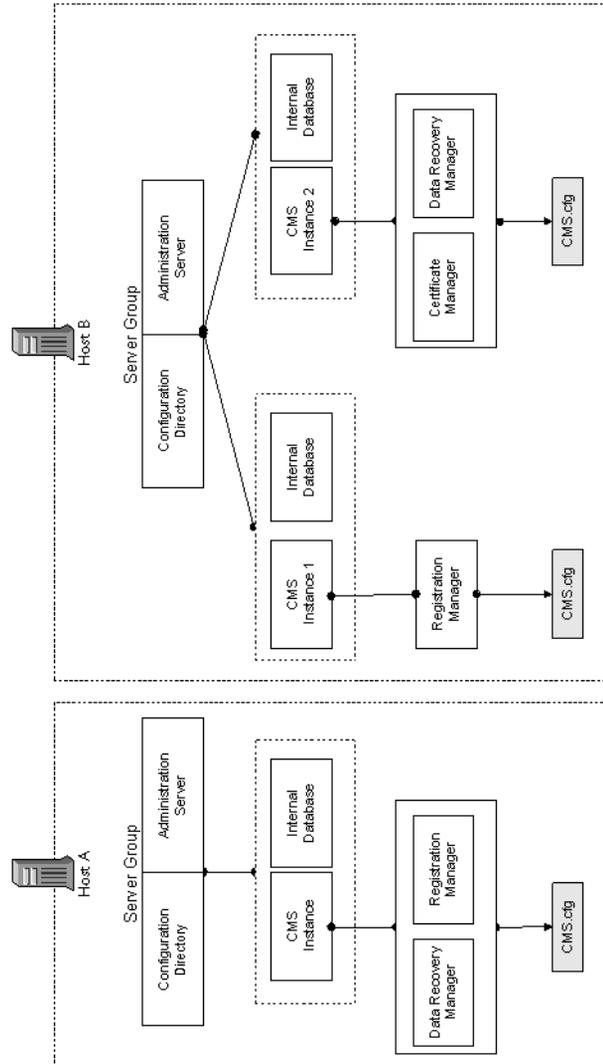
- A single Certificate Manager, Registration Manager, Data Recovery Manager, or Online Certificate Status Manager
- A Certificate Manager and Data Recovery Manager together
- A Registration Manager and Data Recovery Manager together

Figure 10-1 on page 357 illustrates a deployment scenario involving two instances of Certificate Management System running on the same host (Host A) and a single instance running on another host (Host B). Notice the two separate configuration files for the instances running on Host A, one for each CMS instance.

Although the names of both the configuration files are the same, the information included in the files differs according to the subsystems installed in each instance. For example, the configuration file for *CMS Instance 1* includes only those parameters that govern the Registration Manager, whereas the configuration file for *CMS Instance 2* includes parameters that control both the Certificate Manager and Data Recovery Manager.

It is also important to understand that subsystems installed in a CMS instance share certain parts of the configuration. They use the same

- Administration, agent, and end-entity ports for interaction
- Internal token and trust database
- SSL ciphers during SSL negotiation
- Privileged users (administrators and agents)
- Log files to log messages
- Internal database for data storage

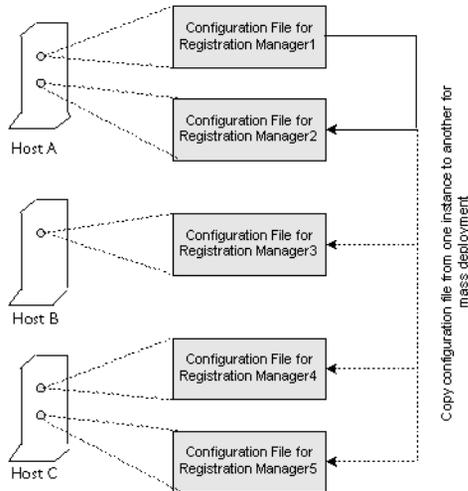
Figure 10-1 How installation affects configuration

Duplicating Configuration From One Instance to Another

If you have deployed a large number of CMS instances that are identical—for example, multiple Registration Managers—and you want all these instances to

have the same configuration, you can accomplish this by configuring one of the instances and then replacing the configuration files of the other instances with the one that contains the required configuration. Figure 10-2 illustrates this quick way of deploying multiple Registration Managers with the same configuration.

Figure 10-2 Duplicating a configuration



NOTE Be careful when replacing configuration of one instance with another. The configuration file for an instance contains instance-specific parameters. If you replace these parameters, the instance will fail to start or function properly.

Locating the Configuration File

Each instance of Certificate Management System has its own configuration file, `CMS.cfg`. The default location for this file is as follows:

```
<server_root>/cert-<instance_id>/config
```

Modifying the Configuration

You can modify the CMS configuration in two ways:

- By changing the configuration parameter values from the CMS window. This is the recommended method for changing configuration. See “Changing the Configuration From the CMS Window” on page 359.
- By manually changing the configuration parameter values in the configuration file, `CMS.cfg`. See “Changing the Configuration by Editing the Configuration File” on page 359.

Changing the Configuration From the CMS Window

The CMS window allows you to view the current configuration of a CMS instance and make the required changes. Because this is the recommended method for changing configuration, the chapters that follow focus on explaining how to change the various configuration parameter values from the CMS window.

NOTE You may find the road map provided in “Road Map to Configuring Subsystems” on page 376 useful in setting up your CMS instances.

Changing the Configuration by Editing the Configuration File

This section explains how to change the CMS configuration by editing the configuration parameter values in the file `CMS.cfg`. This ASCII file is read by Certificate Management System when it is started.

CAUTION Do not edit the configuration file directly if you are not familiar with the configuration parameters or if you are not sure that the changes you intend to make are acceptable by the server. Certificate Management System will fail to start up if you make incorrect modifications to the configuration file. Incorrect configuration can also result in data loss.

Also, before you start editing the configuration file, be sure to read “Guidelines for Editing the Configuration File” on page 360.

To modify the configuration file directly:

1. Stop the CMS instance whose configuration file you want to edit (see “Stopping Certificate Management System” on page 330).
2. Open a terminal window.
3. Go to this directory: `<server_root>/cert-<instance_id>/config`
4. Open the configuration file, `CMS.cfg`, in a text editor.
5. Edit information in the file and save your changes.
6. Restart Certificate Management System (see “Restarting Certificate Management System” on page 332).

Guidelines for Editing the Configuration File

The file-based, configuration-store implementation for Certificate Management System is based on `java.util.Properties`. The following guidelines may help you interpret the information in the configuration file.

- The format of the configuration file is as follows:

```
#comment
[parameter]=value
value

[parameter]
multi
line
value (e.g. base-64 encoded certificate)
```

- Comment lines, blank lines, unknown parameters, or misspelled parameters are ignored by Certificate Management System. Comment lines begin with a number sign (#). A line beginning with white space is considered a continuation of the previous line.
- The configuration file has many sections. Some sections contain parameters specific to the subsystems that have been installed; the other sections contain parameters that are shared by the subsystems. Subsystem-specific parameters are distinguished by a prefix identifying the subsystem:
 - `ca` for the Certificate Manager
 - `ra` for the Registration Manager
 - `kra` for the Data Recovery Manager
 - `ocsp` for the Online Certificate Status Manager

- The parameter names and their values are strings. The parameter names can be hierarchically structured with `' . '` notation with multiple levels—for example, `ca.Policy.rule.RSAKeyRule.maxSize`. The entries corresponding to a lower level (such as `Policy` in the example) can be requested from the configuration corresponding to its higher level (`ca` in the example).
- The values that need to be localized (such as distinguished names in multibyte format) should be entered in `utf8` format. For more information on this format, see the document *UTF-8, a transformation format of Unicode and ISO 10646*, available at this URL:
<http://info.internet.isi.edu:80/in-notes/rfc/files/rfc2044.txt>
- Certificate Management System writes out the configuration in a sorted order.
- The values of some parameters are referenced to other parts of the configuration file. For example, assume that a parameter is defined as `subsystem.id=ca`; when this parameter is processed by the server, all the parameters beginning with `ca` will be used.
- The configuration file supports Unix-style file separator, the forward slash (`/`). If the backward slash (`\`) file separator is required, use two backward slashes (`\\`) instead of one.
- The sample shown on page 363 illustrates how authentication-specific information appears in the configuration file. Keep the following points in mind:
 - All authentication-specific information, such as names of registered authentication plug-in modules and any configured instances, appears in the Authentication section of the configuration file.
 - Each registered authentication plug-in module is identified by its implementation name and the corresponding Java class.
 - Each configured instance of an authentication module is identified by the name or ID you specified when creating it.
 - You can create multiple instances out of an implementation; each instance must have a unique name.
 - The name of an authentication instance must be used in the corresponding enrollment form so that the server is able to determine the authentication method during end-user enrollment. For details, see “Step 5. Set Up the Enrollment Interface” on page 559.

- The sample shown on page 372 shows how Job Scheduler-specific information appears in the configuration file. Note the following:
 - All job-specific information, such as registered job modules and configured instances, appears in the Job Scheduler section of the configuration file.
 - Each registered job module is identified by its implementation name and the corresponding Java class.
 - Each job (or configured instance of a job module) is identified by the name specified when the job was created.
 - You can create as many instances of an implementation as you like; each instance must have a unique name.
- The sample shown on page 364 illustrates how policy-specific information appears in the configuration file. Note the following:
 - All policy-specific information, such as registered policy plug-in implementations, configured rules, and ordering, appear in the Policy section of the configuration file. If you have installed more than one subsystem in a CMS instance, for example Certificate Manager and Data Recovery Manager together, the configuration file will include policy sections that are specific to each of the subsystems that share the configuration.

You can identify policy pertaining to a subsystem by these prefixes: Certificate Manager by `ca`, Registration Manager by `ra`, and Data Recovery Manager by `kra`.
 - Each registered policy plug-in module is identified by its implementation name and the corresponding Java class.
 - Each configured rule of a policy module is identified by the name specified when the rule was created.
 - You can create multiple rules out of an implementation; each rule must have a unique name.
- The sample on page 373 illustrates how information specific to logs appears in the configuration file.

Sample Configuration File

The following sample configuration is of a Certificate Manager.

NOTE This sample file includes some of the parameters used by Certificate Management System. However, there is no guarantee that an arbitrary set of options you create will work.

```

_000=##
_001=## File Created On      : Sun Jan 02 23:02:35 PST 2000
_002=##

instanceRoot=/usr/netscape/cert-testCA
machineName=testCA.siroe.com

agentGateway._000=##
agentGateway._001=## Agent Gateway
agentGateway._002=##

    agentGateway.docRoot=/usr/netscape/cert-testCA/web/agent
    agentGateway.dynamicVariables=serverdate=serverdate()
    agentGateway.enableAdminEnroll=true
    agentGateway.enableBulkInterface=true
    agentGateway.keepAliveOn=true
    agentGateway.mimeTypeConf=/usr/netscape/cert-testCA/config/mime.types
    agentGateway.numServices=1
    agentGateway.service0=https
    agentGateway.CAGetBySerial.successTemplate=/ca/ImportCert.template
    agentGateway.adminEnroll.successTemplate=/ca/EnrollSuccess.template
    agentGateway.bulkissuance.errorTemplate=/ca/bulkissuance.template
    agentGateway.bulkissuance.pendingTemplate=/ca/bulkissuance.template
    agentGateway.bulkissuance.rejectedTemplate=/ca/bulkissuance.template
    agentGateway.bulkissuance.successTemplate=/ca/bulkissuance.template
    agentGateway.bulkissuance.svcpendingTemplate=/ca/bulkissuance.template
    agentGateway.bulkissuance.unauthorizedTemplate=/ca/bulkissuance.template
    agentGateway.bulkissuance.unexpectedErrorTemplate=/ca/bulkissuance.template
    agentGateway.https.backlog=15
    agentGateway.https.nickName=Server-Cert cert-testCA
    agentGateway.https.port=4605
    agentGateway.https.type=https

auths._000=##
auths._001=## Authentication
auths._002=##
auths.impl._000=##
auths.impl._001=## authentication manager implementations
auths.impl._002=##

```

Modifying the Configuration

```
auths.impl.NISAuth.class=com.netscape.certsrv.authentication.NISAuth
auths.impl.PortalEnroll.class=com.netscape.certsrv.authentication.PortalEnroll
auths.impl.UidPwdDirAuth.class=com.netscape.certsrv.authentication.
    UidPwdDirAuthentication
auths.impl.UidPwdPinDirAuth.class=com.netscape.certsrv.authentication.
    UidPwdPinDirAuthentication
auths.revocationChecking.bufferSize=5
auths.revocationChecking.ca=ca
auths.revocationChecking.enabled=true
auths.revocationChecking.unknownStateInterval=0
auths.revocationChecking.validityInterval=120

ca.id=ca
ca.local=true

ca.Policy.order=KeyAlgRule, RSAKeyRule, DefaultValidityRule, RenewalConstraintsRule,
DefaultRenewalValidityRule, RevocationConstraintsRule, DefaultRevocationRule,
NSCertTypeExt, CMCertKeyUsageExt, RMCertKeyUsageExt, ClientCertKeyUsageExt,
ServerCertKeyUsageExt, ObjSignCertKeyUsageExt, SubjectKeyIdentifierExt,
CertificatePoliciesExt, NSCComment, OCSPNoCheckExt, OCSPSigningExt, CODESigningExt,
GenericASN1Ext, CRLDistributionPointsExt, SubjectAltNameExt, SigningAlgRule,
AuthorityKeyIdentifierExt, BasicConstraintsExt, UniqueSubjectName, NameConstraintsExt,
PolicyConstraintsExt, SubCANameCheck, PolicyMappingsExt, IssuerRule

ca.Policy.processor=classic

ca.Policy.impl._000=##
ca.Policy.impl._001=## Policy Implementations
ca.Policy.impl._002=##

ca.Policy.impl.AuthInfoAccessExt.class=com.netscape.certsrv.policy.AuthInfoAccessExt
ca.Policy.impl.AuthorityKeyIdentifierExt.class=com.netscape.certsrv.policy.
    AuthorityKeyIdentifierExt
ca.Policy.impl.BasicConstraintsExt.class=com.netscape.certsrv.policy.
    BasicConstraintsExt
ca.Policy.impl.CRLDistributionPointsExt.class=com.netscape.certsrv.policy.
    CRLDistributionPointsExt
ca.Policy.impl.CertificatePoliciesExt.class=com.netscape.certsrv.policy.
    CertificatePoliciesExt
ca.Policy.impl.DSAKeyConstraints.class=com.netscape.certsrv.policy.DSAKeyConstraints
ca.Policy.impl.DefaultRevocation.class=com.netscape.certsrv.policy.DefaultRevocation
ca.Policy.impl.ExtendedKeyUsageExt.class=com.netscape.certsrv.policy.
    ExtendedKeyUsageExt
ca.Policy.impl.GenericASN1Ext.class=com.netscape.certsrv.policy.GenericASN1Ext
ca.Policy.impl.IssuerAltNameExt.class=com.netscape.certsrv.policy.IssuerAltNameExt
ca.Policy.impl.IssuerConstraints.class=com.netscape.certsrv.policy.IssuerConstraints
ca.Policy.impl.KeyAlgorithmConstraints.class=com.netscape.certsrv.policy.
    KeyAlgorithmConstraints
ca.Policy.impl.KeyUsageExt.class=com.netscape.certsrv.policy.KeyUsageExt
ca.Policy.impl.NSCComment.class=com.netscape.certsrv.policy.NSCComment
ca.Policy.impl.NSCertTypeExt.class=com.netscape.certsrv.policy.NSCertTypeExt
ca.Policy.impl.NameConstraintsExt.class=com.netscape.certsrv.policy.NameConstraintsExt
ca.Policy.impl.OCSPNoCheckExt.class=com.netscape.certsrv.policy.OCSPNoCheckExt
ca.Policy.impl.AttributePresent.class=com.netscape.certsrv.policy.AttributePresent
```

```

ca.Policy.impl.PolicyConstraintsExt.class=com.netscape.certsrv.policy.
  PolicyConstraintsExt
ca.Policy.impl.PolicyMappingsExt.class=com.netscape.certsrv.policy.PolicyMappingsExt
ca.Policy.impl.PrivateKeyUsagePeriodExt.class=com.netscape.certsrv.policy.
  PrivateKeyUsagePeriodExt
ca.Policy.impl.RSAKeyConstraints.class=com.netscape.certsrv.policy.RSAKeyConstraints
ca.Policy.impl.RenewalConstraints.class=com.netscape.certsrv.policy.RenewalConstraints
ca.Policy.impl.RenewalValidityConstraints.class=com.netscape.certsrv.policy.
  RenewalValidityConstraints
ca.Policy.impl.RevocationConstraints.class=com.netscape.certsrv.policy.
  RevocationConstraints
ca.Policy.impl.SigningAlgorithmConstraints.class=com.netscape.certsrv.policy.
  SigningAlgorithmConstraints
ca.Policy.impl.SubCANameCheck.class=com.netscape.certsrv.policy.SubCANameCheck
ca.Policy.impl.SubjectAltNameExt.class=com.netscape.certsrv.policy.SubjAltNameExt
ca.Policy.impl.SubjectDirectoryAttributesExt.class=com.netscape.certsrv.policy.
  SubjectDirectoryAttributesExt
ca.Policy.impl.SubjectKeyIdentifierExt.class=com.netscape.certsrv.policy.
  SubjectKeyIdentifierExt
ca.Policy.impl.UniqueSubjectName.class=com.netscape.certsrv.policy.UniqueSubjectName
ca.Policy.impl.ValidityConstraints.class=com.netscape.certsrv.policy.
  ValidityConstraints

ca.Policy.rule.AuthorityKeyIdentifierExt.enable=true
ca.Policy.rule.AuthorityKeyIdentifierExt.implName=AuthorityKeyIdentifierExt
ca.Policy.rule.AuthorityKeyIdentifierExt.predicate=

ca.Policy.rule.BasicConstraintsExt.enable=true
ca.Policy.rule.BasicConstraintsExt.implName=BasicConstraintsExt
ca.Policy.rule.BasicConstraintsExt.predicate=HTTP_PARAMS.certType == ca
ca.Policy.rule.BasicConstraintsExt.removeBasicExt=true

ca.Policy.rule.CMCertKeyUsageExt.crlSign=true
ca.Policy.rule.CMCertKeyUsageExt.digitalSignature=true
ca.Policy.rule.CMCertKeyUsageExt.enable=true
ca.Policy.rule.CMCertKeyUsageExt.implName=KeyUsageExt
ca.Policy.rule.CMCertKeyUsageExt.keyCertsign=true
ca.Policy.rule.CMCertKeyUsageExt.nonRepudiation=true
ca.Policy.rule.CMCertKeyUsageExt.predicate=certType==ca

ca.Policy.rule.CODESigningExt.critical=false
ca.Policy.rule.CODESigningExt.enable=true
ca.Policy.rule.CODESigningExt.id0=1.3.6.1.5.5.7.3.3
ca.Policy.rule.CODESigningExt.implName=ExtendedKeyUsageExt
ca.Policy.rule.CODESigningExt.predicate=certType==codeSignClient

ca.Policy.rule.CRLDistributionPointsExt.enable=false
ca.Policy.rule.CRLDistributionPointsExt.implName=CRLDistributionPointsExt
ca.Policy.rule.CRLDistributionPointsExt.issuerName0=
ca.Policy.rule.CRLDistributionPointsExt.issuerName1=
ca.Policy.rule.CRLDistributionPointsExt.issuerName2=
ca.Policy.rule.CRLDistributionPointsExt.issuerType0=

```

```

ca.Policy.rule.CRLDistributionPointsExt.issuerType1=
ca.Policy.rule.CRLDistributionPointsExt.issuerType2=
ca.Policy.rule.CRLDistributionPointsExt.numPoints=0
ca.Policy.rule.CRLDistributionPointsExt.pointName0=
ca.Policy.rule.CRLDistributionPointsExt.pointName1=
ca.Policy.rule.CRLDistributionPointsExt.pointName2=
ca.Policy.rule.CRLDistributionPointsExt.pointType0=
ca.Policy.rule.CRLDistributionPointsExt.pointType1=
ca.Policy.rule.CRLDistributionPointsExt.pointType2=
ca.Policy.rule.CRLDistributionPointsExt.predicate=
ca.Policy.rule.CRLDistributionPointsExt.reasons0=
ca.Policy.rule.CRLDistributionPointsExt.reasons1=
ca.Policy.rule.CRLDistributionPointsExt.reasons2=

ca.Policy.rule.CertificatePoliciesExt.enable=false
ca.Policy.rule.CertificatePoliciesExt.implName=CertificatePoliciesExt
ca.Policy.rule.CertificatePoliciesExt.policyId=
ca.Policy.rule.CertificatePoliciesExt.predicate=

ca.Policy.rule.ClientCertKeyUsageExt.digitalSignature=true
ca.Policy.rule.ClientCertKeyUsageExt.enable=true
ca.Policy.rule.ClientCertKeyUsageExt.implName=KeyUsageExt
ca.Policy.rule.ClientCertKeyUsageExt.keyEncipherment=true
ca.Policy.rule.ClientCertKeyUsageExt.nonRepudiation=true
ca.Policy.rule.ClientCertKeyUsageExt.predicate=certType==client

ca.Policy.rule.DSAKeyRule.enable=true
ca.Policy.rule.DSAKeyRule.implName=DSAKeyConstraints
ca.Policy.rule.DSAKeyRule.maxSize=2048
ca.Policy.rule.DSAKeyRule.minSize=512
ca.Policy.rule.DSAKeyRule.predicate=

ca.Policy.rule.DefaultRenewalValidityRule.enable=true
ca.Policy.rule.DefaultRenewalValidityRule.implName=RenewalValidityConstraints
ca.Policy.rule.DefaultRenewalValidityRule.maxValidity=365
ca.Policy.rule.DefaultRenewalValidityRule.minValidity=30
ca.Policy.rule.DefaultRenewalValidityRule.predicate=
ca.Policy.rule.DefaultRenewalValidityRule.renewalInterval=15

ca.Policy.rule.DefaultRevocationRule.enable=true
ca.Policy.rule.DefaultRevocationRule.implName=DefaultRevocation
ca.Policy.rule.DefaultRevocationRule.predicate=

ca.Policy.rule.DefaultValidityRule.enable=true
ca.Policy.rule.DefaultValidityRule.implName=ValidityConstraints
ca.Policy.rule.DefaultValidityRule.maxValidity=365
ca.Policy.rule.DefaultValidityRule.minValidity=30
ca.Policy.rule.DefaultValidityRule.predicate=

ca.Policy.rule.GenericASN1Ext.critical=false
ca.Policy.rule.GenericASN1Ext.enable=false
ca.Policy.rule.GenericASN1Ext.implName=GenericASN1Ext

```

```

ca.Policy.rule.GenericASN1Ext.name=
ca.Policy.rule.GenericASN1Ext.oid=
ca.Policy.rule.GenericASN1Ext.pattern=
ca.Policy.rule.GenericASN1Ext.predicate=
ca.Policy.rule.GenericASN1Ext.attribute.0.source=
ca.Policy.rule.GenericASN1Ext.attribute.0.type=
ca.Policy.rule.GenericASN1Ext.attribute.0.value=
ca.Policy.rule.GenericASN1Ext.attribute.1.source=
ca.Policy.rule.GenericASN1Ext.attribute.1.type=
ca.Policy.rule.GenericASN1Ext.attribute.1.value=
ca.Policy.rule.GenericASN1Ext.attribute.2.source=
ca.Policy.rule.GenericASN1Ext.attribute.2.type=
ca.Policy.rule.GenericASN1Ext.attribute.2.value=
ca.Policy.rule.GenericASN1Ext.attribute.3.source=
ca.Policy.rule.GenericASN1Ext.attribute.3.type=
ca.Policy.rule.GenericASN1Ext.attribute.3.value=
ca.Policy.rule.GenericASN1Ext.attribute.4.source=
ca.Policy.rule.GenericASN1Ext.attribute.4.type=
ca.Policy.rule.GenericASN1Ext.attribute.4.value=
ca.Policy.rule.GenericASN1Ext.attribute.5.source=
ca.Policy.rule.GenericASN1Ext.attribute.5.type=
ca.Policy.rule.GenericASN1Ext.attribute.5.value=
ca.Policy.rule.GenericASN1Ext.attribute.6.source=
ca.Policy.rule.GenericASN1Ext.attribute.6.type=
ca.Policy.rule.GenericASN1Ext.attribute.6.value=
ca.Policy.rule.GenericASN1Ext.attribute.7.source=
ca.Policy.rule.GenericASN1Ext.attribute.7.type=
ca.Policy.rule.GenericASN1Ext.attribute.7.value=
ca.Policy.rule.GenericASN1Ext.attribute.8.source=
ca.Policy.rule.GenericASN1Ext.attribute.8.type=
ca.Policy.rule.GenericASN1Ext.attribute.8.value=
ca.Policy.rule.GenericASN1Ext.attribute.9.source=
ca.Policy.rule.GenericASN1Ext.attribute.9.type=
ca.Policy.rule.GenericASN1Ext.attribute.9.value=

ca.Policy.rule.IssuerRule.enable=false
ca.Policy.rule.IssuerRule.implName=IssuerConstraints
ca.Policy.rule.IssuerRule.issuerDN=
ca.Policy.rule.IssuerRule.predicate=certType==client AND certauthEnroll==on

ca.Policy.rule.KeyAlgRule.algorithms=RSA
ca.Policy.rule.KeyAlgRule.enable=true
ca.Policy.rule.KeyAlgRule.implName=KeyAlgorithmConstraints
ca.Policy.rule.KeyAlgRule.predicate=

ca.Policy.rule.NSCComment.enable=false
ca.Policy.rule.NSCComment.implName=NSCComment
ca.Policy.rule.NSCComment.policyId=
ca.Policy.rule.NSCComment.predicate=

ca.Policy.rule.NSCertTypeExt.enable=true
ca.Policy.rule.NSCertTypeExt.implName=NSCertTypeExt

```

```

ca.Policy.rule.NSCertTypeExt.predicate=certType!=CEP-Request

ca.Policy.rule.NameConstraintsExt.critical=true
ca.Policy.rule.NameConstraintsExt.enable=false
ca.Policy.rule.NameConstraintsExt.implName=NameConstraintsExt
ca.Policy.rule.NameConstraintsExt.numExcludedSubtrees=3
ca.Policy.rule.NameConstraintsExt.numPermittedSubtrees=3
ca.Policy.rule.NameConstraintsExt.predicate=certType == ca
ca.Policy.rule.NameConstraintsExt.excludedSubtrees0.base=
ca.Policy.rule.NameConstraintsExt.excludedSubtrees0.max=-1
ca.Policy.rule.NameConstraintsExt.excludedSubtrees0.min=0
ca.Policy.rule.NameConstraintsExt.excludedSubtrees0.valueType=
ca.Policy.rule.NameConstraintsExt.excludedSubtrees1.base=
ca.Policy.rule.NameConstraintsExt.excludedSubtrees1.max=-1
ca.Policy.rule.NameConstraintsExt.excludedSubtrees1.min=0
ca.Policy.rule.NameConstraintsExt.excludedSubtrees1.valueType=
ca.Policy.rule.NameConstraintsExt.excludedSubtrees2.base=
ca.Policy.rule.NameConstraintsExt.excludedSubtrees2.max=-1
ca.Policy.rule.NameConstraintsExt.excludedSubtrees2.min=0
ca.Policy.rule.NameConstraintsExt.excludedSubtrees2.valueType=
ca.Policy.rule.NameConstraintsExt.permittedSubtrees0.base=
ca.Policy.rule.NameConstraintsExt.permittedSubtrees0.max=-1
ca.Policy.rule.NameConstraintsExt.permittedSubtrees0.min=0
ca.Policy.rule.NameConstraintsExt.permittedSubtrees0.valueType=
ca.Policy.rule.NameConstraintsExt.permittedSubtrees1.base=
ca.Policy.rule.NameConstraintsExt.permittedSubtrees1.max=-1
ca.Policy.rule.NameConstraintsExt.permittedSubtrees1.min=0
ca.Policy.rule.NameConstraintsExt.permittedSubtrees1.valueType=
ca.Policy.rule.NameConstraintsExt.permittedSubtrees2.base=
ca.Policy.rule.NameConstraintsExt.permittedSubtrees2.max=-1
ca.Policy.rule.NameConstraintsExt.permittedSubtrees2.min=0
ca.Policy.rule.NameConstraintsExt.permittedSubtrees2.valueType=

ca.Policy.rule.OCSFNoCheckExt.critical=false
ca.Policy.rule.OCSFNoCheckExt.enable=true
ca.Policy.rule.OCSFNoCheckExt.implName=OCSFNoCheckExt
ca.Policy.rule.OCSFNoCheckExt.predicate=certType==ocspResponder

ca.Policy.rule.OCSFSigningExt.critical=false
ca.Policy.rule.OCSFSigningExt.enable=true
ca.Policy.rule.OCSFSigningExt.id0=1.3.6.1.5.5.7.3.9
ca.Policy.rule.OCSFSigningExt.implName=ExtendedKeyUsageExt
ca.Policy.rule.OCSFSigningExt.predicate=certType==ocspResponder

ca.Policy.rule.ObjSignCertKeyUsageExt.digitalSignature=true
ca.Policy.rule.ObjSignCertKeyUsageExt.enable=true
ca.Policy.rule.ObjSignCertKeyUsageExt.implName=KeyUsageExt
ca.Policy.rule.ObjSignCertKeyUsageExt.keyCertsSign=true
ca.Policy.rule.ObjSignCertKeyUsageExt.predicate=certType==objSignClient

ca.Policy.rule.PolicyConstraintsExt.critical=false

```

```

ca.Policy.rule.PolicyConstraintsExt.enable=false
ca.Policy.rule.PolicyConstraintsExt.implName=PolicyConstraintsExt
ca.Policy.rule.PolicyConstraintsExt.inhibitPolicyMapping=0
ca.Policy.rule.PolicyConstraintsExt.predicate=certType==ca
ca.Policy.rule.PolicyConstraintsExt.reqExplicitPolicy=0

ca.Policy.rule.PolicyMappingsExt.critical=false
ca.Policy.rule.PolicyMappingsExt.enable=false
ca.Policy.rule.PolicyMappingsExt.implName=PolicyMappingsExt
ca.Policy.rule.PolicyMappingsExt.numPolicyMappings=1
ca.Policy.rule.PolicyMappingsExt.predicate=certType==ca
ca.Policy.rule.PolicyMappingsExt.policyMap0.issuerDomainPolicy=
ca.Policy.rule.PolicyMappingsExt.policyMap0.subjectDomainPolicy=

ca.Policy.rule.RMCertKeyUsageExt.digitalSignature=true
ca.Policy.rule.RMCertKeyUsageExt.enable=true
ca.Policy.rule.RMCertKeyUsageExt.implName=KeyUsageExt
ca.Policy.rule.RMCertKeyUsageExt.nonRepudiation=true
ca.Policy.rule.RMCertKeyUsageExt.predicate=certType==ra

ca.Policy.rule.RSAKeyRule.enable=false
ca.Policy.rule.RSAKeyRule.exponents=3,7,17,65537
ca.Policy.rule.RSAKeyRule.implName=RSAKeyConstraints
ca.Policy.rule.RSAKeyRule.maxSize=2048
ca.Policy.rule.RSAKeyRule.minSize=512
ca.Policy.rule.RSAKeyRule.predicate=

ca.Policy.rule.RenewalConstraintsRule.enable=true
ca.Policy.rule.RenewalConstraintsRule.implName=RenewalConstraints
ca.Policy.rule.RenewalConstraintsRule.predicate=

ca.Policy.rule.RevocationConstraintsRule.enable=true
ca.Policy.rule.RevocationConstraintsRule.implName=RevocationConstraints
ca.Policy.rule.RevocationConstraintsRule.predicate=

ca.Policy.rule.ServerCertKeyUsageExt.dataEncipherment=true
ca.Policy.rule.ServerCertKeyUsageExt.digitalSignature=true
ca.Policy.rule.ServerCertKeyUsageExt.enable=true
ca.Policy.rule.ServerCertKeyUsageExt.implName=KeyUsageExt
ca.Policy.rule.ServerCertKeyUsageExt.keyEncipherment=true
ca.Policy.rule.ServerCertKeyUsageExt.nonRepudiation=true
ca.Policy.rule.ServerCertKeyUsageExt.predicate=certType==server

ca.Policy.rule.SigningAlgRule.algorithms=MD5withRSA,MD2withRSA,SHA1withRSA,SHA1withDSA
ca.Policy.rule.SigningAlgRule.enable=true
ca.Policy.rule.SigningAlgRule.implName=SigningAlgorithmConstraints
ca.Policy.rule.SigningAlgRule.predicate=

ca.Policy.rule.SubCANameCheck.enable=true
ca.Policy.rule.SubCANameCheck.implName=SubCANameCheck
ca.Policy.rule.SubCANameCheck.predicate=
ca.Policy.rule.SubjectAltNameExt.enable=true

```

Modifying the Configuration

```
ca.Policy.rule.SubjectAltNameExt.enableManualValues=false
ca.Policy.rule.SubjectAltNameExt.implName=SubjectAltNameExt

ca.Policy.rule.SubjectKeyIdentifierExt.enable=true
ca.Policy.rule.SubjectKeyIdentifierExt.implName=SubjectKeyIdentifierExt
ca.Policy.rule.SubjectKeyIdentifierExt.predicate=certType==ca

ca.Policy.rule.UniqueSubjectName.enable=false
ca.Policy.rule.UniqueSubjectName.implName=UniqueSubjectName
ca.Policy.rule.UniqueSubjectName.predicate=

ca.crl._000=##
ca.crl._001=## CA CRL
ca.crl._002=##

ca.crl.MasterCRL.allowExtensions=false
ca.crl.MasterCRL.autoUpdateInterval=20
ca.crl.MasterCRL.class=com.netscape.certsrv.ca.CRLIssuingPoint
ca.crl.MasterCRL.description=CA's complete Certificate Revocation List

ca.notification.certIssued.emailSubject=Your Certificate Request
ca.notification.certIssued.emailTemplate=/usr/netscape/cert-testCA/emails/
certIssued_CA.html
ca.notification.certIssued.enabled=false
ca.notification.certIssued.senderEmail=

ca.notification.requestInQ.emailSubject=Certificate Request in Queue
ca.notification.requestInQ.emailTemplate=/usr/netscape/cert-testCA/emails/
reqInQueue.html
ca.notification.requestInQ.enabled=false
ca.notification.requestInQ.recipientEmail=
ca.notification.requestInQ.senderEmail=

ca.publish.mapper.impl.LdapDNCompsMap.class=com.netscape.certsrv.ldap.LdapCertCompsMap
ca.publish.mapper.impl.LdapDNExactMap.class=com.netscape.certsrv.ldap.LdapCertExactMap
ca.publish.mapper.impl.LdapSimpleMap.class=com.netscape.certsrv.ldap.LdapSimpleMap
ca.publish.mapper.impl.LdapSubjAttrMap.class=com.netscape.certsrv.ldap.LdapCertSubjMap
ca.publish.mapper.instance.LdapCaCertMap.dnPattern=UID=$cert.cn,OU=people,O=$cert.o
ca.publish.mapper.instance.LdapCaCertMap.pluginName=LdapSimpleMap
ca.publish.mapper.instance.LdapCrlMap.dnPattern=UID=$cert.cn,OU=people,O=$cert.o
ca.publish.mapper.instance.LdapCrlMap.pluginName=LdapSimpleMap
ca.publish.mapper.instance.LdapUserCertMap.dnPattern=UID=$cert.UID,OU=people,O=$cert.o
ca.publish.mapper.instance.LdapUserCertMap.pluginName=LdapSimpleMap
ca.publish.publisher.impl.FileBasedPublisher.class=com.netscape.certsrv.ldap.
FileBasedPublisher
ca.publish.publisher.impl.LdapCaCertPublisher.class=com.netscape.certsrv.ldap.
LdapCaCertPublisher
ca.publish.publisher.impl.LdapCrlPublisher.class=com.netscape.certsrv.ldap.
LdapCrlPublisher
ca.publish.publisher.impl.LdapUserCertPublisher.class=com.netscape.certsrv.ldap.
LdapUserCertPublisher
ca.publish.publisher.impl.ValiCertPublisher.class=com.valicert.publisher.VcPublisher
```

```

ca.publish.publisher.instance.LdapCaCertPublisher.caCertAttr=caCertificate;binary
ca.publish.publisher.instance.LdapCaCertPublisher.caObjectClass=certificationAuthority
ca.publish.publisher.instance.LdapCaCertPublisher.pluginName=LdapCaCertPublisher
ca.publish.publisher.instance.LdapCrlPublisher.crlAttr=certificateRevocationList;binary
ca.publish.publisher.instance.LdapCrlPublisher.pluginName=LdapCrlPublisher
ca.publish.publisher.instance.LdapUserCertPublisher.certAttr=userCertificate;binary
ca.publish.publisher.instance.LdapUserCertPublisher.pluginName=LdapUserCertPublisher
ca.publish.rule.impl.Rule.class=com.netscape.certsrv.ldap.LdapRule

```

```

ca.publish.rule.instance.LdapCaCertRule.enable=true
ca.publish.rule.instance.LdapCaCertRule.mapper=LdapCaCertMap
ca.publish.rule.instance.LdapCaCertRule.pluginName=Rule
ca.publish.rule.instance.LdapCaCertRule.predicate=
ca.publish.rule.instance.LdapCaCertRule.publisher=LdapCaCertPublisher
ca.publish.rule.instance.LdapCaCertRule.type=ca

```

```

ca.publish.rule.instance.LdapCrlRule.enable=true
ca.publish.rule.instance.LdapCrlRule.mapper=LdapCrlMap
ca.publish.rule.instance.LdapCrlRule.pluginName=Rule
ca.publish.rule.instance.LdapCrlRule.predicate=
ca.publish.rule.instance.LdapCrlRule.publisher=LdapCrlPublisher
ca.publish.rule.instance.LdapCrlRule.type=crl

```

```

ca.publish.rule.instance.LdapObjSignCertRule.enable=true
ca.publish.rule.instance.LdapObjSignCertRule.mapper=LdapUserCertMap
ca.publish.rule.instance.LdapObjSignCertRule.pluginName=Rule
ca.publish.rule.instance.LdapObjSignCertRule.predicate=
ca.publish.rule.instance.LdapObjSignCertRule.publisher=LdapUserCertPublisher
ca.publish.rule.instance.LdapObjSignCertRule.type=objSignClient

```

```

ca.publish.rule.instance.LdapUserCertRule.enable=true
ca.publish.rule.instance.LdapUserCertRule.mapper=LdapUserCertMap
ca.publish.rule.instance.LdapUserCertRule.pluginName=Rule
ca.publish.rule.instance.LdapUserCertRule.predicate=
ca.publish.rule.instance.LdapUserCertRule.publisher=LdapUserCertPublisher
ca.publish.rule.instance.LdapUserCertRule.type=client

```

```

ca.signing.cacertnickname=caSigningCert cert-testCA
ca.signing.defaultSigningAlgorithm=MD5withRSA
ca.signing.tokenname=Internal Key Storage Token
cms.version=4.22

```

```

dbs.ldap=internaldb
dbs.newSchemaEntryAdded=true
dbs.nextSerialNumber=1

```

```

eeGateway._000=##
eeGateway._001=## End Entity Gateway
eeGateway._002=##

```

Modifying the Configuration

```
eeGateway.authority=ca
eeGateway.docRoot=/usr/netscape/cert-testCA/web/ee
eeGateway.dynamicVariables=serverdate=serverdate(), subsystemname=subsystemname(),
    http=http()
eeGateway.enableConnector=true
eeGateway.keepAliveOn=true
eeGateway.mimeTypeConf=/usr/netscape/cert-testCA/config/mime.types
eeGateway.numServices=2
eeGateway.service0=http
eeGateway.service1=https
eeGateway.http.backlog=15
eeGateway.http.enable=true
eeGateway.http.port=4603
eeGateway.http.type=http
eeGateway.https.backlog=15
eeGateway.https.nickName=Server-Cert cert-testCA
eeGateway.https.port=4604
eeGateway.https.type=https

internaldb._000=##
internaldb._001=## Internal Database
internaldb._002=##

    internaldb.maxConns=15
    internaldb.minConns=3
    internaldb.ldapauth.authType=BasicAuth
    internaldb.ldapauth.bindDN=cn=Directory Manager
    internaldb.ldapauth.bindPWPrompt=Internal LDAP Database
    internaldb.ldapconn.host=testCA.siroe.com
    internaldb.ldapconn.port=3602
    internaldb.ldapconn.secureConn=false

jobsScheduler._000=##
jobsScheduler._001=## jobScheduler
jobsScheduler._002=##

    jobsScheduler.enabled=false
    jobsScheduler.interval=1
    jobsScheduler.impl.RenewalNotificationJob.class=com.netscape.certsrv.jobs.
        RenewalNotificationJob
    jobsScheduler.impl.RequestInQueueJob.class=com.netscape.certsrv.jobs.
        RequestInQueueJob
    jobsScheduler.impl.UnpublishExpiredJob.class=com.netscape.certsrv.jobs.
        UnpublishExpiredJob
    jobsScheduler.job.certRenewalNotifier.cron=0 3 * * 1-5
    jobsScheduler.job.certRenewalNotifier.emailSubject=Certificate Renewal Notification
    jobsScheduler.job.certRenewalNotifier.emailTemplate=/usr/netscape/cert-testCA/emails/
        rnJob1.txt
    jobsScheduler.job.certRenewalNotifier.enabled=false
    jobsScheduler.job.certRenewalNotifier.notifyEndOffset=30
    jobsScheduler.job.certRenewalNotifier.notifyTriggerOffset=30
    jobsScheduler.job.certRenewalNotifier.pluginName=RenewalNotificationJob
    jobsScheduler.job.certRenewalNotifier.senderEmail=
    jobsScheduler.job.certRenewalNotifier.summary.emailSubject=Certificate Renewal
```

```

Notification Summary
jobsScheduler.job.certRenewalNotifier.summary.emailTemplate=/usr/netscape/cert-testCA/
  emails/rnJoblSummary.txt
jobsScheduler.job.certRenewalNotifier.summary.enabled=true
jobsScheduler.job.certRenewalNotifier.summary.itemTemplate=/usr/netscape/
  cert-testCA/emails/rnJoblItem.txt
jobsScheduler.job.certRenewalNotifier.summary.recipientEmail=
jobsScheduler.job.certRenewalNotifier.summary.senderEmail=
jobsScheduler.job.requestInQueueNotifier.cron=0 0 * * 0
jobsScheduler.job.requestInQueueNotifier.enabled=false
jobsScheduler.job.requestInQueueNotifier.pluginName=RequestInQueueJob
jobsScheduler.job.requestInQueueNotifier.subsystemId=ca
jobsScheduler.job.requestInQueueNotifier.summary.emailSubject=Requests in Queue
  Summary Report
jobsScheduler.job.requestInQueueNotifier.summary.emailTemplate=/usr/netscape/
  cert-testCA/emails/riqlSummary.html
jobsScheduler.job.requestInQueueNotifier.summary.enabled=true
jobsScheduler.job.requestInQueueNotifier.summary.recipientEmail=
jobsScheduler.job.requestInQueueNotifier.summary.senderEmail=
jobsScheduler.job.unpublishExpiredCerts.cron=0 0 * * 6
jobsScheduler.job.unpublishExpiredCerts.enabled=false
jobsScheduler.job.unpublishExpiredCerts.pluginName=UnpublishExpiredJob
jobsScheduler.job.unpublishExpiredCerts.summary.emailSubject=Expired Certs
  Unpublished Summary
jobsScheduler.job.unpublishExpiredCerts.summary.emailTemplate=/usr/netscape/
  cert-testCA/emails/euJobl.html
jobsScheduler.job.unpublishExpiredCerts.summary.enabled=true
jobsScheduler.job.unpublishExpiredCerts.summary.itemTemplate=/usr/netscape/
  cert-testCA/emails/euJoblItem.html
jobsScheduler.job.unpublishExpiredCerts.summary.recipientEmail=
jobsScheduler.job.unpublishExpiredCerts.summary.senderEmail=

jss._000=##
jss._001=## JSS
jss._002=##

jss.certdb=/usr/netscape/cert-testCA/config/cert7.db
jss.enable=true
jss.keydb=/usr/netscape/cert-testCA/config/key3.db
jss.moddb=/usr/netscape/admin-serv/config/secmodule.db
jss.ssl.cipherfortezza=true
jss.ssl.cipherpref=
jss.ssl.cipherversion=cipherdomestic

logAudit._000=##
logAudit._001=## Logging
logAudit._002=##
log.Error._000=##
log.Error._001=## Logging
log.Error._002=##
log.System._000=##
log.System._001=## Logging
log.System._002=##

```

Modifying the Configuration

```
log.impl.NTEventLog.class=com.netscape.certsrv.logging.NTEventLog
log.impl.file.class=com.netscape.certsrv.logging.RollingLogFile

log.instance.Audit.bufferSize=512
log.instance.Audit.enable=true
log.instance.Audit.expirationTime=2592000
log.instance.Audit.fileName=/usr/netscape/cert-testCA/logs/audit
log.instance.Audit.flushInterval=5
log.instance.Audit.level=1
log.instance.Audit.maxFileSize=100
log.instance.Audit.pluginName=file
log.instance.Audit.rolloverInterval=2592000
log.instance.Audit.type=audit

log.instance.Error.bufferSize=512
log.instance.Error.enable=true
log.instance.Error.expirationTime=2592000
log.instance.Error.fileName=/usr/netscape/cert-testCA/logs/error
log.instance.Error.flushInterval=5
log.instance.Error.level=3
log.instance.Error.maxFileSize=100
log.instance.Error.pluginName=file
log.instance.Error.rolloverInterval=2592000
log.instance.Error.type=system

log.instance.NTAudit.NTEventSourceName=cert-testCA
log.instance.NTAudit.enable=true
log.instance.NTAudit.level=1
log.instance.NTAudit.pluginName=NTEventLog
log.instance.NTAudit.type=audit

log.instance.NTSystem.NTEventSourceName=cert-testCA
log.instance.NTSystem.enable=true
log.instance.NTSystem.level=2
log.instance.NTSystem.pluginName=NTEventLog
log.instance.NTSystem.type=system

log.instance.System.bufferSize=512
log.instance.System.enable=true
log.instance.System.expirationTime=2592000
log.instance.System.fileName=/usr/netscape/cert-testCA/logs/system
log.instance.System.flushInterval=5
log.instance.System.level=3
log.instance.System.maxFileSize=100
log.instance.System.pluginName=file
log.instance.System.rolloverInterval=2592000
log.instance.System.type=system
```

```

oidmap.auth_info_access.class=com.netscape.certsrv.cert.AuthInfoAccessExtension
oidmap.auth_info_access.oid=1.3.6.1.5.5.7.1.1
oidmap.challenge_password.class=com.netscape.certsrv.cmsgateway.cert.
    crs.ChallengePassword
oidmap.challenge_password.oid=1.2.840.113549.1.9.7
oidmap.extended_key_usage.class=com.netscape.certsrv.cert.ExtendedKeyUsageExtension
oidmap.extended_key_usage.oid=2.5.29.37
oidmap.extensions_requested_pkcs9.class=com.netscape.certsrv.cmsgateway.cert.
    crs.ExtensionsRequested
oidmap.extensions_requested_pkcs9.oid=1.2.840.113549.1.9.14
oidmap.extensions_requested_vsgn.class=com.netscape.certsrv.cmsgateway.cert.
    crs.ExtensionsRequested
oidmap.extensions_requested_vsgn.oid=2.16.840.1.113733.1.9.8
oidmap.netscape_comment.class=netscape.security.x509.NSCCCommentExtension
oidmap.netscape_comment.oid=2.16.840.1.113730.1.13
oidmap.ocsp_no_check.class=com.netscape.certsrv.cert.OCSPNoCheckExtension
oidmap.ocsp_no_check.oid=1.3.6.1.5.5.7.48.1.5

os.serverName=cert-testCA
os.userid=nobody

radm._000=##
radm._001=## Remote Admin
radm._002=##

    radm.keepAliveOn=true
    radm.mimeTypeConf=/usr/netscape/cert-testCA/config/mime.types
    radm.numServices=1
    radm.service0=https
    radm.https.backlog=15
    radm.https.maxThreads=10
    radm.https.minThreads=3
    radm.https.nickName=Server-Cert cert-testCA
    radm.https.port=4606
    radm.https.timeout=0
    radm.https.type=https

smtp.host=localhost
smtp.port=25

subsystem._000=##
subsystem._001=## Loadable Subsystems
subsystem._002=##

    subsystem.0.class=com.netscape.certsrv.ca.CertificateAuthority
    subsystem.0.id=ca
    subsystem.1.class=com.netscape.certsrv.cmsgateway.EEGateway
    subsystem.1.id=eeGateway

usrgrp._000=##
usrgrp._001=## User/Group
usrgrp._002=##

    usrgrp.ldap=internaldb

```

Road Map to Configuring Subsystems

This section outlines how to configure an instance of Certificate Management System and indicates where to find the information required to accomplish the task.

Step 1. Check Which Subsystems are Installed in the Instance

Log in to the CMS window for the CMS instance you installed, and check the navigation tree to see which subsystems are installed in that instance; this way you will know the subsystems you should configure. To log in to the CMS window, see “Logging In to the CMS Window” on page 351.

Step 2. Check the Port Numbers

Check the port numbers assigned for administration, agent, and end-entity operations. Make the appropriate modifications, if necessary. Keep in mind that all subsystems installed in an instance use the same ports, but can be configured to listen on different IP addresses. For instructions, see “Configuring Port Numbers” on page 384.

Step 3. Verify Key Pair and Certificates

When you install a CMS instance, the server prompts you to create the certificates required for the subsystems in that instance to function. You should check the certificates used by each subsystem, and determine if you need to get additional certificates, use hardware tokens, and so on.

- Each subsystem in an instance has a set of certificates that it uses for specific purposes. Understand how and when the subsystem uses its certificates. For details, see “Keys and Certificates for the Main Subsystems” on page 450.
- Determine if you want to generate any new certificates. For example, if you have two subsystems installed in an instance, you may want them to use separate SSL server certificates; by default, there’s only one SSL server certificate per instance. For details, see “Getting New Certificates for the Subsystems” on page 507.
- Determine if you want to use hardware tokens for generating and storing these certificates. If required, install new hardware tokens. For details, see “Tokens for Storing CMS Keys and Certificates” on page 464.
- Determine if you want to renew any of the existing certificates. For example, if you have issued certificates with very short validity periods, you might want to renew them. For details, see “Renewing Certificates for the Subsystems” on page 515.

- Check the certificate database to see which CA certificates are trusted. Delete any unwanted CA certificates, change the trust settings of CA certificates that you don't want to trust to *untrusted*, and install any new CA certificate or certificate chains. For details, see “Managing the Certificate Database” on page 523.

Step 4. Set up Privileged Users

Set up required administrators and agents. This way you can delegate administration and agent tasks to other individuals. For details, see “Setting Up Privileged Users” on page 413.

If you have installed remote Registration Managers that have certificates signed by third-party CAs (that is, not by a Certificate Manager), you should add their certificates to the Certificate Manager's database to facilitate SSL client authenticated communication. For details, see “Setting Up Trusted Managers” on page 423.

Step 5. Customize End-Entity and Agent Forms

End entities can interact with the Certificate Manager and Registration Manager with the help of end-entity forms; end entities cannot interact with the Data Recovery Manager. Similarly, agents can interact with the appropriate subsystem using the agent forms. Certificate Management System provides HTML forms-based interfaces for end entities and agents out of the box. For details, see *CMS Customization Guide*.

Determine which forms you want to use for end-entity enrollment and whether they require any customization. You may also use your own forms for this purpose, provided you add the required JavaScript.

When customizing end-entity forms, keep in mind the authentication method—manual or automated—you want to employ for your end entities.

Step 6. Setup Authentication for End Users

Depending on how you've deployed Certificate Management System, you may need to do this for a Certificate Manager or Registration Manager, or for both. For example, you may have a PKI setup in which Registration Managers act as front ends to Certificate Managers—that is, end entities interact with Registration Managers only; they do not interact with the Certificate Manager.

Determine which of the authentication plug-in module is suitable for your users and then configure the Certificate Manager or Registration Manager to use that authentication method; see “Configuring Authentication for End-User Enrollment” on page 545.

Step 7: Enable Event-Driven Notifications

You can also configure both Certificate Manager and Registration Manager to send email notifications automatically to end entities, agents, or administrators when certain events occur. Unlike jobs that are executed at preconfigured schedule, these notifications are event-driven—that is, whenever an event occurs, the server notifies the user. Notifiable events include certificate issuance and pending requests in an agent queue.

Decide if you want to turn on any of the notifications. For details, see “Configuring a Subsystem to Send Notifications” on page 583.

Step 8. Schedule Jobs

Each CMS instance includes a *Job Scheduler* component that can execute specific jobs at specified times. The Job Scheduler functions similar to a traditional Unix *cron* daemon in that it takes registered cron jobs and executes them at a preconfigured date and time.

During installation, a few jobs are already created and enabled. Jobs that you might want to schedule include email notifications of timed events (such as the expiration of a certificate) that require action on the part of users, and periodic activities such as removing expired certificates from the publishing directory. For scheduling jobs, follow the instructions in “Configuring a Subsystem to Run Automated Jobs” on page 589.

Step 9. Set up Policies

Each subsystem in a CMS instance has its own policy processor. If you have installed more than one subsystem in an instance, you should apply the instructions in this section to each subsystem. That is, you should configure the Certificate Manager and Registration Manager for certificate formulation, issuance, renewal, and revocation policies. Similarly, configure the Data Recovery Manager for key archival and recovery policies. To understand policy, see “Introduction to Policy” on page 603.

1. During installation, a few policy rules are already created and enabled. Check each policy rule and decide whether you want to use it. If you don't, you can either disable it or delete it altogether from the configuration. For those rules that you want to use, check the configuration parameter values and make changes as appropriate.
2. Determine if you want to add any new policy rules. Check the built-in policy plug-in modules to see if they can be used to create the rules you want. You can also plug-in your own modules in the CMS framework and use them.
3. Add new rules, if required.

For instructions to do all of the above tasks, see “Configuring Policy Rules for a Subsystem” on page 613.

Step 10. Set up Publishing

This step is optional, and is applicable to the Certificate Manager only—you need to do this only if you want the Certificate Manager to publish certificates and CRLs to any of the supported repositories.

- To configure a Certificate Manager to publish certificates and CRLs to an LDAP-compliant directory, such as iPlanet Directory Server, see “Configuring a Certificate Manager to Publish Certificates and CRLs” on page 639.
- To configure a Certificate Manager to publish certificates and CRLs to a flat file, see “Configuring Certificate Manager to Publish to Files” on page 691.
- To configure a Certificate Manager to publish CRLs to the Online Certificate Status Manager (an online validation authority), see “Setting Up a Remote OCSP Responder” on page 732.

Step 11. Set up Key Archival and Recovery

If you have installed the Data Recovery Manager, follow the instructions in “Configuring Key Archival and Recovery Process” on page 775 and set up archival and recovery for end users’ encryption private keys.

Step 12. Set up Logging

Each instance of Certificate Management System maintains extensive audit, error, and system logs. By looking at these logs, you can monitor a server’s activities. Also, by configuring these logs, you can control the information that gets written to the log files. Because Certificate Management System maintains the log files in the file system of the host machine, it is important that you configure the logs appropriately (so that the host machine doesn’t get overloaded). Be sure to read “Introduction to Logs” on page 789; this chapter will help you decide log configuration.

Once you decide the configuration for server logs, follow the information in “Configuring CMS Logs” on page 797 and configure all the three logs. Then, start monitoring the server’s activities as explained in “Monitoring CMS Logs” on page 803.

Step 13. Plan for Backing up CMS Configuration and Data

It is a good practice to periodically back up the CMS data on to some backup media. Creating backups will help you use them for data restoration in the event of data loss. For details, r details, see Chapter 6, “Backing Up and Restoring Data” of *CMS Command-Line Tools Guide*.

Setting Up Ports

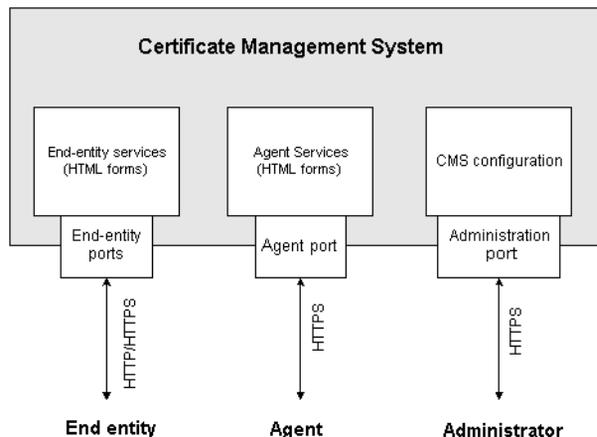
Subsystems installed in an instance of iPlanet Certificate Management Server (CMS) share certain configuration information. For example, they use the same administration, agent, and end-entity ports; internal database for data storage; mail server for automated notifications; internal token and trust database for PKI operations; SSL ciphers during SSL negotiation; privileged users; and log files to log messages to. This chapter explains how to configure the ports for a CMS instance.

The chapter has the following sections:

- CMS Ports (page 381)
- Configuring Port Numbers (page 384)

CMS Ports

Certificate Management System listens to different ports for requests from different users. As illustrated in Figure 11-1, it listens to the administration port, the agent port, and end-entity ports.

Figure 11-1 CMS ports for administration, agent, and end-entity operations

When choosing ports for Certificate Management System, be sure to choose ports that are unique on the host system—that is, no other application can be using, or attempting to use, the port numbers you assign to Certificate Management System. To verify that a port is available for use, check the appropriate file for your operating system; port numbers for network-accessible services are usually maintained in a file named `services`. (On Unix, if you are not running as `root` or `superuser` when you install or start the server, you will have to use a port number higher than 1024.)

Remote Administration Port

The administration port is an SSL (encrypted) port at which Certificate Management System listens to requests from its administration interface; administrators make these requests from the CMS window. When you install Certificate Management System, it assigns a random number (greater than 1024) as the administration port number. You can change this port number at any time, to any number between 1 and 65535. For security reasons you should consider changing the administration port number periodically.

Agent Port

The agent port is an SSL (encrypted) port at which Certificate Management System listens to requests from agents; agents make these requests from the appropriate Agent Services interface.

- The Certificate Manager and Registration Manager agents use the agent port to process certificate issuance and management requests from end entities and to perform certain other privileged operations over HTTPS.
- Data Recovery Manager agents use the agent port for recovering end users' encryption private keys over HTTPS.

Agent functions always require SSL client authentication. For a brief list of supported agent operations, see “Agent Services Interface” on page 68.

When you install Certificate Management System, it assigns a random number (greater than 1024) as the agent port number and prompts you to change it, if necessary; the port number can be any number between 1 and 65535. The number you choose for the agent port affects your agent users—all agents access Certificate Management System by specifying the name of the server (the CMS instance) and the agent port number in the URL. For example, if you choose port number 4430, the URL would look like this:

```
https://<hostname>:4430/<subsystem>
```

`<hostname>` is in the form `<machine_name>.<your_domain>.<domain>`

`<subsystem>` is a prefix identifying the subsystem that hosts the agent interface: `ca` for the Certificate Manager, `ra` for the Registration Manager, `kra` for the Data Recovery Manager, and `ocsp` for Online Certificate Status Manager.

For example, the URL to a Certificate Manager agent interface would look like this:

```
https://demoCA.siroe.com:5600/ca
```

If you change the agent port number, be sure to inform your agent users.

End-Entity Ports

For requests from end entities, Certificate Management System can listen to two ports, an SSL (encrypted) port and a non-SSL port. End entities make these requests from the end entity services interface; see “End-Entity Services Interface” on page 72.

Certificate Management System provides the following services through the HTTP and HTTPS ports:

- The HTTP port can be used to service end-entity-initiated PKI requests, such as enrollment, renewal, and revocation; enrollment requests can include requests from Cisco routers (using the CEP protocol). You have the choice of keeping this port enabled or disabled.
- The HTTPS port can be used to provide the following services for enforcing data privacy and client authentication:
 - End-entity-initiated PKI requests, such as enrollment, renewal, and revocation.
 - General certificate retrieval requests, such as retrieving a single certificate identified by a serial number, listing certificates based on certain criteria (for example, an LDAP search filter defined over standard attributes), and getting a CA's certificate chain.

Similar to the HTTP port, you can enable or disable the HTTPS port. For example, if you don't want end-entity interaction with a Certificate Manager, you can disable the HTTPS port. For details, see "Step 6. Enable End-Entity Interaction" on page 563.

Configuring Port Numbers

Configuring port numbers for a CMS instance involves two steps:

- Step 1. Specify the Port Number
- Step 2: Specify IP Addresses

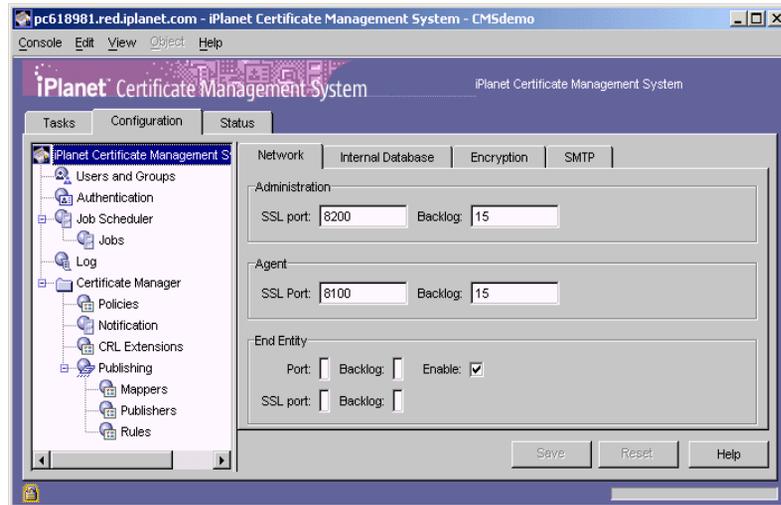
Step 1. Specify the Port Number

To change the administration, agent, or end-entity port numbers used by a CMS instance:

1. Log in to the CMS window (see "Logging In to the CMS Window" on page 351).

2. Select the Configuration tab.

The Network tab appears.



3. To change the administration port number, enter the port number in the Administration section:

SSL port. Type a TCP/IP port number. Certificate Management System uses this port for SSL-enabled communications with the CMS window—that is, HTTPS requests from administrators. Make sure the port number you specify is unique on the host system.

Backlog. Type the number of connections that can be waiting to be serviced at the administration port. The default number is 15. The number you enter in this field is passed to the operating system's `listen()` call.

To change the agent port number, enter the port number in the Agent section:

SSL port. Type a TCP/IP port number. Certificate Management System uses this port for SSL-enabled communications with the Agent Services interface—that is, HTTPS requests from agents. Make sure the port number you specify is unique on the host system.

Backlog. Type the number of connections that can be waiting to be serviced at the agent port. The default number is 15. The number you enter in this field is passed to the operating system's `listen()` call.

4. To change the end-entity port numbers, enter the port numbers in the End Entity section.

Certificate Management System is capable of simultaneous SSL and non-SSL communications at the end-entity port. This means that you do not have to choose between SSL and non-SSL communications; you can use both at the same time. But if you prefer, you can disable the non-SSL port by unchecking the “Enable” option.

Port. Type a TCP/IP port number that is unique on the host system. Certificate Management System uses this port for non-SSL communications with the end entity services interface.

This port is provided to allow enrollments of end entities that do not support SSL; for example, HTTP requests from end entities such as routers. You can use the Enable check box to turn this port on or off. Keep in mind that if this port is enabled, end entities will be able to enroll over HTTP too, which means their certificate requests could be intercepted and replayed to the server.

If the CMS instance includes a Certificate Manager and if the Certificate Manager is configured to service OCSP requests from OCSP-compliant clients, then this port must be enabled so that OCSP-compliant clients can successfully query the Certificate Manager for the revocation status of a certificate. For details, see “Setting Up a Certificate Manager with OCSP Service” on page 719.

Backlog. Type the number of connections that can be waiting to be serviced at the end entity HTTP port. The default number is 15. The number you enter in this field is passed to the operating system’s `listen()` call.

Enable. This check box allows you to enable or disable the HTTP port. Uncheck the option if you want to disable the port.

For issuing certificates to routers (using the CEP protocol), the port must be enabled. For details, see Chapter 25, “Setting Up CEP Enrollment.”

SSL port. Type a TCP/IP port number. Certificate Management System uses this port for SSL-enabled communications with the end entity services interface (that is, HTTPS requests from end entities during certificate enrollment, renewal, and revocation). Make sure the port number you specify is unique on the host system.

If you don’t want end-entity interaction with a subsystem, for example, if you don’t want end entities to interact with a Certificate Manager, you can disable this port too (in addition to the HTTP port). See “Step 6. Enable End-Entity Interaction” on page 563.

Backlog. Type the number of connections that can be waiting to be serviced at the end-entity HTTPS port. The default number is 15. The number you enter in this field is passed to the operating system's `listen()` call.

5. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Step 2: Specify IP Addresses

This step is optional.

You can configure CMS instances to listen to specific IP addresses. For example, you can install the Certificate Manager and Data Recovery Manager on a single host, in separate instances, and then configure the instances so that the Certificate Manager is served on one IP address and the Data Recovery Manager is served on another address.

To clarify this further, consider the machine that hosts the Certificate Manager and Data Recovery Manager has two Ethernet cards that respond to the IP addresses 197.1.137.97 and 197.1.137.98. You can set up the Certificate Manager to listen to port 443 for the IP address 197.1.137.97 and the Data Recovery Manager to listen to port 443 for the IP address 197.1.137.98.

To configure a CMS instance to listen to specific IP addresses:

1. Stop the CMS instance; see “Stopping Certificate Management System” on page 330.
2. Open the configuration file in a text editor; to locate the file, see “Locating the Configuration File” on page 358.
3. Add one or more of the following as appropriate:
 - For remote administration port, add this line: `radm.https.host=`
 - For agent port, add this line: `agentGateway.https.host=`
 - For end-entity HTTPS port, add this line: `eeGateway.https.host=`
 - For end-entity HTTP port, add this line: `eeGateway.http.host=`

4. Add the IP address or the host name or interface name as the value for the parameter you just added. For example,
 - o If you entered an IP address as the value, the parameter would look similar to this: `radm.https.host=197.1.137.98`
 - o If you entered the host name as the value, the parameter would look similar to this: `radm.https.host=cert.siroe.com`
5. If necessary, repeat step 4 for the other ports.
6. Save your changes, and close the configuration file.
7. Start the CMS instance; see “Starting Certificate Management System” on page 322.

Setting Up Internal Database

Subsystems installed in an instance of iPlanet Certificate Management Server (CMS) share certain configuration information. For example, they use the same administration, agent, and end-entity ports; internal database for data storage; mail server for automated notifications; internal token and trust database for PKI operations; SSL ciphers during SSL negotiation; privileged users; and log files to log messages to. This chapter explains how to configure the internal database for a CMS instance.

The chapter has the following sections:

- Internal Database (page 389)
- Configuring the Internal Database (page 390)

Internal Database

Certificate Management System performs various certificate and key-management functions in response to the requests it receives. These functions include the following:

- Storing and retrieving of certificate issuance requests
- Storing and retrieving of certificate records
- Storing of CRLs
- Storing and retrieving of end users' encryption private key records

To fulfill these functions, Certificate Management System maintains a persistent store—a preconfigured iPlanet Directory Server—referred to as the *internal database* or *local database*. The internal database is installed automatically as a part of the CMS installation. It is used as an embedded database exclusively by Certificate Management System and can be managed using Directory management tools that come with Directory Server.

The Directory Server instance used for the internal database is different from the LDAP-compliant directory that you use to manage your corporatewide data (users and groups, their certificates, CRLs, and so on).

- In iPlanet Console, you can distinguish an internal database instance from other Directory Server instances. It is in this form:

```
<cms_instance_id>-db
```

`<cms_instance_id>` is the ID of the CMS instance that is using the database. You first specified this when you installed this server.

- If you check the files installed under `<server_root>`, the internal database instance appears like this: `slapd-<cms_instance_id>-db`

Keep in mind that the subsystems use the database for storing different objects. A Certificate Manager stores all the data, certificate issuance requests, certificates, CRLs, and related information; a Registration Manager only stores the certificate issuance requests it receives; and a Data Recovery Manager only stores key records and related data.

Configuring the Internal Database

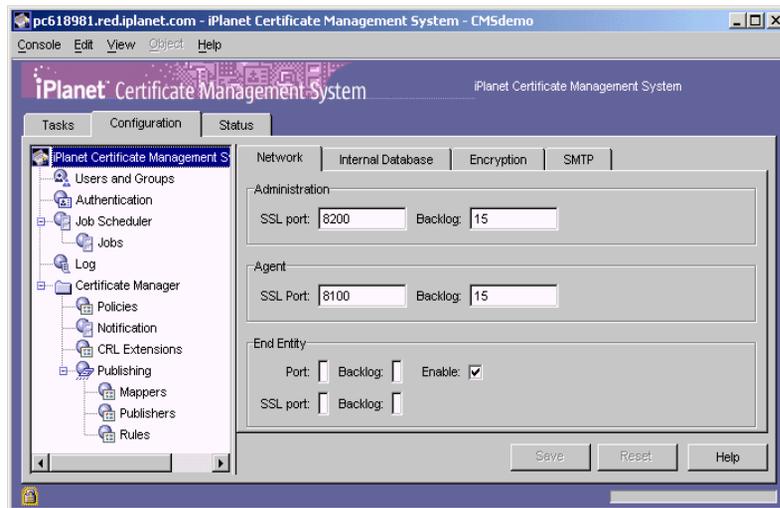
Each instance of Certificate Management System uses a iPlanet Directory Server instance as its internal database. All the subsystems that were installed in a CMS instance use the same Directory Server instance to store their data. For example, if you installed a Certificate Manager and Data Recovery Manager together, they use the same internal database for data storage.

CAUTION The internal database schema is preconfigured for storing CMS data only. Do not make any changes to it or configure Certificate Management System to use any other LDAP directory. Doing so can result in loss of data. Also, do not attempt to use this database for any other purpose.

Step 1. Identify the Directory Server Instance

To identify the Directory Server instance that a CMS instance should use as its internal database:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab, and then in the right pane, select the Internal Database tab.



3. Identify a Directory Server instance by providing the following details:

Host name. Type the full host name of the machine on which iPlanetDirectory Server is installed. Certificate Management System uses this name to access the directory. The format for the host name is as follows:

```
<machine_name>.<your_domain>.<domain>
```

By default, the host name of the Directory Server instance being used as the internal database is shown as `localhost` instead of the actual host name (for example, `certificates.netscape.com`). This is done on purpose to insulate the internal database from being visible outside the system—that is, a server on `localhost` can only be accessed from the local machine. Thus, the default configuration minimizes the risk of someone connecting to this Directory Server instance from outside the local machine.

You can configure the host name to something other than `localhost` if you know what you are doing and you think you can limit the visibility of the internal database to a local subnet. For example, if you installed Certificate Management System and Directory Server on separate machines for load balancing, you will have to specify the host name of the machine in which Directory Server is installed.

Port number. Type a TCP/IP port number; Certificate Management System uses this port for non-SSL communications with the Directory Server instance that is functioning as the internal database. Make sure that the port you specify is unique on the host system.

Directory manager DN. Type the distinguished name (DN) of an entry in your LDAP directory that has read and write permission to the entire directory tree. Certificate Management System will use this DN when it accesses the directory tree to communicate with the directory. Keep in mind that the access control set up for this DN determines whether Certificate Management System can communicate with the directory. Typically, you would want to enter the directory manager's DN (the *root DN*) because this DN will have read/write permission to the entire directory tree.

4. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Step 2. Restrict Access to the Internal Database

This step is optional.

iPlanet Console displays an entry or icon for the Directory Server instance that Certificate Management System uses as its internal database. You can distinguish an internal database instance from other Directory Server instances. It is in this form: `slapd-<cms_instance_id>-db`

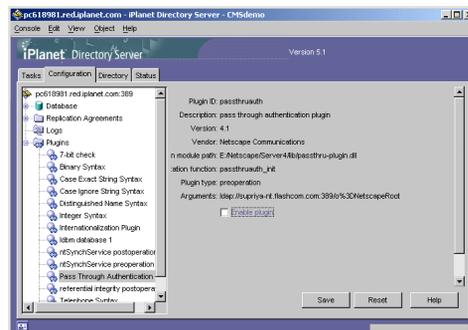
Unlike the CMS window, access to which is restricted to users with *CMS administrator* privileges, the Directory Server window can be accessed by the person who has privileges to access iPlanet Console. That is, this person can open the Directory Server window for the internal database and make changes to the data stored there. For example, this person can make changes to the CMS administrators group, such as deleting existing users and adding entries for self.

If you are concerned about this, you can restrict access to the internal database to only those users who know its Directory Manager DN and corresponding password. You can change this password by modifying the single sign-on password cache. For instructions, check the section that explains how to change the password of an entry in the password cache in Chapter 2, “Password Cache Utility” of *CMS Command-Line Tools Guide*.

1. Log in to iPlanet Console (see “Logging In to the CMS Window” on page 351).
2. In the Console tab, select the server group that contains the CMS instance you want.
3. Select the entry that corresponds to the internal database to which you want to restrict access, and click Open.

The Directory Server window appears.

4. Select the Configuration tab.
5. In the navigation tree, expand Plugins, and then select Pass Through Authentication.
6. In the right pane, uncheck or disable the “Enable plugin” option.



Click Save to save your changes.

You are prompted to restart the server.

7. Click the Tasks tab and click “Restart the Directory Server.”
8. Close the Directory Server window.

9. When the server is restarted, from iPlanet Console, open the Directory Server window.

The “Login to Directory” dialog box appears; the Distinguished Name field displays the Directory Manager DN and you’re required to enter the password that corresponds to this entry.



The Directory Server window (for the internal database) opens only if you enter the correct password.

Managing Privileged Users and Groups

Privileged users are users who are designated to perform privileged operations on iPlanet Certificate Management Server (CMS); these operations are privileged because no one else can perform them. You assign *privileged-user* status to a user by storing the user's login information in the internal database of Certificate Management System, associating the user's login information with a personal certificate (if the user is an agent or a trusted manager), and granting access permissions to various CMS resources by adding the user to appropriate groups.

This chapter describes the types of privileged users you need to set up for a CMS instance, what each user does, how Certificate Management System identifies these users, and how you create and manage these users. The chapter also describes what a group is and discusses the groups that Certificate Management System provides by default.

The chapter has the following sections:

- Privileged-User Types and Responsibilities (page 396)
- Groups and Their Privileges (page 409)
- Setting Up Privileged Users (page 413)
- Changing Privileged-User Information (page 444)
- Deleting a Privileged User (page 448)

Privileged-User Types and Responsibilities

After you install Certificate Management System, your first task is to set up privileged users. There are three types of privileged users: administrators, agents, and trusted managers.

- *Administrators* are users (people) who manage server-specific tasks for the CMS managers, the Certificate Manager, Registration Manager, Data Recovery Manager, and Online Certificate Status Manager. For details, see “Administrators” on page 396.
- *Agents* are users (people) who manage the request queues for the CMS managers. For details, see “Agents” on page 397.
- *Trusted managers* are CMS subsystems that are connected to other subsystems and that are trusted to perform certain activities for them. For example, you might set up a Registration Manager to screen end-entity certificate requests for a Certificate Manager. Because the Certificate Manager trusts the Registration Manager, it approves all certificate requests received from this Registration Manager. For details, see “Trusted Managers” on page 405.

The role of a privileged user—whether administrator, agent, or trusted manager—is determined by the group to which the user belongs. This is explained in “Groups and Their Privileges” on page 409.

Administrators

Administrators are users who have been assigned CMS administration privileges—permission to access the CMS window and perform all the system administration tasks defined there. You assign these privileges to users by adding them to the internal database and assigning membership in a group called `Administrators` that Certificate Management System creates during installation.

For each CMS instance, the server must have at least one administrator. You can also have more than one individual administering the server.

During installation, Certificate Management System prompts you to provide information for creating the first user entry in the `Administrators` group. Following installation, therefore, this group has a single user entry. For more information about this group, see “Group for Administrators” on page 409.

Certificate Management System authenticates users with administrator-level privileges based on its built-in authentication mechanism. This is explained in “Authentication of Administrators” on page 532.

Agents

Agents are users who have been assigned end-entity certificate- and key-management privileges. Certificate Management System defines four agent roles, one for each of its subsystems: Certificate Manager agents, Registration Manager agents, Data Recovery Manager agents, and Online Certificate Status Manager agents.

Agents interact with the corresponding CMS manager to manage operations such as these:

- List, approve, and reject pending certificate issuance and renewal requests
- List certificates
- Search for certificates
- Revoke end-entity certificates
- Manually update certificates and CRLs stored in a publishing directory
- Manage key archival and retrieval requests
- Manually add CRLs to the Online Certificate Status Manager
- See the list of OCSP requests processed by the Online Certificate Status Manager

For a complete list of agent tasks, see *CMS Agent's Guide*. To locate this guide, see “Where to Go for Related Information” on page 29.

All agents perform their tasks through HTML forms-based interfaces. The HTML forms an agent uses to manage a specific subsystem are grouped together and named after the subsystem they represent. For example, the forms-based interface provided for the Certificate Manager is called *Certificate Manager Agent Services* (see Figure 13-1). For more information, see “Agent Services Interface” on page 68.

Agents cannot access the CMS window and perform the tasks provided within the iPlanet Console framework—unless they are given *administrator* privileges.

Figure 13-1 Agents use the HTML forms-based interface called Agent Services

Each subsystem installed in a CMS instance must have at least one agent. You can also have more than one individual managing agent services.

You create agents by adding them to the internal database of a CMS instance, assigning membership in the appropriate agent groups, and identifying certificates that the agents must use for SSL client authentication to the subsystem (for it to service requests from the agents). For information about agents' certificates, see "Agent's Certificate for SSL Client Authentication" on page 399. For information on creating agents for a CMS instance, see "Setting Up Agents" on page 416.

During installation, Certificate Management System automatically creates appropriate groups with agent privileges for the CMS managers installed; for example, if you install a Certificate Manager and a Data Recovery Manager in a CMS instance, you'll see two agent groups. For more information about these groups, see "Groups for Agents" on page 410.

Certificate Management System authenticates users with agent-level privileges based on its built-in authentication mechanism. This is explained in "Authentication of Agents" on page 534.

Agent's Certificate for SSL Client Authentication

To make a user an agent for a subsystem, one of the things you must do is store the user's client (personal) certificate information in the internal database of the subsystem. For example, if you set up an agent for a Certificate Manager, you store the agent's client certificate in the internal database of that Certificate Manager. Then, when the subsystem receives a request from the agent, it uses this certificate to verify the authenticity of the request before servicing it. For details on how the subsystem verifies the authenticity of a request from an agent, see "Authentication of Agents" on page 534.

If the user you want to set up as an agent does not own a client certificate, ask the user to get one. Depending on your company's PKI policy, the user could get the client certificate from either an internally deployed CA or any public CA.

Keep in mind that the CA that signs your agents' certificates must be *trusted* by the subsystem that processes requests sent by these agents; for example, if your subsystems are set up not to trust public CAs, your agents should not get their certificates signed by public CAs. Make sure that the CA's certificate exists in the subsystem's certificate or trust database and that the certificate is valid and trusted. To check whether or not the CA's certificate exists in a subsystem's trust database, follow the instructions in "Viewing the Certificate Database Content" on page 523.

- If the CA's certificate isn't listed, follow the instructions in "Using the Wizard to Install a Certificate or Certificate Chain" on page 493 and add the certificate to the subsystem's certificate database.
- If the CA's certificate is listed but *untrusted*, follow the instructions in "Changing the Trust Settings of a CA Certificate" on page 526 and change the setting to *trusted*.

Getting an Agent's Certificate from a Public CA

The following general guidelines explain how a user can get a client certificate from a public CA and how you can copy that certificate (in base-64 encoded form) to the internal database of the appropriate subsystem:

1. The user sends a client certificate request to the public CA from the client machine that he or she will use to access the subsystem from the Agent Services interface. It is important that the user generate and submit this request from the machine she or he will use later to access the subsystem, because part of this request process generates a private key on the local machine. Alternatively, if location independence is required, the user can use a hardware token, such as a smart card, to generate and store the key pair (and the certificate when the user receives it from the public CA).

2. When the user receives the certificate from the public CA, the user imports the certificate into the web browser that he or she will use to access the subsystem. It is a good idea to ask the user to inform you that the certificate has been installed.
3. Ask the user to send you the certificate information sent by the public CA. In the information that you receive, locate the user's certificate in base-64 encoded form.

You can also get the user's certificate from the public CA that issued it. Access the public CA site, search for the user's certificate, and locate the certificate in base-64 encoded form.

4. Copy the base-64 encoded certificate, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines, to a text file.

The copied information should look similar to the following example:

```
-----BEGIN CERTIFICATE-----
MIICJzCCAZCgAwIBAgIBAzANBgkqhkiG9w0BAQQFADBCMSAwHgYDVQQKEXdOZXRz
Y2FwZSBDb21tdW5pYF0aW9uc2ngjhnMVQ2VydG1maWNhdGUgQXV0aG9yaXR5MB4X
DTk4MDgyNzE5MDAwMFoXDTk5MDIyMzE5MDAwMnBjdGngYoxIDAeBgNVBAoTF05ld
HNjYXB1IENvbW11bmljYXRpb25zMQ8wDQYDVQQLEWZQZW9wbGUxZjYwMTEjMCEG
CSqGSIb3DbnndGARYUc3Vwcm15YUBuZXRzY2FwZS5jb20wXDANBgkqhkiG9w0BAQE
FAANLADBIAKEAoYiYgthgtbbnjfngjnjgnagwJjAOBgNVHQ8BAf8EBAMCBLAwFAYJYI
ZIAYb4QgEj
-----END CERTIFICATE-----
```

5. Save the text file and use it to store a copy of the certificate in a subsystem's internal database (see "Step 3. Store the Agent's SSL Client Certificate in the Internal Database" on page 420).

Getting an Agent's Certificate from Certificate Management System

The following general instructions explain how a user can get a client certificate from Certificate Management System and how you can copy that certificate (in base-64 encoded form) to the internal database of a subsystem:

1. The user sends a client certificate request to Certificate Management System from the client machine that he or she will use to access the subsystem from the Agent Services interface. It is important that the user generate and submit this request from the machine he or she will use later to access the subsystem,

because part of this request process generates a private key on the local machine. Alternatively, if location independence is required, the user can also use a hardware token, such as a smart card, to generate and store the key pair (and the certificate when the user receives it from the public CA).

2. Depending on how your Certificate Management System is configured for certificate issuance, one of the following events happen:
 - If Certificate Management System is configured for manual certification, an issuing agent must process the request and approve it for issuance. Once the request is approved, the server issues the client certificate to the user.
 - If Certificate Management System is configured for automated certification and the request passes authentication and policy checks, the server automatically issues the client certificate to the user.
3. When the user receives the certificate, the user must import the certificate into the web browser that he or she will use to access the subsystem. It is a good idea to ask the user to inform you that the certificate has been installed.
4. After the user imports the certificate into the web browser, you need to copy the certificate (in base-64 encoded form) in order to be able to add it to a subsystem's internal database.

To copy an agent's certificate:

1. Open a web browser window.
2. Go to the Certificate Management System home page for end entities (by default, it is called *End Entity Registration Services*).

The default URL for this page is in this form:

```
http://<host_name>:<end_entity_HTTP_port> or
https://<host_name>:<end_entity_HTTPS_port>
```

In both cases, the <host_name> must be in this form:

```
<machine_name>.<your_domain>.<domain>
```

For example, the URL may look like this: <https://testCA.siroe.com>

3. Click the Retrieval tab.
4. In the left frame, click either the List Certificates or Search For Certificates link, and search for the user's certificate.

5. In the page listing the results of your search, click the Details button (next to the corresponding user's entry) to see detailed information about the certificate.
6. Scroll down to the section named "Installing This Certificate in a Client," which contains the user's certificate in base-64 encoded form.
7. Copy the base-64 encoded certificate, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines, to a text file.

The copied information should look similar to the following example:

```
-----BEGIN CERTIFICATE-----
MIICJzCCAZCgAwIBAgIBAzANBgkqhkiG9w0BAQQFADBMSAwHgYDVQQKEXdOZXRz
Y2FwZSBDb21tdW5pY2F0aW9ucznz jhnMVQ2VydG1maWNhdGUgQXV0aG9yaXR5MB4
XDTk4MDgyNzE5MDAwMFoXDTk5MDIyMzE5MDAwMnBjdGngYoxIDAeBgNVBAoTF05l
dHNjYXB1IENvbW11bml jYXRpb25zMQ8wDQYDVQQLEWZQZW9wbGUxZzAVBgoJkiaJ
kIsZAEBEwdzdXByaXlhMRcwFQYDVQQDEw5TdXByaXlhIFNoZXR0eTEjMCEGCSqGS
Ib3DbndgJARYUc3Vwcm15YUBuZXRzY2FwZS5jb20wXDANBgkqhkiG9w0BAQEFAAN
LADBIAkEAoYiYgthgtbbnjfng jn jgnagwJ jAOBgNVHQ8BAf8EBAMCBLAwFAYJYIZ
IAYb4QgEBAQHBAQDAgCAMA0GCSq
-----END CERTIFICATE-----
```

8. Save the text file and use it to store a copy of the certificate in a subsystem's internal database (see "Step 3. Store the Agent's SSL Client Certificate in the Internal Database" on page 420).

Revocation Status Checking of Agent Certificates

You can configure a Certificate Manager and Registration Manager to check the revocation status of an agent's certificate the server receives during SSL client authentication. You can configure a Data Recovery Manager (or Online Certificate Status Manager) to check the revocation status of its agents' certificates only if you have deployed an OCSP responder and have issued agent certificates with Authority Information Access extension pointing to the OCSP responder. For information about adding Authority Information Access extension to certificates, see "Configuring Policy Rules for a Subsystem" on page 613. For information about setting up an OCSP responder, see Chapter 21, "Setting Up an OCSP Responder."

NOTE The CMS configuration file (`CMS.cfg`) includes a parameter named `jss.ocspcheck.enable`, which enables you to specify whether a CMS manager should use Online Certificate Status Protocol (OCSP) to verify the revocation status of the certificate it receives as a part of SSL client or server authentication (from clients or servers it makes connections with). If you change the value of this parameter to `true`, the CMS manager reads the Authority Information Access extension in the certificate and verifies the revocation status of the certificate from the OCSP responder specified in the extension.

The configuration files of both Certificate Manager and Registration Manager include parameters that enable you to specify whether the server should do the revocation checking and if it should, at what interval. Note that the revocation-status verification works for only those agent certificates that have been issued by the Certificate Manager (and not by any third-party CAs).

The configuration parameters pertaining to this feature of the Certificate Manager are as follows:

```
auths.revocationChecking.bufferSize=5
auths.revocationChecking.ca=ca
auths.revocationChecking.enabled=true
auths.revocationChecking.unknownStateInterval=0
auths.revocationChecking.validityInterval=120
```

The configuration parameters pertaining to this feature of the Registration Manager are as follows:

```
auths.revocationChecking.bufferSize=5
auths.revocationChecking.enabled=true
auths.revocationChecking.ra=ra
auths.revocationChecking.unknownStateInterval=0
auths.revocationChecking.validityInterval=120
```

If you have a Data Recovery Manager installed in the same instance, in addition to the above lines, you'll also notice this line:

```
auths.revocationChecking.kra=kra
```

Table 13-1 provides details for the above mentioned parameters.

Table 13-1 Configuration parameters for checking the revocation status of agents' certificates

Parameter name	Description
<code>revocationChecking.bufferSize</code>	Specifies the total number of last-checked certificates the server should maintain in its cache. For example, if you configure the buffer size to be 2, the server retains the last two certificates it checked in its cache. By default, the server caches the last 5 certificates.
<code>revocationChecking.<subsystem></code>	Specifies the name of the CMS instance. <code><subsystem></code> indicates whether the subsystem is a Certificate Manager (<code>ca</code>) or Registration Manager (<code>ra</code>). You must not change the default values.
<code>revocationChecking.enabled</code>	Specifies whether revocation checking is to be enabled or disabled. To enable the feature, enter <code>true</code> ; to disable the feature, enter <code>false</code> . By default, the feature is enabled.
<code>revocationChecking.unknownStateInterval</code>	The default interval is 0 seconds.
<code>revocationChecking.validityInterval</code>	Specifies how long, in seconds, the cached certificates are considered valid. Be judicious when choosing the interval, especially when configuring a Registration Manager. For example, if you configure the validity period to be 60 seconds, the server discards the certificates in its cache every minute and attempts to retrieve them from their source—the Certificate Manager uses its internal database to retrieve and verify the revocation status of the certificates, whereas the Registration Manager retrieves certificates from its own internal database and then requests the Certificate Manager for the revocation status of these certificates. The default validity period is 120 seconds (2 minutes).

To configure a Certificate Manager or Registration Manager to verify the revocation status of its agents' certificates:

1. Stop the CMS instance; see “Stopping Certificate Management System” on page 330.
2. Go to this directory: `<server_root>/cert-<instance_id>/config`
3. Open the configuration file (`CMS.cfg`) in a text editor.
4. Locate the parameters mentioned above and edit their values as appropriate.

5. Save your changes, and close the configuration file.
6. Start the CMS instance; see “Starting Certificate Management System” on page 322.

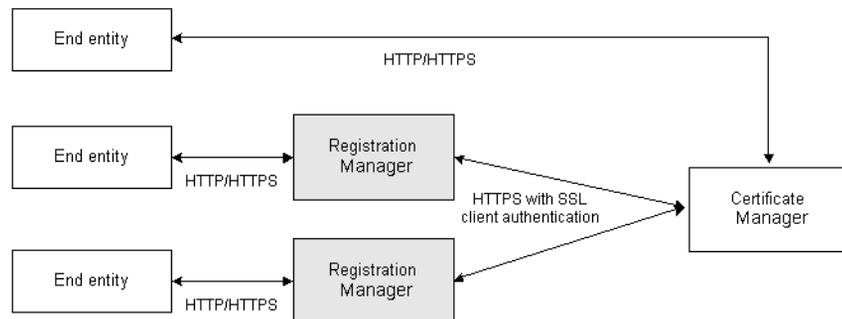
Trusted Managers

Trusted managers are those CMS subsystems or managers that are connected to other CMS subsystems and that are trusted to perform specific functions for them. In other words, a trusted manager acts as a front end to the subsystem that trusts it, performing specific functions, depending on the subsystem to which it is connected. You establish this trust between the two subsystems by configuring them to function in certain way.

Subsystems That Can Function as Trusted Managers

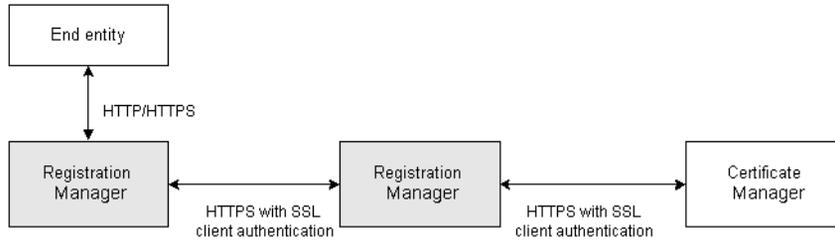
In Certificate Management System, the Registration Manager and Certificate Manager can function as a trusted manager; the Data Recovery Manager and Online Certificate Status Manager cannot function as a trusted manager.

You can configure a Certificate Manager to delegate its end-entity interactions to a trusted Registration Manager, for reasons of localizability (proximity to end entities), customizability, and CA scalability; the Certificate Manager trusts the Registration Manager and signs all certificate signing requests sent by this Registration Manager. For example, as illustrated in the figure below, you might deploy one or more Registration Managers to process, approve, and forward certificate signing requests to a Certificate Manager.

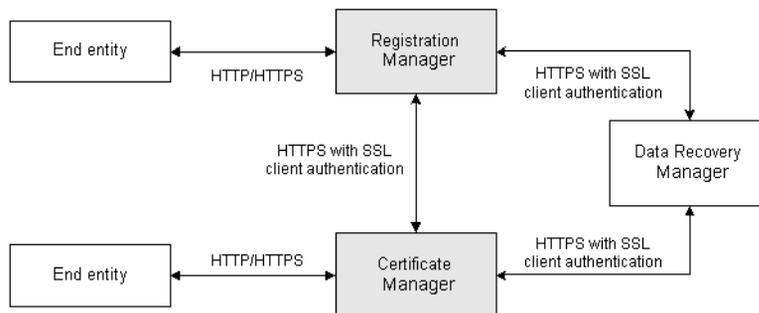


You can configure a Registration Manager to delegate its end-entity interactions to a trusted Registration Manager, for reasons of localizability (proximity to end entities), customizability, and scalability; the Registration Manager trusts the Registration Manager and services all certificate requests sent by this Registration

Manager. For example, as illustrated in the figure below, you might deploy one or more Registration Managers to forward requests to another Registration Manager in a Registration Manager chain. (The Registration Manager passing the request acts as the client and the one receiving the request acts as the server.)



You can configure a Data Recovery Manager to delegate its end-entity interactions to a trusted Certificate Manager or Registration Manager for security reasons; the Data Recovery Manager trusts the Certificate Manager or Registration Manager and services all key archival and recovery requests initiated by this subsystem. For example, as illustrated in figure below, you might deploy one or more Certificate Managers or Registration Managers to send key archival or recovery requests to a Data Recovery Manager.



Connectors for Linking Trusted Managers

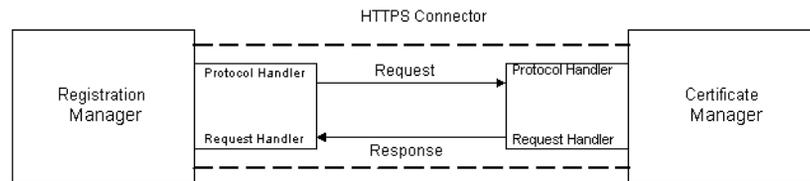
Certificate Management System supports proprietary HTTPS connectors for linking CMS subsystems. You can use these connectors to make the following connections:

- Registration Manager to Certificate Manager
- Registration Manager to Registration Manager
- Registration Manager to Data Recovery Manager

- Certificate Manager to Data Recovery Manager
- Certificate Manager to Certificate Manager (in a cloned-CA setup, which is explained in “Cloning a Certificate Manager” on page 288)

Figure 13-2 illustrates how a trusted Registration Manager communicates with a Certificate Manager.

Figure 13-2 Connectivity service between a trusted Registration Manager and other subsystems



Keep in mind that a trusted manager does not take on the main functions of the subsystem that trusts it. For example, if a Registration Manager is connected to a Certificate Manager, the Registration Manager has no authority to issue (sign) certificates or CRLs. It receives end-entity requests, authenticates them, and forwards them to the Certificate Manager for signing. After receiving a response from the Certificate Manager, it notifies the end entity of the results.

Similarly, a Certificate Manager or Registration Manager connected to a Data Recovery Manager has no authority to archive and recover end users' encryption private keys.

You can configure a subsystem to trust one or more managers. You do this by adding these managers as privileged users to the internal database of that subsystem, assigning them memberships in the appropriate group, and identifying the certificates the managers must use for SSL client authentication to the subsystem they report to. For information about adding a trusted manager, see “Setting Up Trusted Managers” on page 423.

During installation, Certificate Management System automatically creates a group with trusted manager privileges. For more information about this group, see “Group for Trusted Managers” on page 412.

Trusted Manager's Certificate for SSL Client Authentication

By default, a Registration Manager that has been set up to function as a trusted manager uses its *signing certificate* for SSL client authentication to the subsystem that trusts it. For information on this certificate, see “Signing Key Pair and Certificate” on page 459. Similarly, a Certificate Manager that has been set up to function as a trusted manager uses its *SSL server certificate* for SSL client authentication to the subsystem that trusts it. For information on this certificate, see “SSL Server Key Pair and Certificate” on page 455.

When you set up a trusted manager for a CMS subsystem, it is important to know which CA has issued the certificate the trusted manager will use for SSL client authentication to the subsystem. The certificate must be issued by a CA that the subsystem trusts. For example, when you set up a trusted Registration Manager for a subsystem, it is important to know which CA has issued the Registration Manager's signing certificate. The certificate must be issued by a CA that the subsystem trusts. If the subsystem is a Certificate Manager, the certificate must be issued by either the Certificate Manager itself or a CA that the Certificate Manager trusts. Similarly, if the Registration Manager is connected to a Data Recovery Manager, the signing certificate must be issued by the CA that the Data Recovery Manager trusts.

The issuer of a Registration Manager's signing certificate is the CA from which you requested the certificate when you installed the Registration Manager. If you have renewed the certificate since installation, the issuer is the CA from which you requested the renewed certificate. Check the signing certificate for its issuer's name; see “Viewing the Certificate Database Content” on page 523. You can also find this information by looking at the installation worksheet you completed in preparation for installing the system.

Once you learn the issuer's name, verify that this CA's certificate exists in the subsystem's trust database and that the certificate is trusted. To check whether the CA's certificate exists in the subsystem's trust database, follow the instructions in “Viewing the Certificate Database Content” on page 523.

- If the CA's certificate isn't listed, follow the instructions in “Using the Wizard to Install a Certificate or Certificate Chain” on page 493 and add the certificate to the subsystem's certificate database.
- If the CA's certificate is listed but *untrusted*, follow the instructions in “Changing the Trust Settings of a CA Certificate” on page 526 and change the trust setting to *trusted*.

Groups and Their Privileges

In Certificate Management System, a *group* refers to a collection of privileged users—administrators, agents, or trusted Registration Managers. Each group has predetermined privileges, based on its access control. All users belonging to a group automatically inherit the privileges of that group.

When you installed Certificate Management System, it automatically created the following groups for the subsystems you installed:

- Group for Administrators
- Groups for Agents
- Group for Trusted Managers

These default groups are created in the internal database of the appropriate CMS instance. They can help you set up your privileged users quickly and easily.

You can add new privileged users to these groups; see “Setting Up Privileged Users” on page 413. You cannot delete or change the group names. Also, don't change the internal database in which the groups are stored.

Group for Administrators

During installation, Certificate Management System automatically creates a group called `Administrators` and adds a user to this group; the server sets the name of this user to the *certificate administrator ID* (of the CMS administrator) you specified during installation. If you don't remember this name, see the installation worksheet you completed in preparation for installing the system (see “Administrator” on page 197). For example, if you specified `admin` as the user ID for the CMS administrator, the name of the user in the `Administrators` group will be `admin`.

Keep in mind that the `Administrators` group must always contain at least one user entry. This means you can delete the entry that was created in this group during installation, provided you add another user to the group.

After installation, be sure to do the following:

1. Log in to the CMS window with the administrator ID and password specified during installation.

2. Depending on the components you installed, create one or more privileged users and add them to the appropriate groups. It is recommended that you add at least one more user to the `Administrators` group. For instructions on creating privileged users and adding them to one or more groups, see “Setting Up Privileged Users” on page 413.

Groups for Agents

Depending on the subsystems you chose to install, Certificate Management System automatically creates a combination of the following groups for a CMS instance:

- `Certificate Manager Agents` group, if you have installed the Certificate Manager
- `Registration Manager Agents` group, if you have installed the Registration Manager
- `Data Recovery Manager Agents` group, if you have installed the Data Recovery Manager
- `Online Certificate Status Manager Agents` group, if you have installed the Online Certificate Status Manager

Group for Certificate Manager Agents

When the Certificate Manager is installed, a group called `Certificate Manager Agents` is automatically created in its internal database. After installation, this group has a single user entry—when you get the first agent certificate from the Certificate Manager (see “Stage 3. Enrolling for Administrator/Agent Certificate” on page 277), the server automatically adds the initial administrator as the agent and stores a copy of the agent certificate against that user entry. The user ID for this agent user is the same as the *certificate administrator ID*, as specified during installation.

The `Certificate Manager Agents` group has access rights to agent-specific resources of the Certificate Manager; that is, privileged users you add to this group automatically inherit access rights to the agent port of the Certificate Manager. For information on ports, see “CMS Ports” on page 381.

After installation, you should add to this group the privileged users to whom you want to assign Certificate Manager agent privileges. All agents who belong to the `Certificate Manager Agents` group can access the Certificate Manager Agent Services interface; see “Certificate Manager Agent Services” on page 69.

For an agent to be able to carry on SSL client-authenticated communication with a Certificate Manager, you need to do additional configurations. See “Setting Up Agents” on page 416.

Group for Registration Manager Agents

When the Registration Manager is installed, a group called `Registration Manager Agents` is automatically created in its internal database. By default, this group has no entries.

The `Registration Manager Agents` group has access rights to agent-specific resources of the Registration Manager; that is, privileged users you add to this group automatically inherit access rights to the agent ports of the Registration Manager. For information on ports, see “CMS Ports” on page 381.

After installation, you should add to this group the privileged users to whom you want to assign Registration Manager agent privileges. All agents who belong to the `Registration Manager Agents` group can access the Registration Manager Agent Services interface; see “Registration Manager Agent Services” on page 70.

For an agent to be able to do SSL client-authenticated communication with a Registration Manager, you need to do additional configurations. See “Setting Up Agents” on page 416.

Group for Data Recovery Manager Agents

When the Data Recovery Manager is installed, a group called `Data Recovery Manager Agents` is automatically created in its internal database. By default, this group has no entries. Note that if the Data Recovery Manager is colocated with a Certificate Manager, following installation, this group has a single user entry—when you get the very first agent certificate from the Certificate Manager, the server automatically adds the initial administrator as the agent and stores a copy of the agent certificate against that user entry. The user ID for this agent user is the same as the *certificate administrator ID* (as specified during installation).

The `Data Recovery Manager Agents` group has access rights to agent-specific resources of the Data Recovery Manager; that is, privileged users you add to this group automatically inherit access rights to the agent ports of the Data Recovery Manager. For information on ports, see “CMS Ports” on page 381.

After installation, you should add to this group the privileged users to whom you want to assign Data Recovery Manager agent privileges. All agents who belong to the `Data Recovery Manager Agents` group can access the Data Recovery Manager Agent Services interface; see “Data Recovery Manager Agent Services” on page 71.

For an agent to be able to carry on SSL client-authenticated communication with a Data Recovery Manager, you need to do additional configurations. See “Setting Up Agents” on page 416.

Group for Online Certificate Status Manager Agents

When the Online Certificate Status Manager is installed, a group called `Online Certificate Status Manager Agents` is automatically created in its internal database. By default, this group has no entries.

The `Online Certificate Status Manager Agents` group has access rights to agent-specific resources of the Online Certificate Status Manager; that is, privileged users you add to this group automatically inherit access rights to the agent ports of the Online Certificate Status Manager. For information on ports, see “CMS Ports” on page 381.

After installation, you should add to this group the privileged users to whom you want to assign Online Certificate Status Manager agent privileges. All agents who belong to the `Online Certificate Status Manager Agents` group can access the Online Certificate Status Manager Agent Services interface; see “Online Certificate Status Manager Agent Services Interface” on page 71.

For an agent to be able to do SSL client-authenticated communication with a Online Certificate Status Manager, you need to do additional configurations. See “Setting Up Agents” on page 416.

Group for Trusted Managers

When the Certificate Manager, Registration Manager, or Data Recovery Manager is installed, a group called `Trusted Managers` is automatically created in its internal database. By default, this group has no entries.

The `Trusted Managers` group has access rights to the corresponding agent gateway; that is, the subsystems you add to this group automatically inherit access rights to the agent port of the corresponding Certificate Manager, Registration Manager, or Data Recovery Manager. For information on ports, see “CMS Ports” on page 381.

After installation, you should add to this group the subsystems that you want to function as trusted managers. All subsystems that belong to the `Trusted Managers` group can carry on SSL client-authenticated communication with the subsystem that trusts them and receive responses back.

For a Registration Manager to be able to do SSL client-authenticated communication with a subsystem, you need to do additional configurations. See “Setting Up Trusted Managers” on page 423.

Setting Up Privileged Users

Setting up privileged users for a CMS instance involves adding the appropriate user information to the internal database of that instance. You can set up any number of privileged users for a CMS instance. If the user is a person (that is, an administrator or agent), you can put that user into as many groups as you like.

This section describes the following tasks:

- Setting Up Administrators
- Setting Up Agents
- Setting Up Trusted Managers

Setting Up Administrators

You need at least one administrator for each instance of Certificate Management System. To understand the role of an administrator, see “Administrators” on page 396.

To set up an administrator follow these steps:

- Step 1. Find the Required Information
- Step 2. Add the Information to the Internal Database

Step 1. Find the Required Information

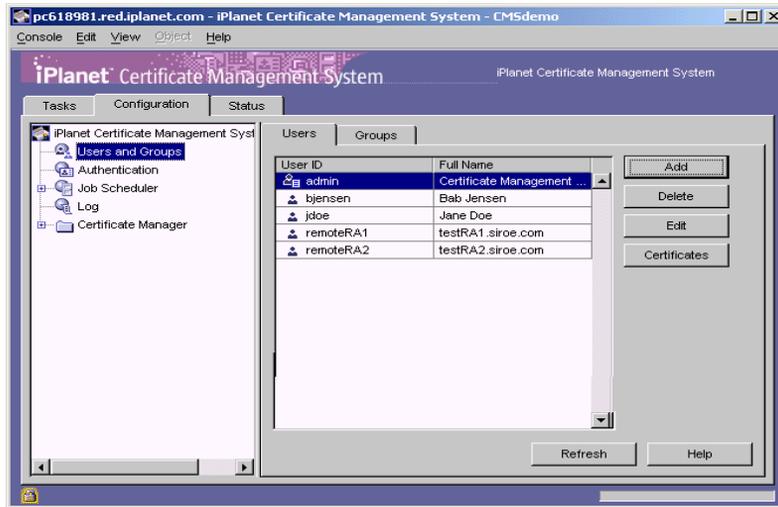
Note the user’s corporate information, such as name, user ID, email address, and phone number.

Step 2. Add the Information to the Internal Database

To add the information to the internal database of a CMS instance:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).

2. In the navigation tree, select Users and Groups.
The Users tab appears on the right pane.



3. Click Add.
The Select User Type window appears.



4. Select Administrator and click OK.

The Edit User Information window appears.

5. Specify information as appropriate:

User ID. Type a user ID or login name for the user. The ID can be an alphanumeric string of up to 255 characters. Give this ID to the user. The user is required to enter this ID in the login screen of the CMS window; see “Logging In to the CMS Window” on page 351.

Full name. Type the user’s full name. The user never sees this. This field is to help you keep track of your users. The name can be an alphanumeric string of up to 255 characters.

Password. Type a password of up to eight characters for the user. Give this password to the user. The user is required to enter this password in the login screen of the CMS window.

Confirm password. Retype the password exactly as you typed it in the Password field.

Email. Type the user’s complete email address. The user never sees this. This field is to help you contact the user, if the need arises.

Phone. Type the user’s phone number. The user never sees this. This field is to help you contact the user, if the need arises.

Group. Select Administrators; see “Group for Administrators” on page 409. When you set up a user, you can add him or her to only one group. To add the user to another group, see “Changing Members in a Group” on page 446.

6. Click OK.

You are returned to the Users tab. The administrator you just added will be displayed in the list of users.

7. Click Refresh to view the updated configuration.

Setting Up Agents

You need an agent for each subsystem installed in a given CMS instance. To understand the role of an agent, see “Agents” on page 397. This section explains how to add agents to a CMS instance.

You can set up agents for a CMS instance in two ways:

- Setting up Agents Using the Automated Process
- Setting up Agents Using the Manual Process

Setting up Agents Using the Automated Process

Certificate Management System automates the process of setting up agents if agents request their certificate using the *manual* enrollment form. The automated process is built into the request-approval form (the page that displays the pending request) in the Agent Services interface and it enables the person who has both *Certificate Manager agent* and *Administrator* privileges to create new agents for a CMS instance—that is, the Certificate Manager agent who approves new agents’ certificate requests must belong to both `Certificate Manager Agents` and `Administrators` groups in the internal database of the Certificate Manager.

The request-approval form includes a checkbox labeled “This certificate is for a <subsystem> agent”, where <subsystem> indicates Certificate Manager, Registration Manager, or Data Recovery Manager. Selecting the checkbox indicates that the user who has requested the certificate should be made an agent for the specified subsystem. Selecting the checkbox also requires the Certificate Manager agent to specify a user ID for the new agent.

If the Certificate Manager agent approves the certificate request with the checkbox selected and user ID specified, the server automatically adds the user as an agent to its internal database, copies the user’s client certificate to the database, and associates the certificate with the new user’s entry.

If you want to test this feature, follow these steps:

1. Open a web browser window.
2. Access the end-entity interface.
3. In the Enrollment tab, under Browser, select Manual.
4. In the enrollment form that appears, enter sample data and submit the request.
5. Next, access the Certificate Manager Agent Services interface.
6. Click List Requests.

7. In the page that displays, select the “Show pending requests”, and click Find.
8. In the list of certificate signing requests that displays, select the request you submitted.
9. In the request approval form for user enrollment requests, verify the request. If required, adjust some of the parameters such as the subject name and validity period.
10. Next, check the box labeled “This certificate is for a Certificate Manager agent”, specify a user ID for the new agent, and approve the certificate request.

The Certificate Manager processes the requests, issues the certificate to the user, automatically adds the user as an agent to its user and group database, copies the user’s client certificate to the database, and associates the certificate with the new user’s entry.

11. To verify, log in to the CMS window for the Certificate Manager.
12. In the navigation tree, click Users and Groups.
13. In the list of users, you should find the user ID you specified for the new agent. To view the certificate issued to the new agent, select the user ID and click Certificates.

Setting up Agents Using the Manual Process

Typically, you add agents to a CMS instance by manually creating a privileged-user entry for the agent in the corresponding subsystem’s user and group database and then add the agent’s certificate to the database.

Setting up an agent involves the following steps:

- Step 1. Find the Required Information
- Step 2. Add the Information to the Internal Database
- Step 3. Store the Agent’s SSL Client Certificate in the Internal Database
- Step 4. Check the Certificate Database for the CA Certificate

Step 1. Find the Required Information

Before adding an agent to the internal database of a CMS instance:

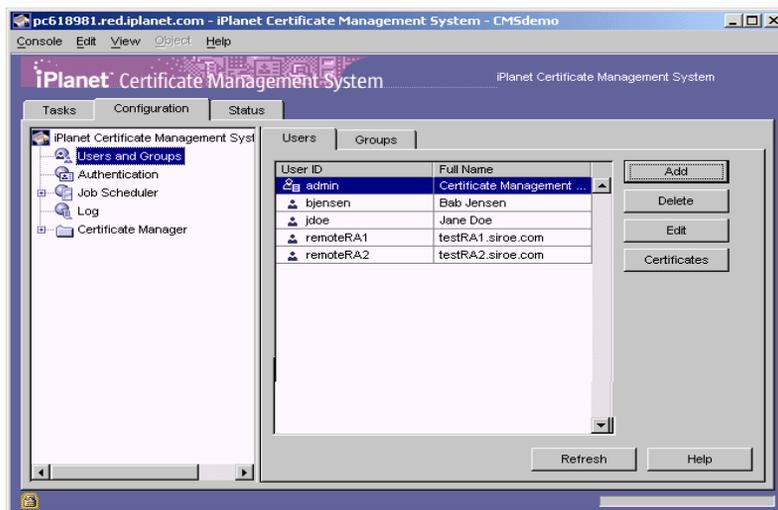
- Note the user's corporate information, such as name, login ID, password, email address, and phone number.
- Make sure the user has one or more client certificates that are currently valid; the certificate must not have expired, been revoked, or been signed by an *untrusted* authority. If the user does not own a client certificate, either issue the user a certificate or ask the user to get a certificate. For details, see "Agent's Certificate for SSL Client Authentication" on page 399.
- Identify the certificate that the user must use for SSL client authentication to Certificate Management System. You can identify more than one certificate if you want.
- Copy this certificate, in base-64 encoded format, to a text file.

Step 2. Add the Information to the Internal Database

To add the information to the internal database of a CMS instance:

1. Log in to the CMS window (see "Logging In to the CMS Window" on page 351).
2. In the navigation tree, select Users and Groups.

The Users tab appears.



3. Click Add.

The Select User Type window appears.



4. Select Agent and click OK.

The Edit User Information window appears.



5. Specify information as appropriate.

The information you enter here is to help you keep track of your agent users; the user never sees or uses it. The server relies solely on the agent's client certificate (which you will add next) for authentication.

User ID. Type the user ID or login name. The ID can be an alphanumeric string of up to 255 characters.

Full name. Type the user's full name. The name can be an alphanumeric string of up to 255 characters.

Email. Type the user's complete email address.

Phone. Type the user's phone number.

Group. Choose the appropriate agent group; for more information about this group, see “Groups for Agents” on page 410. When you set up a user, you can add her or him to only one group. To add the user to another group, see “Changing Members in a Group” on page 446.

6. Click OK.

You are returned to the Users tab. The agent you just added is displayed in the list of users.

What you do next depends on whether you have the agent’s certificate:

- If you copied the user’s certificate in base-64 encoded form to a text file, proceed to Step 3. (For details on getting the user’s certificate, see “Agent’s Certificate for SSL Client Authentication” on page 399.)
- Otherwise, save your changes.

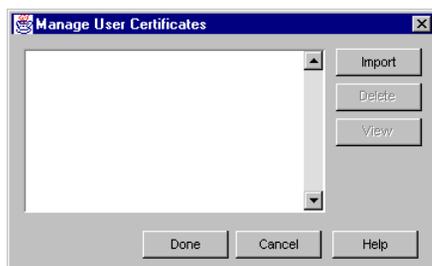
You can add the certificate to the internal database later, following the instructions provided in “Changing a Privileged User’s Certificate” on page 445.

Step 3. Store the Agent’s SSL Client Certificate in the Internal Database

To store a copy of an agent’s SSL client certificate in the internal database:

1. In the Users tab, click Certificates.

The Manage User Certificates window appears.

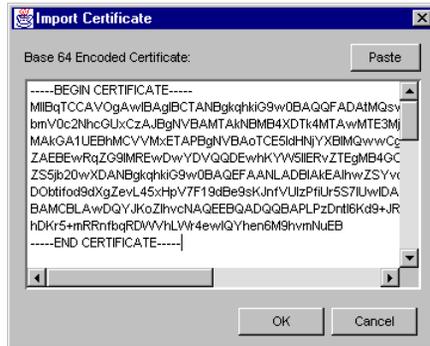


2. Click Import.

The Import Certificate window appears.

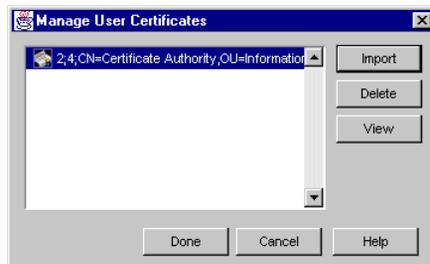
- Click inside the text area, and paste the user's certificate in base-64 encoded form.

Be sure to include the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines.



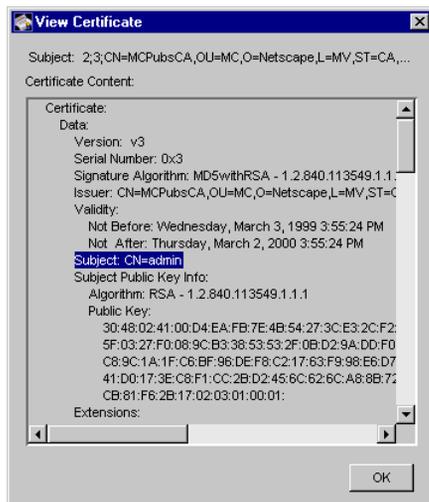
- Click OK.

You are returned to the Manage User Certificates window. The certificate you imported should now be listed in this window.



- To view the certificate you imported, select it and click View.

The certificate information appears.



- Click Done.

You are returned to the Users tab.

- Click Refresh to view the updated configuration.

Step 4. Check the Certificate Database for the CA Certificate

The CA that signed the agent's SSL client certificate must be *trusted* by the subsystem that services requests from the agent. Make sure that this CA's certificate exists in the subsystem's certificate database (internal or external) and that it is trusted. To check whether the CA's certificate exists in your subsystem's certificate database, follow the instructions in "Viewing the Certificate Database Content" on page 523.

- If the CA certificate isn't listed, follow the instructions in "Using the Wizard to Install a Certificate or Certificate Chain" on page 493 and add the certificate to the certificate database.
- If the CA's certificate is listed but *untrusted*, follow the instructions in "Changing the Trust Settings of a CA Certificate" on page 526 and change the trust setting to *trusted*.

Setting Up Trusted Managers

You can set up a Registration Manager or Certificate Manager to function as a trusted manager to another CMS instance. This section explains how to do this.

- Setting up Trusted Managers Using the Automated Process
- Setting Up a Registration Manager as a Trusted Manager
- Setting Up a Certificate Manager as a Trusted Manager

To understand the role of a trusted manager in your PKI, see “Trusted Managers” on page 405.

Setting up Trusted Managers Using the Automated Process

Certificate Management System automates the process of setting up trusted managers. The automated process is built into the request-approval form (the page that displays the pending request) in the Agent Services interface and it enables the person who has both *Certificate Manager agent* and *Administrator* privileges to create new trusted managers for a CMS instance—that is, the Certificate Manager agent who approves the subsystems’ certificate requests must belong to both the Certificate Manager Agents and Administrators groups in the user and group database of the Certificate Manager. For more information about these groups, see “Groups and Their Privileges” on page 409.

- The request-approval form for Certificate Manager’s SSL server certificate request includes a checkbox labeled “This certificate is for a Trusted Manager.”
- Similarly, The request-approval form for Registration Manager’s signing certificate request includes a checkbox labeled “This certificate is for a Trusted Manager.”

If selected, the checkbox indicates that the subsystem that has requested the certificate must be made a trusted manager. Selecting the checkbox also requires the agent to specify an ID for the subsystem that will be set up as a trusted manager.

If the Certificate Manager agent approves the certificate request with the checkbox selected and user ID specified, the server automatically adds the subsystem as a new privileged user to its user and group database, adds the user to the Trusted Managers group, copies the corresponding certificate to the database, and associates the certificate with the new user’s entry.

Note that for a Certificate Manager to add the Registration Manager this way, the Certificate Manager agent who approves the Registration Manager signing certificate request must belong to both the Certificate Manager Agents and Administrators groups in the internal database of the Certificate Manager. For more information about these groups, see “Groups and Their Privileges” on page 409.

Setting Up a Registration Manager as a Trusted Manager

You can set up a remote Registration Manager to function as a trusted manager to a Certificate Manager, another Registration Manager, or a Data Recovery Manager.

- Step 1. Find the Required Information
- Step 2. Create a User Entry for the Registration Manager
- Step 3. Copy the Registration Manager’s Certificate to the Internal Database
- Step 4. Check the Certificate Database for the CA Certificate
- Step 5. Configure Registration Manager’s Connector Settings
- Step 6. Configure the List Certificates Page

Step 1. Find the Required Information

Before setting up a Registration Manager to function as a trusted manager to another CMS subsystem:

- Note identifying information, such as the instance ID and host name of the Registration Manager.
- Make sure that the Registration Manager has the certificate you want it to use for SSL client authentication to the subsystem that will trust it; by default, the Registration Manager uses its *signing certificate* for this purpose. The certificate must be currently valid; the certificate must not have expired, been revoked, or been signed by an authority *untrusted* by the subsystem. For details, see “Trusted Manager’s Certificate for SSL Client Authentication” on page 408.
- Locate the certificate in base-64 encoded format. Copy the certificate, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines, to a text file.
- Identify the subsystem—Certificate Manager, Registration Manager, or Data Recovery Manager—to which you want to connect the Registration Manager. Note details, such as the host name and port number of that subsystem.

- If you are planning to connect the Registration Manager to a Certificate Manager, keep this in mind: during the installation of a Registration Manager, you generated a signing certificate for the Registration Manager. If you requested the signing certificate from a Certificate Manager, you were given an opportunity to add the Registration Manager as a trusted manager to that Certificate Manager's database. If you chose this option, then the Registration Manager is already set up to function as a trusted manager to that Certificate Manager—in this case, you are not required to go through these steps.

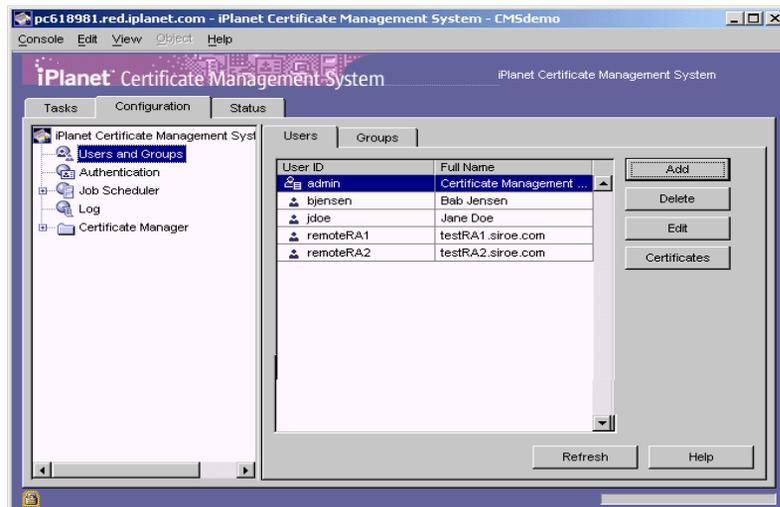
Step 2. Create a User Entry for the Registration Manager

In this step, you create a privileged-user entry for the Registration Manager in the internal database of the subsystem. As a part of creating this entry, you also add the user entry to the `Trusted Managers` group in order to give the entry access privileges to the agent port of the subsystem.

To create a user entry with appropriate access privileges for a Registration Manager:

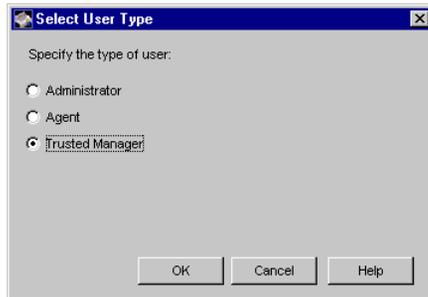
1. Log in to the CMS window for the subsystem (see “Logging In to the CMS Window” on page 351). For the purposes of completing these instructions, let us assume that the subsystem is a Certificate Manager.
2. In the navigation tree, select `Users and Groups`.

The `Users` tab appears.



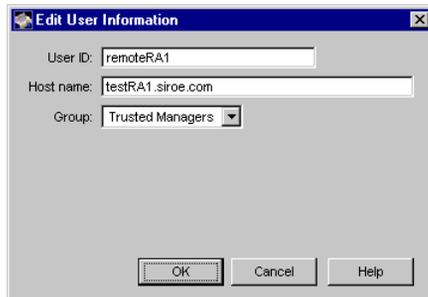
3. Click Add.

The Select User Type window appears.



4. Select Trusted Manager and click OK.

The Edit User Information window appears.



5. Specify information as appropriate.

The information you enter here is to help you keep track of the Registration Manager; the subsystem never uses it. The subsystem relies solely on the Registration Manager's SSL client certificate (which you will add in Step 3) for authentication.

User ID. Type the Registration Manager's instance ID (or any other ID that will help you identify the Registration Manager in the list of privileged users). The ID can be an alphanumeric string of up to 255 characters.

Host name. Type the full host name of the Registration Manager. The host name can be an alphanumeric string of up to 255 characters. It must be in the <machine_name>.<your_domain>.<domain> form.

Group. Select Trusted Managers; for more information about this group, see "Group for Trusted Managers" on page 412.

6. Click OK.

You are returned to the Users tab. The Registration Manager you just added appears in the list of users.

What you do next depends on whether you have the Registration Manager's SSL client certificate:

- If you copied the Registration Manager's certificate in base-64 encoded form to a text file, proceed to Step 3. (For details on getting this certificate, see "Trusted Manager's Certificate for SSL Client Authentication" on page 408.)
- Otherwise, skip to Step 5. You can add the certificate later, following the instructions in "Changing a Privileged User's Certificate" on page 445.

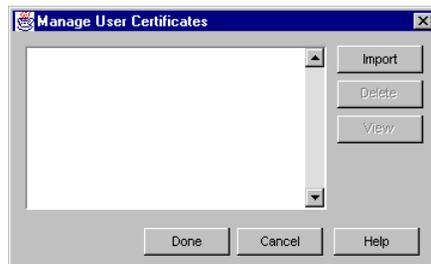
Step 3. Copy the Registration Manager's Certificate to the Internal Database

In this step, you add a copy of the Registration Manager's SSL client authentication certificate to the internal database of the subsystem and associate the certificate with the user entry you created in Step 2.

To store the Registration Manager's SSL client certificate in the internal database of the subsystem:

1. In the Users tab, select the user entry you just added for the Registration Manager and click Certificates.

The Manage User Certificates window appears.

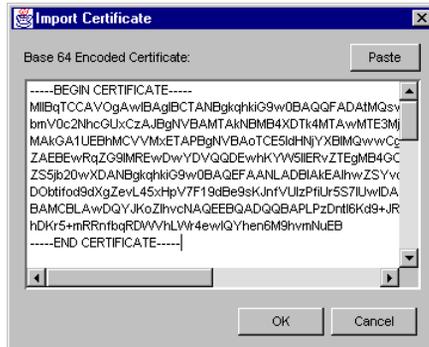


2. Click Import.

The Import Certificate window appears.

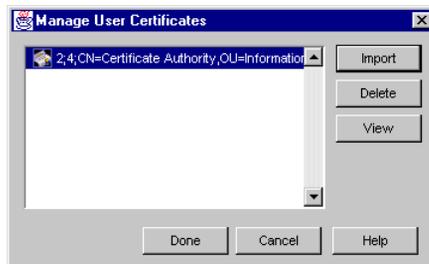
- Click inside the text area, and paste the Registration Manager's certificate in base-64 encoded form.

Be sure to include the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines.



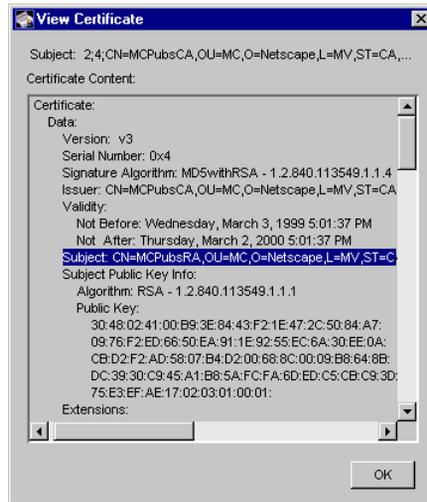
- Click OK.

You are returned to the Manage User Certificates window. The certificate you imported should now be listed in this window.



- To view the certificate you imported, select it and click View.

The certificate information appears. Verify that the certificate you added is the correct one.



- Click Done.

You are returned to the Users tab.

Step 4. Check the Certificate Database for the CA Certificate

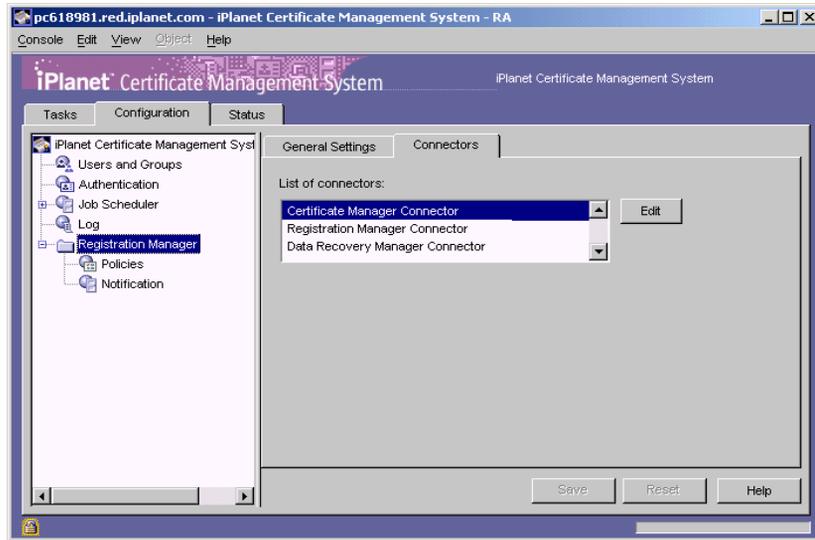
The issuer of the Registration Manager's certificate that you added in Step 3 must be *trusted* by the subsystem that services certificate requests approved by the Registration Manager. Make sure that this CA's certificate exists in the subsystem's certificate database (internal) and that it is trusted. To check whether the CA's certificate exists in the subsystem's certificate database, follow the instructions in "Viewing the Certificate Database Content" on page 523.

- If the CA certificate isn't listed, follow the instructions in "Using the Wizard to Install a Certificate or Certificate Chain" on page 493 and add the certificate to the certificate database.
- If the CA's certificate is listed but *untrusted*, follow the instructions in "Changing the Trust Settings of a CA Certificate" on page 526 and change the trust setting to *trusted*.

Step 5. Configure Registration Manager's Connector Settings

In this step, you configure the connector settings of the Registration Manager. This enables the Registration Manager to utilize the proprietary HTTPS connectors to communicate with the subsystem (following successful SSL client authentication).

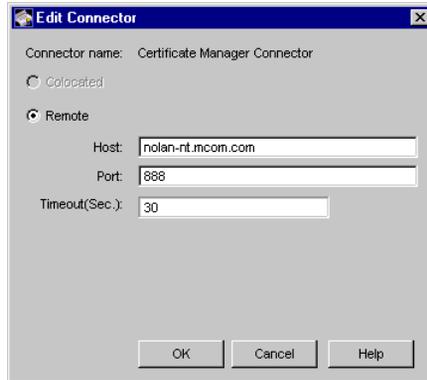
1. Log in to the CMS window for the Registration Manager (see “Logging In to the CMS Window” on page 351).
2. In the navigation tree, select Registration Manager.
The General Settings tab appears in the right pane.
3. Select the Connectors tab.



4. In the “List of connectors” select the connector:
 - If you are connecting the Registration Manager to a Certificate Manager, select Certificate Manager Connector and click Edit.
 - If you are connecting the Registration Manager to another Registration Manager, select Registration Manager Connector and click Edit.
 - If you are connecting the Registration Manager to a Data Recovery Manager, select Data Recovery Manager Connector and click Edit.

For the purposes of completing these instructions, let us assume you selected Certificate Manager Connector.

The Edit Connector dialog box appears.



5. Select the Enable checkbox to enable the connector configuration.
6. Select Remote, and enter the appropriate information:

Host. Type the full host name of the subsystem that trusts this Registration Manager; in this case, it would be the host name of the Certificate Manager. The Registration Manager uses this name to locate the Certificate Manager. The format for the host name must be as follows:

```
<machine_name>.<your_domain>.<domain>
```

Port. Type the number of the TCP/IP port at which the Certificate Manager will listen to requests from the trusted Registration Manager. The default port designated for communication between a trusted Registration Manager and a subsystem is the agent's port. See "Agent Port" on page 383.

Timeout. Connection timeout. By default, it is 30 seconds.

The sample screen above shows how to connect the Registration Manager to a Certificate Manager running on a host called `nolan-nt.mcom.com` listening for HTTPS requests on port 888.

7. Click OK.

You are returned to the Connectors tab.

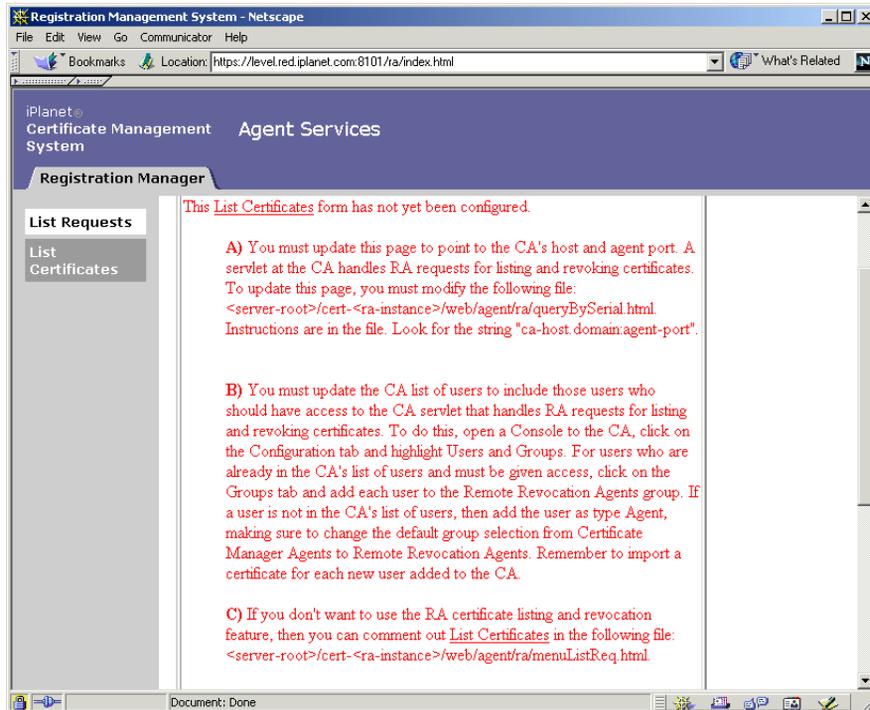
8. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, stop and restart the server.

Step 6. Configure the List Certificates Page

In this step, you configure the List Certificates page of the Registration Manager GUI. This is a required if you want the Registration Manager to be able to view and revoke certificates. By default, the List Certificates GUI is filled with a text message—and rendered unusable—until you configure the page.

NOTE If you do not want the Registration Manager to be able to view and revoke certificates, then skip this step and follow the instructions in the next section, “Disabling The List Certificates Feature.”

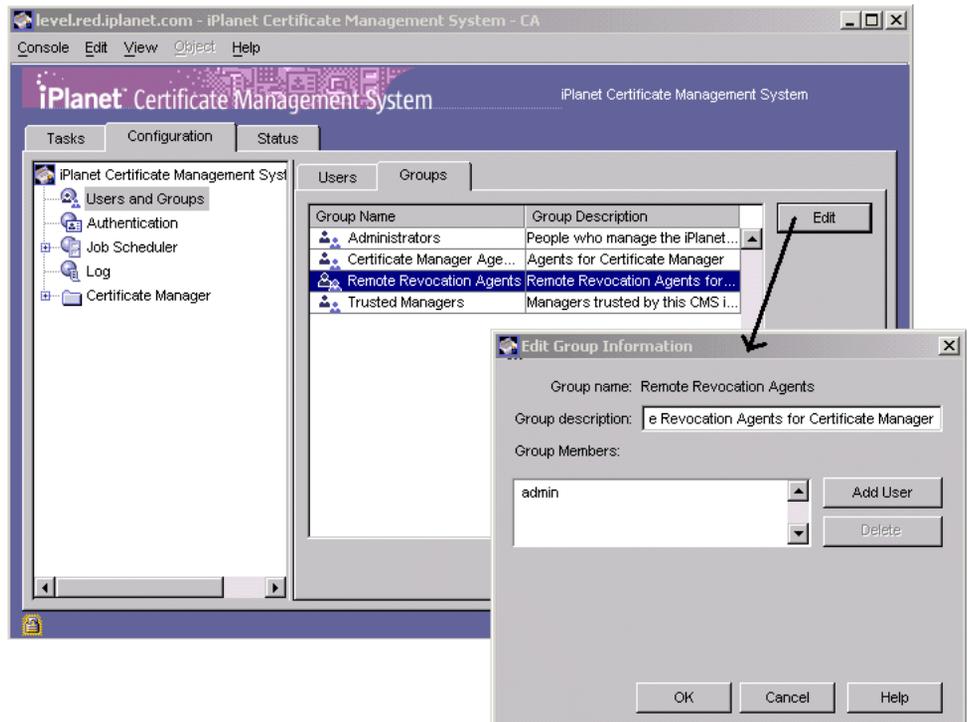


1. You must update this page to point to the CA's host and agent port. Modify the following file:

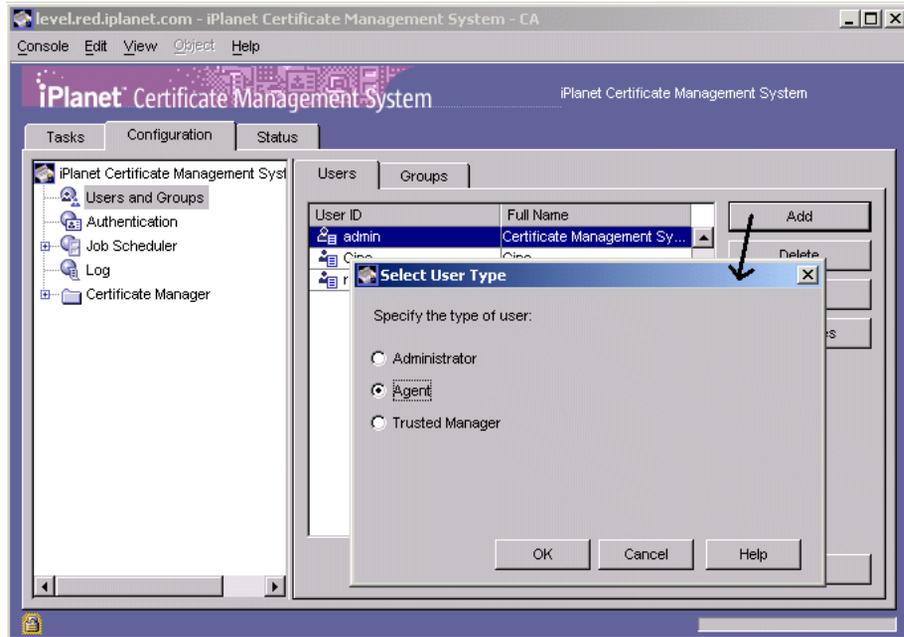
```
<server-root>/cert-<ra-instance>/web/agent/ra/queryBySerial.html.
```

Instructions are in the file. Look for the string `ca-host.domain:agent-port`.

2. Update the CA list of users to include those users who should have access to the CA servlets which handle RA requests for listing and revoking certificates.
 - a. In the Certificate Server window, click Configuration.
 - b. In the navigation tree, highlight Users and Groups.
 - o For users who are already in the CA's list of users and must be given access, click on the Groups tab, and then add each user to the Remote Revocation Agents group.

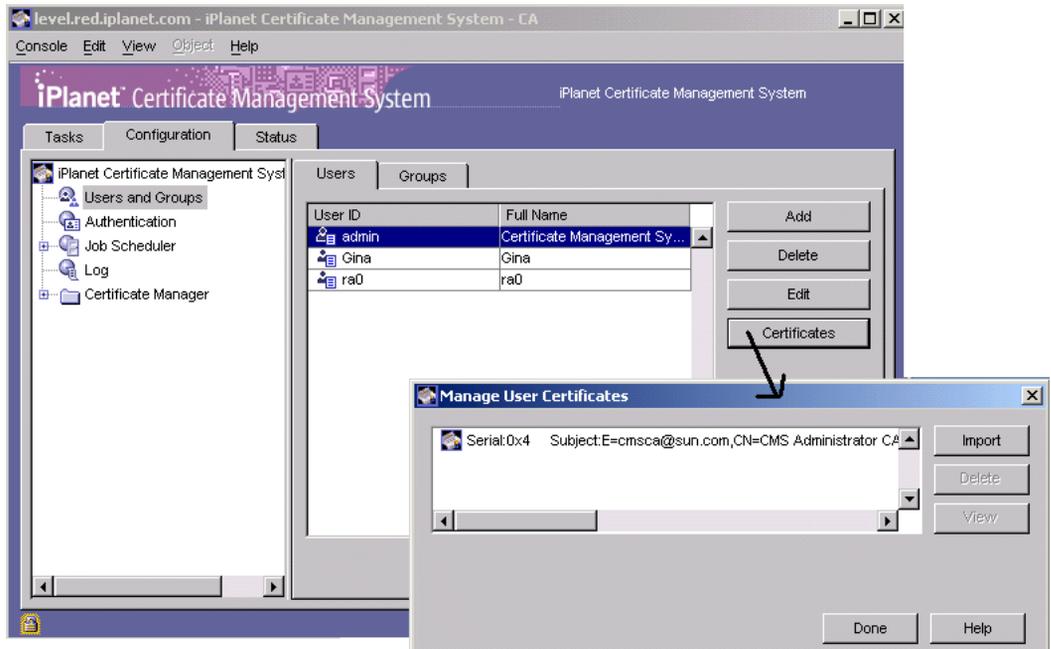


- If a user is not in the CA's list of users, then add the user as type Agent. Be sure to change the default group selection from Certificate Manager Agents to Remote Revocation Agents.



3. Import a certificate for each new user added to the CA.

NOTE It is not necessary to issue a second certificate to the RA agent. The same certificate used by the RA agent to access the RA's agent pages may be used here.



Disabling The List Certificates Feaure

If you don't want to use the RA certificate listing and revocation feature, then you can comment out the string `List Certificates` in the following file:

```
<server-root>/cert-<ra-instance>/web/agent/ra/menuListReq.html
```

Example:

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#CCCCCC" link="#FFFFFF" vlink="#FFFFFF" alink="#333399">
  <table border="0" cellspacing="4" cellpadding="4" width="100%">
    <tr>
      <td bgcolor="white"><font size="-1" face="PrimaSans BT, Verdana,
sans-serif">
        <a href="frameListReq.html" target="middle"><b><font color=black>List
Requests</font></b></a></font></td>
    </tr>
<!--
    <tr>
      <td bgcolor="#999999"><font size="-1" face="PrimaSans BT, Verdana, sans-serif">
<a href="frameList.html" target="middle"><b>List Certificates</b></a></font></td>
    </tr>
-->
  </table>
</body>
</html>
```

Setting Up a Certificate Manager as a Trusted Manager

You can set up a Certificate Manager to function as a trusted manager to a remote Data Recovery Manager. The setup process involves the following steps:

- Step 1. Find the Required Information
- Step 2. Create a User Entry for the Certificate Manager
- Step 3. Copy the Certificate Manager's Certificate to the Internal Database
- Step 4. Check the Certificate Database for the CA Certificate
- Step 5. Configure Certificate Manager's Connector Settings

Step 1. Find the Required Information

Before setting up a Certificate Manager to function as a trusted manager to a Data Recovery Manager:

- Note identifying information, such as the instance ID and host name of the Certificate Manager.
- Make sure that the Certificate Manager has the certificate you want it to use for SSL client authentication to the Data Recovery Manager that will trust it; by default, the Certificate Manager uses its SSL server certificate for this purpose. The certificate must be currently valid; the certificate must not have expired, been revoked, or been signed by an authority *untrusted* by the subsystem. For details, see “Trusted Manager’s Certificate for SSL Client Authentication” on page 408.
- Locate the certificate in base-64 encoded format. Copy the certificate, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines, to a text file.
- Identify the Data Recovery Manager to which you want to connect the Certificate Manager. Note details, such as the host name and port number of that Data Recovery Manager.

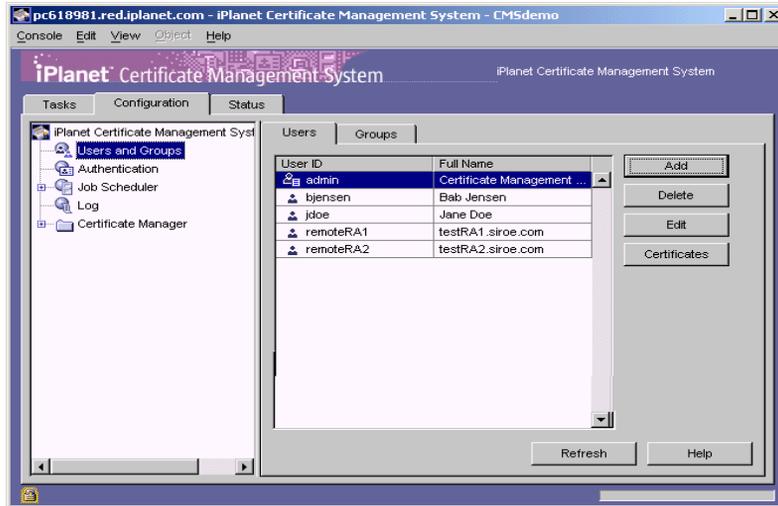
Step 2. Create a User Entry for the Certificate Manager

In this step, you create a privileged-user entry for the Certificate Manager in the internal database of the Data Recovery Manager. As a part of creating this entry, you also add the user entry to the `Trusted Managers` group in order to give the entry access privileges to the agent port of the Data Recovery Manager.

To create a user entry with appropriate access privileges for a Certificate Manager:

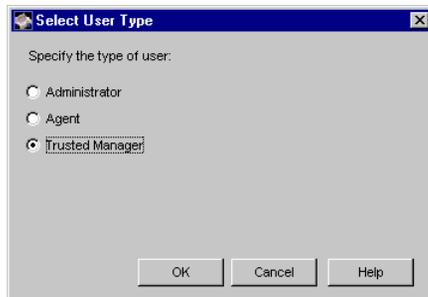
1. Log in to the CMS window for the Data Recovery Manager (see “Logging In to the CMS Window” on page 351).
2. In the navigation tree, select Users and Groups.

The Users tab appears in the right pane.



3. Click Add.

The Select User Type window appears.



4. Select Trusted Manager and click OK.

The Edit User Information window appears.



5. Specify information as appropriate.

The information you enter here is to help you keep track of the Certificate Manager; the Data Recovery Manager never uses it. The Data Recovery Manager relies solely on the Certificate Manager's SSL server certificate (which you will add in Step 3) for authentication.

User ID. Type the Certificate Manager's instance ID (or any other ID that will help you identify the Certificate Manager in the list of privileged users). The ID can be an alphanumeric string of up to 255 characters.

Host name. Type the fully qualified host name of the Certificate Manager. The host name can be an alphanumeric string of up to 255 characters. It must be in this form: <machine_name>.<your_domain>.<domain>

Group. Select Trusted Managers; for more information about this group, see "Group for Trusted Managers" on page 412.

6. Click OK.

You are returned to the Users tab. The Certificate Manager you just added is displayed in the list of users.

What you do next depends on whether you have the Certificate Manager's SSL server certificate:

- If you copied the Certificate Manager's certificate in base-64 encoded form to a text file, proceed to Step 3. (For details on getting this certificate, see "Trusted Manager's Certificate for SSL Client Authentication" on page 408.)
- Otherwise, skip to Step 5. You can add the certificate later, following the instructions in "Changing a Privileged User's Certificate" on page 445.

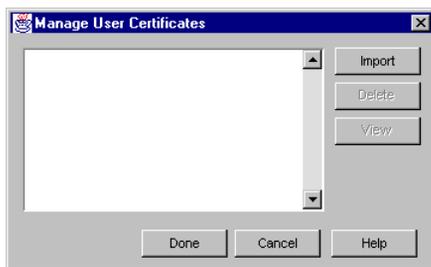
Step 3. Copy the Certificate Manager's Certificate to the Internal Database

In this step, you add the Certificate Manager's SSL server certificate to the internal database of the Data Recovery Manager and associate the certificate with the user entry you created in Step 2.

To store the Certificate Manager's SSL server certificate in the internal database of the subsystem:

1. In the Users tab, select the user entry you just added for the Certificate Manager and click Certificates.

The Manage User Certificates window appears.

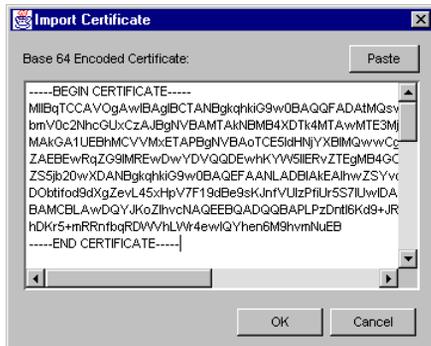


2. Click Import.

The Import Certificate window appears.

3. Click inside the text area, and paste the Certificate Manager's certificate in base-64 encoded form.

Be sure to include the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines.



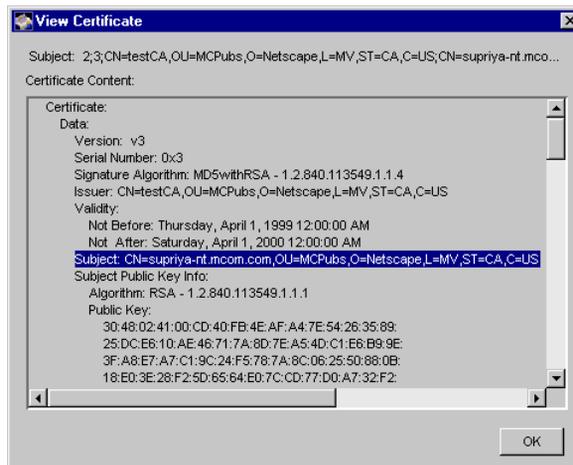
- Click OK.

You are returned to the Manage User Certificates window. The certificate you imported should now be listed in this window.



- To view the certificate you imported, select it and click View.

The certificate information appears. Verify that the certificate you added is the correct one.



- Click Done.

You are returned to the Users tab.

Step 4. Check the Certificate Database for the CA Certificate

The issuer of the Certificate Manager's certificate that you added in Step 3 must be *trusted* by the Data Recovery Manager that services the key archival requests initiated by the Certificate Manager. Make sure that this CA's certificate exists in the Data Recovery Manager's certificate database (internal) and that it is trusted. To check whether the CA's certificate exists in the Data Recovery Manager's certificate database, follow the instructions in "Viewing the Certificate Database Content" on page 523.

- If the CA certificate isn't listed, follow the instructions in "Using the Wizard to Install a Certificate or Certificate Chain" on page 493 and add the certificate to the certificate database.
- If the CA's certificate is listed but *untrusted*, follow the instructions in "Changing the Trust Settings of a CA Certificate" on page 526 and change the trust setting to *trusted*.

Step 5. Configure Certificate Manager's Connector Settings

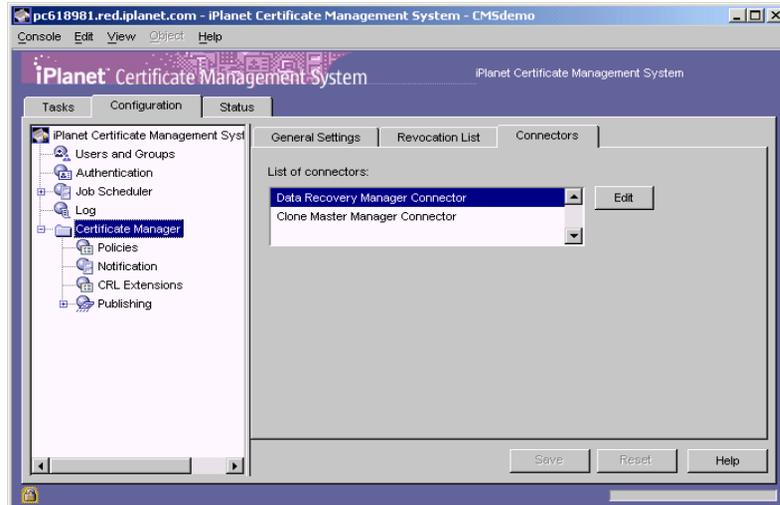
In this step you configure the connector settings of the Certificate Manager. This enables the Certificate Manager to utilize the proprietary HTTPS connectors to communicate with the Data Recovery Manager (following successful SSL client authentication).

Note that during the installation of a Data Recovery Manager, you were prompted to specify the host name and port number of the Certificate Manager to which the Data Recovery Manager will be connected. If you specified this information, you are not required to go through this step. However, it is recommended that you verify the connector setting and make sure that the information you entered during installation is correct.

1. Log in to the CMS window for the Certificate Manager (see "Logging In to the CMS Window" on page 351).
2. In the navigation tree, select Certificate Manager.

The General Settings tab appears in the right pane.

3. Select the Connectors tab.



4. In the “List of connectors” select Data Recovery Manager Connector and click Edit.

The Edit Connector dialog box appears.



5. Select the Enable checkbox to enable the connector configuration.

6. Select Remote, and enter the appropriate information:

Host. Type the full host name of the Data Recovery Manager that trusts this Certificate Manager. The Certificate Manager uses this name to locate the Data Recovery Manager. The format for the host name must be in the `<machine_name>.<your_domain>.<domain>` form.

Port. Type the number of the TCP/IP port at which the Data Recovery Manager will listen to requests from the trusted Certificate Manager. The port designated for communication between a trusted Certificate Manager and a Data Recovery Manager is the agent port. See “Agent Port” on page 383.

Timeout. Connection timeout. By default, it is 30 seconds.

The sample screen above shows how to connect the Certificate Manager to a Data Recovery Manager running on a host called `testKRA.siroe.com` listening for HTTPS requests at port 8000.

7. Click OK.

You are returned to the Connectors tab.

8. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, stop and restart the server.

Changing Privileged-User Information

You can change privileged-user information in several ways:

- To change the login information of a privileged user, see “Changing a Privileged User’s Login Information” on page 444.
- To add or remove certificates of a privileged user, see “Changing a Privileged User’s Certificate” on page 445.
- To change the group membership or access permissions of a privileged user, see “Changing Members in a Group” on page 446.

Changing a Privileged User’s Login Information

To change a privileged user’s login information:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. In the navigation tree, select Users and Groups.

The Users tab appears in the right pane.

- In the User ID list, select the user you want to edit, and click Edit.
The Edit User Information window appears.

- Make the appropriate modifications.
If you need details about individual fields, see “Setting Up Privileged Users” on page 413.
- Click OK.
You are returned to the Users tab.
- Click Refresh to view the updated configuration.

Changing a Privileged User’s Certificate

To change a privileged user’s certificate:

- Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
- In the navigation tree, select Users and Groups.
The Users tab appears in the right pane.

3. In the User ID list, select the user whose certificate information you want to change, and click Certificates.

The Manage User Certificate window appears.



4. Take the appropriate action:
 - o To view a certificate, select the certificate and click View.
 - o To delete a certificate, select the certificate and click Delete.
 - o To add a new certificate for this user to the internal database, click Import. In the Import Certificate window that appears, paste the new certificate in the text area. Be sure to paste the entire base-64 encoded block, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines. For details on getting the user's certificate, see "Agent's Certificate for SSL Client Authentication" on page 399.
5. Click Done.

You are returned to the Users tab.
6. Click Refresh to view the updated configuration.

Changing Members in a Group

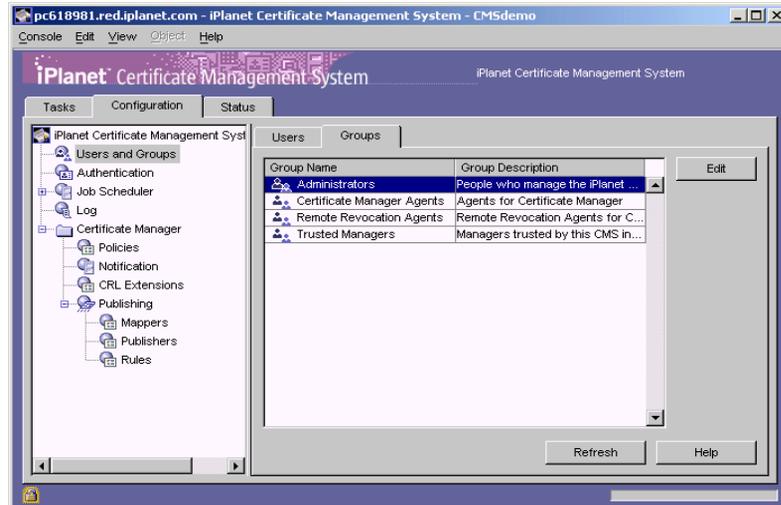
You can add or remove members from all groups. Keep in mind that the group for administrators must have at least one user entry. For details, see "Groups and Their Privileges" on page 409.

To change a group's members:

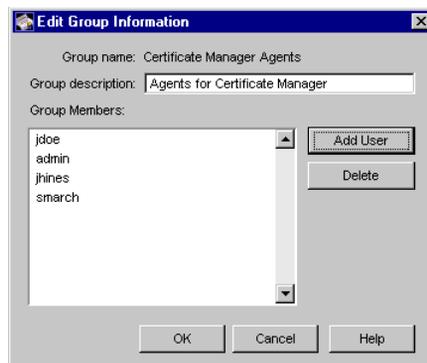
1. Log in to the CMS window (see "Logging In to the CMS Window" on page 351).
2. In the navigation tree, select Users and Groups.

The Users tab appears in the right pane.

3. Click the Groups tab.



4. In the Group Name list, select the group you want to change, and click Edit. The Edit Group Information window appears.



5. Make the appropriate changes:

- To change the group description, type a new description in the “Group description” field.
- To remove a user from the group, select the user and click Delete.
- To add users, click Add User. In the User Selection window that appears, select the users you want to add and click OK. You are returned to the Edit Group Information window.

6. Click OK when you are done with the changes.
You are returned to the Groups tab.
7. Click Refresh to view the updated configuration.

Deleting a Privileged User

You can delete privileged users from the internal database. Deleting a user from the internal database deletes that user from all groups to which the user belongs. If you want to delete a user from specific groups, you should modify the appropriate groups; for details, see “Changing Members in a Group” on page 446.

To delete a privileged user from the internal database:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. In the navigation tree, select Users and Groups.
The Users tab appears in the right pane.
3. In the User ID list, select the user you want to delete, and click Delete.
4. When prompted, confirm your action.
If you click OK, the user entry is deleted from the internal database.
5. Click Refresh to view the updated configuration.

Managing CMS Keys and Certificates

The main subsystems of iPlanet Certificate Management System (CMS)—the Certificate Manager, Registration Manager, Data Recovery Manager and Online Certificate Status Manager—use certificates for various purposes, including authentication during SSL-enabled communication. For example, when a Registration Manager forwards a certificate issuance request to a Certificate Manager for signing, the Certificate Manager expects the Registration Manager to have performed SSL client authentication before processing the request.

When you installed Certificate Management System, the installation program prompted you to generate the required certificates for the subsystems you chose to install. This chapter provides an overview of those certificates and it explains how to perform operations such as renewing the existing certificates before their validity period expires, getting new certificates for the subsystems, adding trusted CA certificates and certificate chains to the CMS trust database, and changing the trust setting of CA certificates. The chapter also explains the certificate Setup Wizard, which automates the process of requesting and installing CMS certificates.

The chapter has the following sections:

- Keys and Certificates for the Main Subsystems (page 450)
- Tokens for Storing CMS Keys and Certificates (page 464)
- Hardware Cryptographic Accelerators (page 477)
- Certificate Setup Wizard (page 478)
- Configuring the Server's Security Preferences (page 500)
- Getting New Certificates for the Subsystems (page 507)
- Renewing Certificates for the Subsystems (page 515)
- Managing the Certificate Database (page 523)

Keys and Certificates for the Main Subsystems

This section explains the various certificates required and used by the CMS managers:

- Certificate Manager's Key Pairs and Certificates
- Registration Manager's Key Pairs and Certificates
- Data Recovery Manager's Key Pairs and Certificates
- Online Certificate Status Manager's Key Pairs and Certificates

The key pairs that correspond to certificates used by these subsystems can be stored either in an internal or an external token, or in both. It depends on the token you chose for the generation and storage of the keys and certificates. For information on tokens, see “Tokens for Storing CMS Keys and Certificates” on page 464.

As an administrator, you must make sure that the private keys that correspond to all certificates, especially the CA signing certificate, used by CMS managers are adequately protected. This includes protecting them from damage (in other words, by archiving and backing up the keys) as well as protecting them from unauthorized access or use. The passwords that protect the tokens containing these keys must also be carefully guarded. Access to the token itself should be limited.

- If the keys are in the internal token (the `key3.db` file), make sure that only you or authorized administrators have access to this file. It's also important to know if the file is stored on backup tapes or is otherwise available for someone to intercept. Because the destruction of a private key in a disk crash can be disastrous if you are depending upon that key for a hierarchy of certificate authorities, backing up your key data is commensurately important. If you do make copies of your keys, however, you must protect your backups with the same level of security that you use for protecting your original keys.
- If the keys are in an external token, such as a smart card, keep it in a locked facility. Also, periodically change the passwords that protect these keys. See “Changing a Token's Password” on page 477.

All CMS certificates have a validity period, as specified when the certificates were generated, beyond which they cannot be used. For a certificate to be valid beyond its expiration date, it must be renewed. For instructions to renew a CMS certificate, see section “Renewing Certificates for the Subsystems” on page 515.

All key pairs associated with CMS certificates must be well protected to ensure that they are never compromised. However, if you know or suspect that a key pair has been compromised, reissue the certificate with a new key pair. For instructions to get a new CMS certificate, see section “Getting New Certificates for the Subsystems” on page 507.

Certificate Manager’s Key Pairs and Certificates

The Certificate Manager uses the following key pairs and corresponding certificates:

- CA Signing Key Pair and Certificate
- wTLS CA Signing Certificate
- OCSP Signing Key Pair and Certificate
- CRL Signing Key Pair and Certificate
- SSL Server Key Pair and Certificate
- Remote Administration Server Certificate

CA Signing Key Pair and Certificate

Every Certificate Manager you installed has a certificate, identified as the *Certificate Manager CA signing certificate*, whose public key corresponds to the private key the Certificate Manager uses to sign the X.509 certificates it issues. The first time you generated this certificate is when you installed the Certificate Manager. The default nickname for the certificate is `caSigningCert cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Certificate Manager is installed, and the default validity period for the certificate is two years.

The subject name of the CA signing certificate reflects the name of your certificate authority (CA) as specified during the installation. All certificates signed or issued by the Certificate Manager include this name to identify the issuer of the certificate.

The Certificate Manager’s status as a root or subordinate CA is determined by whether its CA signing certificate is self-signed or is signed by another CA.

- If the Certificate Manager is a root CA, its CA signing certificate is self-signed—that is, the subject name and issuer name of the certificate is the same.

- If the Certificate Manager is a subordinate CA, its CA signing certificate is signed by another CA, usually the one that is a level above in the CA hierarchy (which may or may not be a root CA). If you have deployed the Certificate Manager as a subordinate CA in a CA hierarchy, you must import your root CA's signing certificate into individual clients and servers before you can use the Certificate Manager to issue certificates to them.

NOTE You cannot change the CA name; doing so would make all previously issued certificates invalid. Similarly, reissuing a Certificate Manager's CA signing certificate with a new key pair invalidates all certificates that have been signed by the old key pair.

wTLS CA Signing Certificate

During the installation of a Certificate Manager, you're given the option to enable issuance of Wireless Transport Layer Security (wTLS)-compliant certificates for use with wireless applications. If you chose to enable this option, the Installation Wizard transparently generates a *wTLS CA signing certificate*.

Note that for the wTLS CA signing certificate, the wizard does not generate a separate key pair. Instead, it uses the same key pair that you generated for the CA signing certificate, which is explained in section "CA Signing Key Pair and Certificate" on page 451. The subject name and validity period of the wTLS CA signing certificate will be the same as the one you specified for the CA signing certificate. The Certificate Manager uses the private key (that corresponds to the public key used to generate the wTLS CA signing certificate) to sign both X.509 and wTLS certificates it issues.

OCSP Signing Key Pair and Certificate

During the installation of a Certificate Manager, you're given the option to enable its OCSP-service feature. This feature enables the Certificate Manager to function as an OCSP responder, enabling OCSP-compliant clients to query the Certificate Manager for the revocation status of certificates issued by the Certificate Manager. For more information about an OCSP responder and setting up a Certificate Manager to function as an OCSP responder, see Chapter 21, "Setting Up an OCSP Responder."

Irrespective of whether you chose to enable the OCSP service feature, the Installation Wizard transparently generates a key pair and a corresponding certificate identified as the *OCSP signing certificate*. The reason for generating this certificate even if you chose to not enable the OCSP service is that you can enable the OCSP service feature in the CMS window after installation. This way, if you decide to enable the feature in a future date, you wouldn't have to go through the process of requesting an OCSP signing certificate.

Note that for generating the OCSP signing key pair, the wizard uses some of the information you provide for the CA signing key pair, which is explained in section “CA Signing Key Pair and Certificate” on page 451. The key type, key size, key algorithm, and validity period of the OCSP signing certificate is the same as the one you specified for the CA signing key pair. The subject name of the OCSP signing certificate is in the form `CN=OCSP cert-<cms_instance_id>`, and it contains extensions, such as `OCSPSigning` and `OCSPNoCheck`, required for signing OCSP responses.

The Certificate Manager uses the private key (that corresponds to the public key used to generate the OCSP signing certificate) to sign the OCSP responses it sends to the OCSP-compliant clients when queried about the revocation status of certificates. The Certificate Manager's signature provides persistent proof to the client that the Certificate Manager has processed the request.

The default nickname for the OCSP signing certificate is `ocspSigningCert cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Certificate Manager is installed.

CRL Signing Key Pair and Certificate

By default, a Certificate Manager you have installed uses the same key pair, the one that corresponds to the *CA signing certificate* explained in “CA Signing Key Pair and Certificate” on page 451, for signing certificates and certificate revocation lists (CRLs). For details about CRLs, see “What's a CRL?” on page 635.

If you want a Certificate Manager to use a separate key pair for signing the CRL it generates, you can do so after installation. The instructions are provided below. Note that a Certificate Manager's CRL signing certificate must be signed or issued by itself; make sure you submit the request to the Certificate Manager itself.

1. Request and install a CRL signing certificate for the Certificate Manager. To do this, you may use either of these options:
 - Use the Certificate Setup Wizard available within the CMS window.

- Use the Key Database (`keyutil`) tool to generate a key pair, the Certificate Database (`certutil`) tool to request a certificate for the key pair and install the certificate in the Certificate Manager's certificate database. For more information about the Key Database and Certificate Database tools, see *CMS Command-Line Tools Guide*.

To request and install a CRL signing certificate for a Certificate Manager using its Certificate Setup Wizard, follow these instructions:

- a. Log in to iPlanet Console; see “Logging In to iPlanet Console” on page 344.
 - b. Locate the CMS instance for the Certificate Manager, make sure it's started, and then log in to the CMS window of the Certificate Manager.
 - c. Select the Configuration tab, and then select the Encryption tab.
 - d. Click the Certificate Setup Wizard button to launch the wizard, which is explained in “Certificate Setup Wizard” on page 478.
 - e. Select the option to request a certificate and then follow the on-screen prompts to generate a certificate request for the CRL signing certificate—in the Certificate Selection window, select `Other` and specify `caCrlSigning` as the certificate type in the associated text field.
 - f. Once you have the certificate request ready, submit it to the Certificate Manager so that it can issue a certificate—in the request submission screen of the wizard, use the auto-submission feature by entering the Certificate Manager's hostname and port number so that the request gets added to the Certificate Manager's agent queue. For general instructions to use the wizard to request a certificate, see section “Using the Wizard to Request a Certificate” on page 479.
 - g. Log in to the Agent Services interface, check the request for required extensions. For example, the CRL signing certificate must contain the Key Usage extension with the `crlSigning` bit set. (By default, the Certificate Manager's policy is configured to add the Key Usage extension with correct bits to the CRL signing certificate; see the policy rule named `CRLSignCertKeyUsageExt`, which is an instance of `KeyUsageExt` plug-in.)
 - h. Approve the request.
 - i. Once you have the CRL signing certificate ready, restart the wizard and install the certificate in the Certificate Manager's database. For general instructions to use the wizard to add a certificate, see “Using the Wizard to Install a Certificate or Certificate Chain” on page 493.
2. After you've installed the certificate successfully, go to the Tasks tab and stop the Certificate Manager.

3. Update the Certificate Manager's configuration to recognize the new key pair and certificate.
 - a. In the Certificate Manager host machine, go to this directory:
`<server_root>/cert-<instance_id>/config`
 - b. Open the configuration file (`CMS.cfg`) in a text editor.
 - c. Add the following lines to the configuration file:


```
ca.crl_signing.cacertnickname=<nickname> cert-<instance_id>
ca.crl_signing.defaultSigningAlgorithm=<signing_algorithm>
ca.crl_signing.tokenname=<token_name>
```
 - d. Edit the lines as below. Replace
 - `<nickname>` with the name assigned to the CRL signing certificate.
 - `<instance_id>` with the name assigned to the Certificate Manager instance.
 - `<signing_algorithm>` with `MD5withRSA`, `MD2withRSA`, or `SHA1withRSA`, if the key type is RSA, or `SHA1withDSA`, if the key type is DSA.
 - `<token_name>` with the name of the token used for generating the key pair and the certificate. If you used the internal/software token, use `Internal Key Storage Token` as the value.

For example, your edited entries might look like this:

```
ca.crl_signing.cacertnickname=crlSigningCert cert-demoCA
ca.crl_signing.defaultSigningAlgorithm=MD5withRSA
ca.crl_signing.tokenname=Internal Key Storage Token
```
 - e. Save your changes and close the file.
4. Restart the Certificate Manager. Now the Certificate Manager is ready to use the CRL signing certificate to sign the CRLs it generates.

SSL Server Key Pair and Certificate

Every Certificate Manager you have installed has at least one *SSL server certificate*. The first time you generated this certificate is when you installed the Certificate Manager. The default nickname for the certificate is `Server-Cert cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Certificate Manager is installed.

The Certificate Manager's SSL server certificate was issued by the CA to which you submitted the certificate signing request. You might have submitted the request to the Certificate Manager itself, another internally deployed CA, or a public CA. To find out the issuer name, follow the instructions in "Viewing the Certificate Database Content" on page 523.

The Certificate Manager uses its SSL server certificate to do SSL server-side authentication to the following:

- The End-Entity Services interface (the HTTPS port)
- The Certificate Manager Agent Services interface
- Clone Certificate Managers, when used as a master Certificate Manager in a cloned CA setup (see "Cloning a Certificate Manager" on page 288)

By default, the Certificate Manager uses a single SSL server certificate for authentication purposes. However, you can request and install additional SSL server certificates for the Certificate Manager. For example, you can configure the Certificate Manager to use separate server certificates for authenticating to the End-Entity Services interface and Agent Services interface. For instructions, see "Configuring the Server to Use Separate SSL Server Certificates" on page 500.

If you configure the Certificate Manager for SSL-enabled communication with a publishing directory, the Certificate Manager also uses its SSL server certificate for SSL client authentication to the publishing directory. This is the default configuration. You can configure the Certificate Manager to use an alternate certificate for this purpose; see "Getting an SSL Client Certificate for a Subsystem" on page 502.

If you configure the Certificate Manager to function as a *trusted manager* to a Data Recovery Manager, the Certificate Manager also uses its SSL server certificate for SSL client authentication to the Data Recovery Manager. For details on trusted managers, see "Trusted Managers" on page 405. You can also configure the Certificate Manager to use an alternate certificate for this purpose; see "Getting an SSL Client Certificate for a Subsystem" on page 502.

NOTE If you have installed the Certificate Manager with a Data Recovery Manager, both subsystems use the same SSL server certificate.

Remote Administration Server Certificate

Netscape Console (version 4.2) does not support the DSA key algorithm. To work around this problem, during the installation of a Certificate Manager, the Installation Wizard transparently generates an SSL server certificate identified as the *Remote Administration Server Certificate*. The Certificate Manager uses the certificate for SSL server authentication to the remote administration interface, Netscape Console. The certificate is self-signed, and is generated with RSA key type and a key size of 512 bits. The validity period of the certificate is set to the same validity period that you chose for the SSL server certificate, which is used by the Certificate Manager for SSL server authentication to its HTTP interfaces; see “SSL Server Key Pair and Certificate” on page 455.

Note that the remote administration server certificate is not listed in the internal database, and thus, you’ll not be able to list or search for it in the Retrieval tab of the Certificate Manager’s end-entity interface. However, you’ll see the certificate if you use the command-line tool, Certificate Database Tool (`certutil`) to list certificates in the Certificate Manager’s certificate database (the `cert7.db` file):

- The nickname for the certificate is
Remote Admin Server-Cert `cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Certificate Manager is installed.
- The CN component in both the subject name and issuer name of the certificate is set to `CN=SSLserver cert-<instance_id>`.

Like any certificate, the remote administration server certificate has a validity period. You must renew the certificate before it expires. For instructions to renew a certificate, see “Renewing Certificates for the Subsystems” on page 515.

Note that the “SSL Server for Remote Admin” option in the Certificate Setup Wizard allows you to renew the remote administration certificate by submitting the request to a CA only—it doesn’t allow you to renew the certificate as a self-signed one (as done during installation). If the CA signing certificate of the CA to which you submit the renewal request is based on the DSA key algorithm, then resulting certificate will be unusable because Netscape Console doesn’t support the DSA algorithm.

If you want to self sign the certificate, you must use `certutil` and `keyutil` tools to extract the key ID from the key database first and then generate a certificate for the key. The steps below outline how to use these tools to renew the certificate. Be sure to check the *CMS Command-Line Tools Guide* for details on `certutil` and `keyutil` tools to customize your commands to suit your requirements.

1. Note the name (also called nickname) of the remote administration SSL server certificate; the default name is
Remote Admin Server-Cert `cert-<instance_id>`.

2. Stop Certificate Management System.
3. Open a command window.
4. Go to this directory: <server_root>/cert-<instance_id>/config
5. Enter the command below, replacing <certname> with the name of the remote administration SSL server certificate. You may use the -h <tokenname> argument to specify whether the certificate database is on a particular hardware or software token.

```
certutil -L -n "<certname>"
```

For example, your command might look like this:

```
certutil -L -n "Remote Admin Server-Cert cert-firefly"
```

You should see detailed information about the remote administration SSL server certificate.

6. Locate the "Subject Public Key Info:" section and then the Modulus section. For example:

```
RSA Public Key:
```

```
Modulus:
```

```
00:f6:9e:71:37:62:af:7c:46:af:cb:bf:1e:d8:1a:
64:0b:5e:71:e2:d8:ec:88:18:6d:eb:32:65:6f:f2:
18:4b:ef:b3:70:ae:61:de:6f:21:d5:4e:0e:7b:9b:
b7:42:98:94:1c:d7:46:42:53:39:db:10:07:6c:b8:
75:7e:94:18:b5
```

7. Note the second and third byte (f69e in the above example) in the modulus; this is the short key ID for the certificate.
8. Delete the certificate you want to renew.
9. Run the certutil command again to regenerate the certificate for the correct key and to add the resulting certificate to the database. Be sure to use the same name for the certificate and to add the required certificate extensions, such as the Key Usage extension. A sample command syntax is below:

```
certutil -S -k <shortkeyID> -y rsa|dsa -n "<certname>"
-s "<subject>" -t "<trustargs>" -x -m <serial-number>
-v <valid-months> -d <certdir> -1)
```

For example, your command might look like this:

```
certutil -S -k f69e -y rsa -n "Remote Admin Server-Cert
cert-firefly" -s "cn=SSLserver cert-firefly" -t "u,u,u" -x -m 3
-v 12 -d . -1
```

10. Restart Certificate Management System.

Registration Manager's Key Pairs and Certificates

The Registration Manager uses the following certificates:

- Signing Key Pair and Certificate
- SSL Server Key Pair and Certificate
- Remote Administration Server Certificate

Signing Key Pair and Certificate

Every Registration Manager you have installed has a certificate, identified as the *Registration Manager signing certificate*, whose public key corresponds to the private key the Registration Manager uses to sign certificate requests before sending them to the Certificate Manager for signing. The Registration Manager's signature provides persistent proof to the Certificate Manager that the Registration Manager has processed the request. The first time you generated this certificate is when you installed the Registration Manager. The default nickname for the certificate is `raSigningCert cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Registration Manager is installed.

The Registration Manager's signing certificate was issued by the CA to which you submitted the certificate signing request. You might have submitted the request to an internally deployed CA or a public CA. To find out the issuer name, follow the instructions in "Viewing the Certificate Database Content" on page 523.

If you configure the Registration Manager to function as a *trusted manager* to another subsystem, the Registration Manager uses its signing certificate for SSL client authentication to the subsystem; this is the default configuration. For details, see "Trusted Manager's Certificate for SSL Client Authentication" on page 408.

SSL Server Key Pair and Certificate

Every Registration Manager you have installed has at least one *SSL server certificate*. The first time you generated this certificate is when you installed the Registration Manager. The default nickname for the certificate is `Server-Cert cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Registration Manager is installed.

The Registration Manager's SSL server certificate was issued by the CA to which you submitted the certificate signing request. You might have submitted the request to an internally deployed CA or a public CA. To find out the issuer name, follow the instructions in "Viewing the Certificate Database Content" on page 523.

The Registration Manager uses its SSL server certificate to do SSL server-side authentication to the following:

- The end entity services interface (the HTTPS port)
- The Registration Manager Agent Services interface

By default, the Registration Manager uses a single SSL server certificate for authentication purposes. However, you can request and install additional SSL server certificates for the Registration Manager. For example, you can configure the Registration Manager to use separate server certificates for authenticating to iPlanet Console, the end entity services interface, and the Registration Manager Agent Services interface. For instructions, see “Configuring the Server to Use Separate SSL Server Certificates” on page 500.

NOTE If you installed the Registration Manager with a Data Recovery Manager, both subsystems use the same SSL server certificate.

Remote Administration Server Certificate

Similar to the Certificate Manager, the Registration Manager too has a remote administration server certificate. For details, see “Remote Administration Server Certificate” on page 457.

Data Recovery Manager’s Key Pairs and Certificates

The Data Recovery Manager uses the following key pairs and certificates:

- Transport Key Pair and Certificate
- Storage Key Pair
- SSL Server Key Pair and Certificate
- Remote Administration Server Certificate

Transport Key Pair and Certificate

Every Data Recovery Manager you have installed has a *Data Recovery Manager transport certificate*. The public key of the key pair that is used to generate the transport certificate is used by the client software to encrypt an end user's *encryption private key* before it is sent to the Data Recovery Manager for archival; only those clients capable of generating dual-key pairs (one for signing and one for encryption) use the transport certificate. For more information on how this certificate is used, see “Key Archival Process” on page 761.

The first time you generated this certificate is when you installed the Data Recovery Manager. The default nickname for the certificate is `kraTransportCert cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Data Recovery Manager is installed.

The transport certificate was issued by the CA to which you submitted the certificate signing request. You might have submitted the request to the Certificate Manager that is installed in the same instance, internally deployed another CA, or a public CA. To find out the issuer name, follow the instructions in “Viewing the Certificate Database Content” on page 523.

Storage Key Pair

Every Data Recovery Manager you have installed has a *Data Recovery Manager storage key pair*. The first time you generated this key pair is when you installed the Data Recovery Manager.

The Data Recovery Manager uses the public component of this key pair to encrypt (or wrap) end users' encryption private keys during the key archival operation; it uses the private component to decrypt (or unwrap) the archived key during the recovery operation. That is, the public key is used to encrypt the key repository the server uses to store end users' encryption private keys. For more information on how this key pair is used, see Chapter 22, “Setting Up Key Archival and Recovery.”

Note that the public component of the storage key pair is not certified; there is no certificate that corresponds to the public key.

Keys encrypted with the storage key can be retrieved only by authorized key recovery agents. For details, see “Key Recovery Agents and Their Passwords” on page 765.

SSL Server Key Pair and Certificate

Every Data Recovery Manager you have installed has at least one *SSL server certificate*. The first time you generated this certificate is when you installed the Data Recovery Manager. The default nickname for the certificate is `Server-Cert cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Data Recovery Manager is installed.

The Data Recovery Manager's SSL server certificate was issued by the CA to which you submitted the certificate signing request. You might have submitted the request to the Certificate Manager that is installed in the same instance, an internally deployed CA, or a public CA. To find out the issuer name, follow the instructions in "Viewing the Certificate Database Content" on page 523.

The Data Recovery Manager uses its SSL server certificate to do SSL server-side authentication to the following:

- The end entity services interface (the HTTPS port)
- The Data Recovery Manager Agent Services interface

By default, the Data Recovery Manager uses a single SSL server certificate for authentication purposes. However, you can request and install additional SSL server certificates for the Data Recovery Manager. For example, you can configure the Data Recovery Manager to use separate server certificates for authenticating to iPlanet Console, the end entity services interface, and the Data Recovery Manager Agent Services interface. For instructions, see "Configuring the Server to Use Separate SSL Server Certificates" on page 500.

NOTE If you installed the Data Recovery Manager with a Certificate Manager or Registration Manager, both subsystems use the same SSL server certificate.

Remote Administration Server Certificate

Similar to the Certificate Manager and Registration Manager, the Data Recovery Manager too uses a remote administration server certificate. For details, see section "Remote Administration Server Certificate" on page 457.

Online Certificate Status Manager's Key Pairs and Certificates

The Online Certificate Status Manager uses the following certificates:

- OCSP Signing Key Pair and Certificate
- SSL Server Key Pair and Certificate
- Remote Administration Server Certificate

OCSP Signing Key Pair and Certificate

Every Online Certificate Status Manager you have installed has a certificate, identified as the *Online Certificate Status Manager signing certificate*, whose public key corresponds to the private key the Online Certificate Status Manager uses to sign OCSP responses before sending them to OCSP-compliant clients. The Online Certificate Status Manager's signature provides persistent proof to an OCSP-compliant client that the Online Certificate Status Manager has processed the request. The first time you generated this certificate is when you installed the Online Certificate Status Manager. The default nickname for the certificate is `ocspSigningCert cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Online Certificate Status Manager is installed.

The Online Certificate Status Manager's signing certificate was issued by the CA to which you submitted the certificate signing request. You might have submitted the request to an internally deployed CA or a public CA. To find out the issuer name, follow the instructions in "Viewing the Certificate Database Content" on page 523.

SSL Server Key Pair and Certificate

Every Online Certificate Status Manager you have installed has at least one *SSL server certificate*. The first time you generated this certificate is when you installed the Online Certificate Status Manager. The default nickname for the certificate is `Server-Cert cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Online Certificate Status Manager is installed.

The Online Certificate Status Manager's SSL server certificate was issued by the CA to which you submitted the certificate signing request. You might have submitted the request to an internally deployed CA or a public CA. To find out the issuer name, follow the instructions in "Viewing the Certificate Database Content" on page 523.

The Online Certificate Status Manager uses its SSL server certificate to do SSL server-side authentication to the Online Certificate Status Manager Agent Services interface.

By default, the Online Certificate Status Manager uses a single SSL server certificate for authentication purposes. However, you can request and install additional SSL server certificates for the Online Certificate Status Manager. For example, you can configure the Online Certificate Status Manager to use separate server certificates for authenticating to iPlanet Console and the Online Certificate Status Manager Agent Services interface. For instructions, see “Configuring the Server to Use Separate SSL Server Certificates” on page 500.

Remote Administration Server Certificate

Similar to the Certificate Manager and Registration Manager, the Online Certificate Status Manager too uses a remote administration server certificate. For details, see section “Remote Administration Server Certificate” on page 457.

Tokens for Storing CMS Keys and Certificates

A token is a hardware or software device that performs cryptographic functions and optionally stores public-key certificates, cryptographic keys, and data defined by the application using the cryptographic services. Alternatively, a token can also be considered as a device that you can use to generate and store your key pairs and corresponding certificates.

Certificate Management System defines two types of tokens, *internal* and *external*, for storing key pairs and certificates that belong to the Certificate Manager, Registration Manager, Data Recovery Manager, and Online Certificate Status Manager.

NOTE Only those who have the password that protects a token can access it. For information on changing the password that protects a token, use the command-line utility called the Key Database Tool, which is explained in the *CMS Command-Line Tools Guide*. To locate an online version of this document, see “Where to Go for Related Information” on page 29.

Internal Token

An internal (software) token refers to a pair of software files, usually called *certificate database* and *key database*, that Certificate Management System uses to generate and store its key pairs and certificates. Certificate Management System automatically generates these files in the file system of its host machine when you choose to use the internal token for the first time. These files were created for you during CMS installation if you chose to use the internal token for key-pair generation.

In the CMS host system, the certificate database is identified by the name `cert7.db`; the key database is identified by the name `key3.db`. You can find both these files at this location: `<server_root>/cert-<instance_id>/config`

External Token

An external (hardware) token refers to an external hardware device, such as a smart card, FORTEZZA card, or other crypto card, that Certificate Management System uses to generate and store its key pairs and certificates. If you haven't already done so, consider using external tokens for generating and storing the key pairs and certificates used by Certificate Management System. These devices represent another security measure you can take to safeguard private keys because hardware tokens are sometimes considered more secure than software tokens. For additional details, check the literature provided by hardware-token vendors.

Installing External Tokens

Certificate Management System supports any hardware tokens that are compliant with PKCS#11 version 2.01. For details, see the information provided at this URL:

http://developer.netscape.com/support/faqs/pkcs_11.html

Certificate Management System also supports FIPS 140-1 Level 2 Security requirements, and it supports Level 3 Security requirements on Hardware Security Modules (HSMs) such as those manufactured by Ncipher and Chrysalis.

NOTE For detailed information about FIPS 140-1 security levels and their requirements, see <http://csrc.nist.gov/publications/fips/>.

Both the Certificate Manager and Data Recovery Manager (DRM) can store keys in certified FIPS 140-1 Level 3 tokens. You can enable FIPS 140-1 Level 3 Support during installation. When this option is enabled, the DRM will not set the password on a hardware token device.

For more information, see the following:

- For detailed information on configuring Certificate Server to work with HSMs, see "Configuring Certificate Server for FIPS 140-1 Level 3 Support."
- For detailed information about other manufacturers' hardware tokens, see the manufacturer's website or token documentation.
- For detailed information about FIPS 140-1 levels of support, see <http://csrc.nist.gov/publications/fips/>.

Installing Level 2 External Tokens

To use external encryption devices or tokens, you need to take the following steps:

- Step 1. Install the Cryptographic Device
- Step 2. Install the PKCS #11 Module

Step 1. Install the Cryptographic Device

To install the drivers provided by the device manufacturer, follow the instructions that came with the device. When you install a hardware token, you are given an opportunity to name it; be sure to use a name that will help you identify the token later.

Step 2. Install the PKCS #11 Module

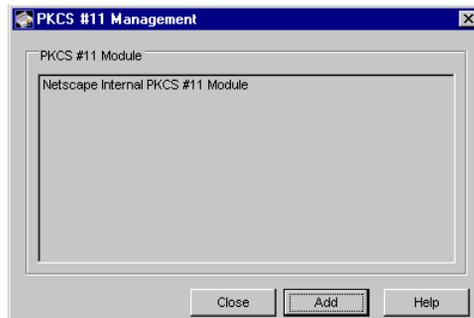
PKCS #11 is a standard set of APIs and shared libraries used by Netscape and a number of encryption vendors. PKCS #11 isolates an application from the details of the cryptographic device, thus enabling the application to provide a unified interface for PKCS #11-compliant cryptographic devices.

The PKCS #11 module implemented in Certificate Management System (in Netscape Administration Server) enables it to support cryptographic devices supplied by many different manufacturers. Specifically, it allows Certificate Management System to plug in shared libraries or DLLs supplied by manufacturers of external encryption devices and use them for generating and storing keys and certificates for the CMS managers.

There are two ways in which you can install a PKCS #11 module, by using the interface provided within iPlanet Console or by using the command-line utility named `modutil`. Both the methods are documented below.

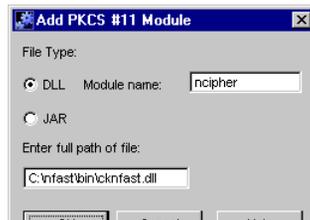
- To install the PKCS #11 module using iPlanet Console:
 - a. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
 - b. From the Console menu, choose Manage PKCS#11.

The PKCS #11 Management window appears.



- c. Click Add.

The Add PKCS #11 Module window appears.



- d. Enter information as appropriate. If you choose JAR as your file type, you are required to provide the path to the JAR file that contains the DLLs. If you choose DLL as your file type, in addition to the path to the DLL you are also required to provide a name for the module you’re attempting to install (so as to help you identify it easily later). The sample in the figure shows how you would install an nCipher™ token.

Pick DLL to add a UNIX shared/dynamic library, which on a Solaris machine is identified with the `.so` extension.

- e. Click OK.
- To install the PKCS #11 module using the `modutil` tool:

- a. Locate the CMS instance for which you want to install the PKCS #11 module.
- b. Open a terminal window.
- c. Go to the configuration directory of Administration Server; it is located here: `<server_root>/admin-serv/config`

- d. At the prompt, enter this command:

```
<server_root>/shared/bin/modutil -dbdir . -nocertdb
-create
```

This creates the required security module database file (`secmod.db`) in the configuration directory.

- e. At the prompt, enter this command:

```
<server_root>/shared/bin/modutil -dbdir . -nocertdb
-add <module_name> -libfile <library_file>
```

`<library_file>` specifies the path to the DLL or other library file containing the implementation of the PKCS #11 interface module.

`<module_name>` specifies the name of the PKCS #11 module (which you specified in Step 1 when you installed the drivers).

For example, if you are installing a Litronic™ token, the command would look like this:

```
<server_root>/shared/bin/modutil -dbdir . -nocertdb
-add CryptOS -libfile core32
```

- f. Copy the new `secmod.db` (`secmodule.db` on Unix) to the following location: `<server-root>/cert-<instance_id>/config/`

This overwrites the old `secmod.db` in this directory.

Installing Level 3 External Tokens

This document provide details for configuring Sun™ ONE Certificate Server 4.7 to work with Hardware Security Modules (HSM) such as those manufactured by Ncipher and Chrysalis. Topics included in this document are:

- Overview of Configuration Steps
- Part 1: Install the HSM
- Part 2: Install and Configure Certificate Server

For detailed information about FIPS 140-1 levels of support, see <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>.

Overview of Configuration Steps

Configuring support for FIPS 140-1 Level 3 security is a two-part process. For Part 1, see the documentation that comes with the HSM. For Part 2, this document provides detailed instructions.

1. Install the HSM.
2. Install and Configure Certificate Server.
 - a. Install of Certificate Server.
 - b. Link the HSM manufacturer's library to Certificate Server.
 - c. Configure Certificate Server

Part 1: Install the HSM

For detailed instructions, see the documentation that comes with the HSM, or visit the manufacturer's website.

For detailed information about Ncipher HSM products, go to the Ncipher website at http://www.ncipher.com/safebuilder/codesafe_specs.html

For detailed information about Chrysalis HSM products, go to the Chrysalis website at http://www.chrysalis-its.com/trusted_systems/luna_ca3.htm.

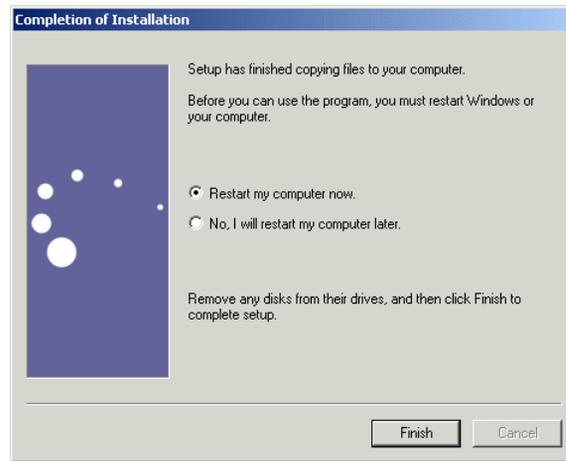
Part 2: Install and Configure Certificate Server

In this part, you run the Certificate Server setup program, and then you run the Installation Wizard to create the first administrator.

2a. Install Certificate Server.

1. Run the Certificate Server installation script.
 - To run the installation script in Windows, open the distribution directory for the system software you are using and double-click the file `setup.exe`.
 - To run the installation script in Solaris, change to the distribution directory (where you have downloaded the distribution files) and execute the file `setup`.
2. Proceed through the Setup program following the instructions in the *Installation and Setup Guide*. When you reach the end of the program, the first phase of the installation is complete.

Figure 14-1 The Windows version of the Setup program uses a GUI; the Solaris version is text-base.



2b. Link the HSM library to Certificate Server.

In this step, you create the security module database and then add the HSM to that database. The following instructions assume you're using `tsch` (Solaris) or `cmd` (Windows).

1. Go to the admin server config directory of the CMS installation.

Solaris: `#> cd <server-root>/admin-serv/config`

Windows: `D:\> cd <server-root>\admin-serv\config`

2. Set the `LD_LIBRARY_PATH` equivalent:

Solaris:

```
#> setenv LD_LIBRARY_PATH <server-root>/lib:$LD_LIBRARY_PATH
```

Windows:

```
D:\> set PATH=<server-root>\lib;%PATH%
```

3. Create the Certificate Server db.

Solaris:

```
#> ../../shared/bin/modutil -dbdir . -nocertdb -create
```

Windows:

```
D:\> ../../shared\bin\modutil -dbdir . -nocertdb -create
```

4. Link the HSM library to the Certificate Server db.

Solaris:

```
#> ../../shared/bin/modutil -dbdir . -nocertdb -add
    <HSM-manufacturer> -libfile <libraries>/<library>.so
```

where <libraries> is the location of the manufacturer's libraries and <library> is the name of the manufacturer's library file.

Windows:

```
D:\> ..\..\shared\bin\modutil -dbdir . -nocertdb -add Chrysalis
    -libfile <libraries\library>.dll
```

where <libraries> is the location of the manufacturer's dll libraries and <library> is the name of the manufacturer's dll file.

5. Verify that the HSM tokens are now available:

Solaris:

```
#> ../../shared/bin/modutil -dbdir . -nocertdb -list
```

Windows:

```
D:\> ..\..\shared\bin\modutil -dbdir . -nocertdb -list
```

2c. Configure the Cryptographic Tokens.

In this step, you run the Installation Wizard to configure the cryptographic tokens.

1. To begin running the Installation Wizard, follow these steps:

a. If iPlanet Console is not running, start it.

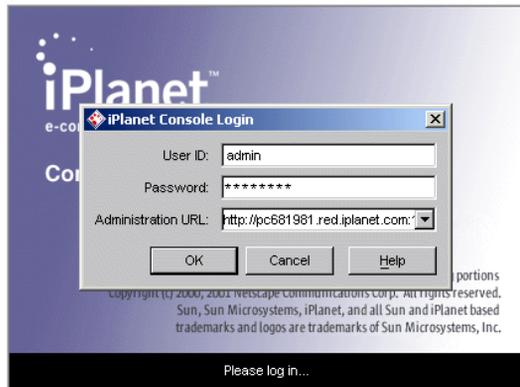
- o On a Windows NT system, click Start>Programs>iPlanet Server Products>iPlanet Console

Alternatively, click the iPlanet Console shortcut in the iPlanet Server Products directory that opens on your desktop after setup completes.

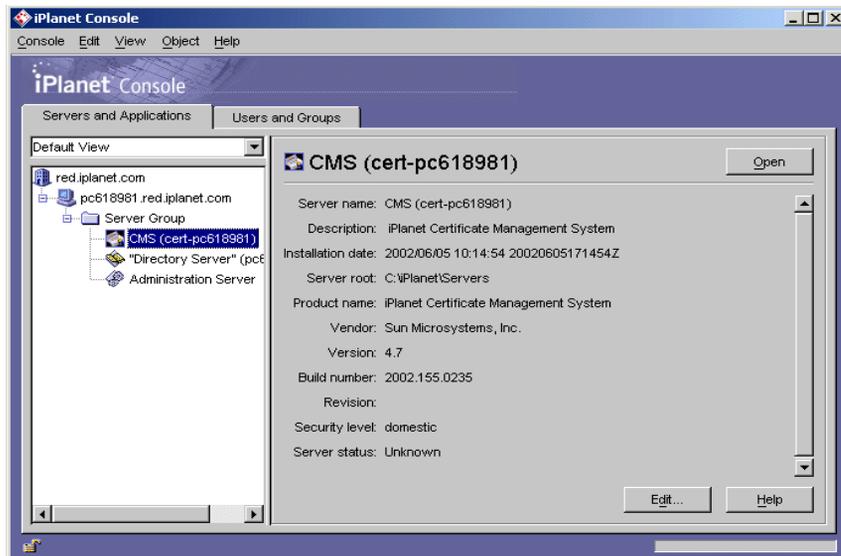
- o On a Unix system, open a command shell, change to the directory /usr/iPlanet/servers, and execute the file startconsole.

b. Log in as admin, giving the password <admin password>.

The main window of iPlanet Console appears. Enter your information, and then click OK.



2. In the navigation tree at the left, open your computer, then open Server Group.
3. Select Certificate Server and double-click it; alternatively, you can also click the Open button on the Certificate Management System panel on the right.

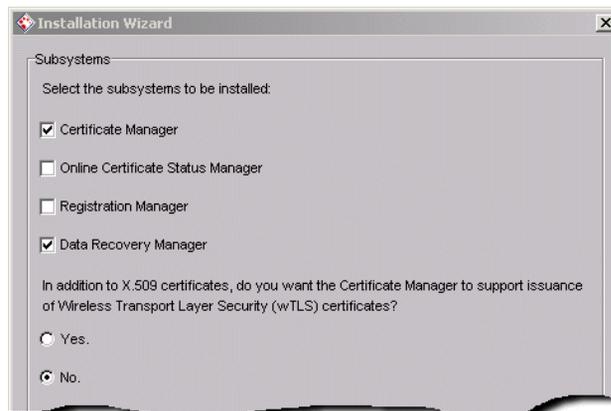


After a few moments, the Installation Wizard appears. You use the wizard to get the initial certificates and set the initial configuration for this instance of Certificate Management System.

Introduction. Click Next.



4. Proceed through the Installation Wizard using the instructions in the *Installation and Setup Guide* until you get to the following screen:



- a. Determine whether you want to install a Data Recovery Manager subsystem. A Data Recovery Manager performs the long-term archival and recovery of private encryption keys for end entities. If you plan to store keys so that you can recover them in the event a key becomes lost, corrupt, or compromised, then check this box. This is highly recommended. If you do not plan to store keys, then leave the box unchecked.
 - b. If you want to co-locate the Certificate Manager and Data Recovery Manager (install instances of both on the same host), then check both of their checkboxes.
 - c. If you want to install the Data Recovery Manager as a stand-alone instance, you can uncheck the Certificate Manager.
5. Continue with the Setup program following instructions in the *Installation and Setup Guide* until you get to a Key-Pair Information screen. Each time you are prompted for Key-Pair Information (see Figure 14-2), repeat this step. Provide the following information, and then click Next:

Token: Choose the FIPS Level 3 hardware token that you specified when you installed the HSM.

FIPS Level 3: Check this checkbox.

Password (again): Enter the PIN/password for the configured token.

Key type: Select a value.

Key length: Select a value.

Figure 14-2 A Key-Pair Information window.

The screenshot shows a window titled "Installation Wizard" with a close button in the top right corner. The main content area is titled "Key-Pair Information for Certificate Manager CA Signing Certificate". Below the title, it says "Select the token (cryptographic device) for the key pair:". There is a "Token:" label followed by a dropdown menu showing "Token1". Below that is a checked checkbox labeled "FIPS Level 3". Underneath, it says "Initialize the selected token:" followed by three password input fields: "Password:" (with asterisks), "Password (again):" (with asterisks), and "Security officer password:". Below these is the section "Specify the key type and key length:" with a "Key type:" dropdown menu showing "RSA" and a "Key length:" dropdown menu showing "1024" followed by "bits". At the bottom, there is a text input field labeled "Enter a value for the customized key length:" followed by "bits".

- Proceed through Installation Wizard using the instructions in the *Installation and Setup Guide* until you get to the Storage Key Creation window (see Figure 14-3). Provide the following information, and then click Next:

Token: Choose a FIPS 140-1 Level 3 token other than the one you specified in the Key-Pair Information window.

FIPS Level 3: Check this checkbox.

Password: Enter the PIN/password for the configured token.

Key type: Select a value.

Key length: Select a value.

Figure 14-3 The Storage Key Creation window.

- Proceed through the Installation Wizard using the instructions in the *Installation and Setup Guide* until you reach the end.

Keys stored in the HSM will now be used for issuing End Entity certificates.

Managing Tokens Used by the Subsystems

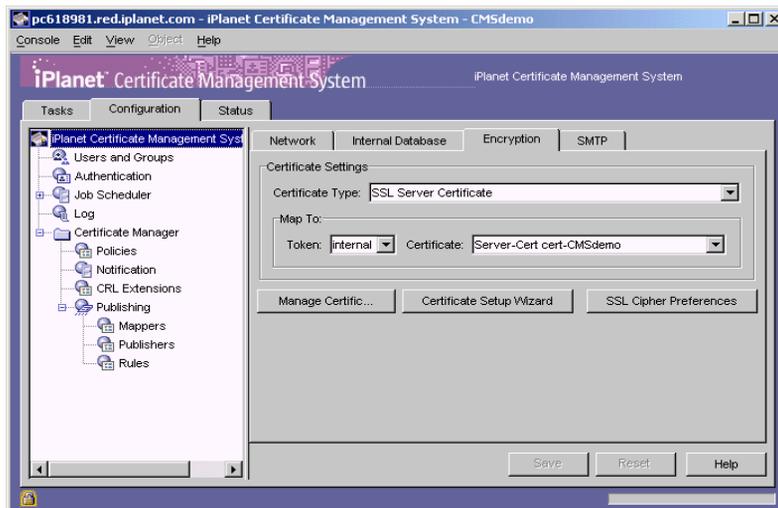
There are two main tasks involved in managing the tokens used by Certificate Management System:

- Viewing Tokens
- Changing a Token's Password

Viewing Tokens

To view a list of the tokens currently installed for a CMS instance:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab, and then in the right pane, select the Encryption tab.



3. In the Map To section, check the Token drop-down list.

It shows the names (as specified when the tokens were installed) of external tokens installed for the currently selected CMS instance. For information on installing external tokens, see “External Token” on page 465.

Changing a Token's Password

The token, internal or external, that stores the key pairs and certificates for the subsystems is protected (encrypted) by a password. To decrypt the key pairs or to gain access to them, you must enter that password. The first time you specified this password is when you used the token the first time, most likely during CMS installation.

It is good security practice to periodically change the password that protects your server's keys and certificates; changing the password periodically minimizes the risk of someone finding out the password. To change a token's password, use the command-line utility called the Key Database Tool, which is documented in *CMS Command-Line Tools Guide*.

Note that the single sign-on password cache stores the passwords for tokens in order to start the server using a single password; for details, see "Required Start-up Information" on page 322. Whenever you change the password, the cache is updated with the new password.

Hardware Cryptographic Accelerators

Certificate Management System allows you to use hardware cryptographic accelerators with external tokens. Many of the accelerators provide the following security features:

- Fast SSL connections—speed is important if you want your Certificate Manager, Registration Manager, or Data Recovery Manager to be able to accommodate a high number of simultaneous enrollment or service requests.
- Hardware protection of private keys—these devices behave like smart cards, in that they do not allow the private keys to be copied or removed from the hardware token. This is important if you are concerned about the risks associated with key theft from an active attacker of your online Registration Manager or Certificate Manager.

Certificate Setup Wizard

Certificate Management System provides a wizard, called the *Certificate Setup Wizard*, which automates the process of requesting and installing the certificates required by the CMS managers—Certificate Manager, Registration Manager, Data Recovery Manager, and Online Certificate Status Manager—installed in the currently selected CMS instance. For details about these certificates, see “Keys and Certificates for the Main Subsystems” on page 450.

The Certificate Setup Wizard is integrated into the CMS window, enabling you to accomplish the following tasks:

- Renew certificates of the CMS managers installed in a CMS instance; *renewing* a certificate means getting a new certificate with the same subject name and public and private key material as that of the existing certificate, but with an extended validity period.
- Request and install new certificates for the CMS managers installed in a CMS instance; *reissuing* or requesting a new certificate means getting a certificate based on a new public and private key pair.
- Install CA certificates in the certificate or trust database of a CMS instance.
- Install CA certificate chains in the certificate database of a CMS instance.

When you start the wizard, which you do by clicking the Certificate Setup Wizard button in the Encryption tab of the CMS window (see the figure on page 476), you are asked to specify whether you want to request or install a certificate. The wizard presents you with the screens appropriate to your choice and walks you through the entire process.

For installing certificates, except for cases when the certificate is self-signed by the CA, you will need to run the wizard twice: once, to request the certificate and once to install the certificate. The reason for this is, if you submit the certificate request to a non-local CA, you will have to wait for the certificate until it is delivered to you.

The following sections explain the process of requesting and installing a certificate by using the Certificate Setup Wizard:

- Using the Wizard to Request a Certificate
- Using the Wizard to Install a Certificate or Certificate Chain

For instructions on getting new certificates, see “Getting New Certificates for the Subsystems” on page 507. For instructions on renewing existing certificates, see “Renewing Certificates for the Subsystems” on page 515.

Using the Wizard to Request a Certificate

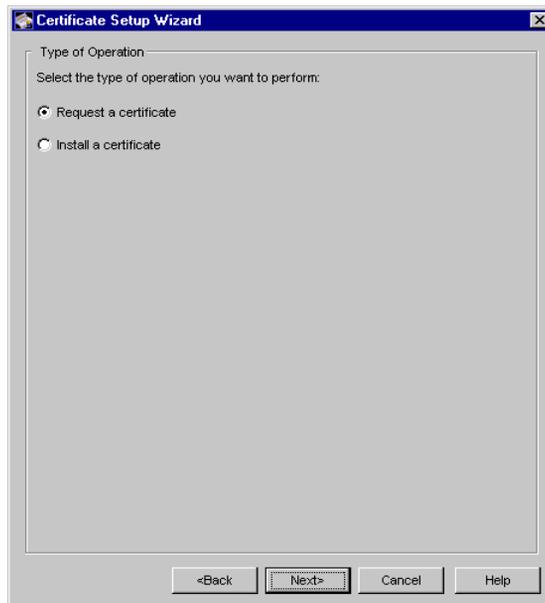
The Certificate Setup Wizard allows you to request any of the certificates used by the Certificate Manager, Registration Manager, Data Recovery Manager, or Online Certificate Status Manager installed in the currently selected CMS instance.

Using the wizard to request a certificate involves the following steps:

- Step 1. Select the Operation
- Step 2. Choose the Certificate
- Step 3. Specify the Key-Pair Information
- Step 4. Specify the Subject Name for the Certificate
- Step 5. Specify the Validity Period
- Step 6. Specify Extensions
- Step 7. Copy the Certificate Signing Request
- Step 8. Check the Certificate Request Status

Step 1. Select the Operation

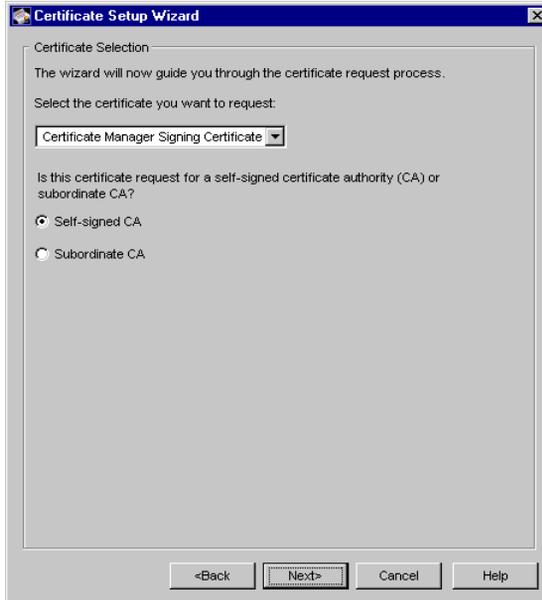
Indicate whether you want to request a certificate or install a certificate.



For the purposes of completing the instructions that follow, assume that you chose to request a certificate.

Step 2. Choose the Certificate

Choose the certificate (by name) that you want to request.



The drop-down list shows various certificates used by the currently selected CMS instance. Choose the one you want to request—which certificates you see in the list depends on the subsystems installed in the currently selected CMS instance. You may see a combination of the following options:

- If a Certificate Manager is installed, the list includes the Certificate Manager’s CA signing, OCSP signing, remote administration server, and SSL server certificates. For details, see “Certificate Manager’s Key Pairs and Certificates” on page 451.
- If a Registration Manager is installed, the list includes the Registration Manager’s signing, remote administration server, and SSL server certificates. For details, see “Registration Manager’s Key Pairs and Certificates” on page 459.
- If a Data Recovery Manager is installed, the list includes the Data Recovery Manager’s transport, remote administration server, and SSL server certificate. For details, see “Data Recovery Manager’s Key Pairs and Certificates” on page 460.

- If a Online Certificate Status Manager is installed, the list includes the Online Certificate Status Manager's signing, remote administration server, and SSL server certificate. For details, see "Online Certificate Status Manager's Key Pairs and Certificates" on page 463.

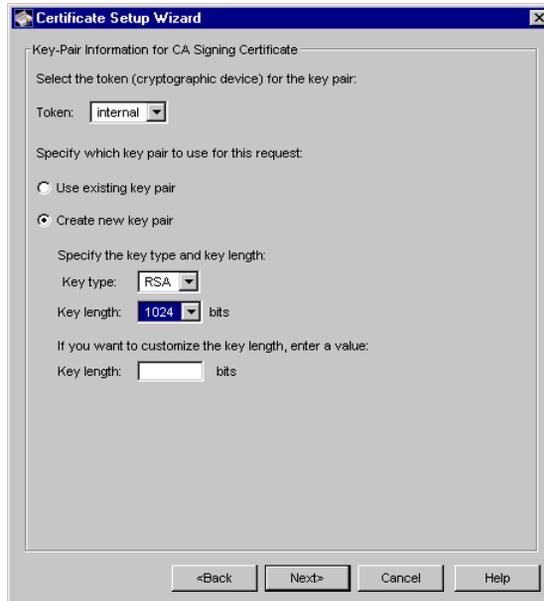
Depending on the certificate you want to generate, choose the one in the drop-down list:

- **Certificate Manager Signing Certificate**—choose this option if you want to request a signing certificate for the Certificate Manager installed in the currently selected CMS instance. If you choose this option, you must also specify whether the certificate request is for a self-signed CA (also known as the root CA) or a subordinate CA.
- **Certificate Manager OSCP Signing Certificate**—choose this option if you want to request an OSCP signing certificate for the Certificate Manager installed in the currently selected CMS instance.
- **Data Recovery Manager Transport Certificate**—choose this option if you want to request a transport certificate for the Data Recovery Manager installed in the currently selected CMS instance.
- **Online Certificate Status Manager Signing Certificate**—choose this option if you want to request a signing certificate for the Online Certificate Status Manager installed in the currently selected CMS instance.
- **Registration Manager Signing Certificate**—choose this option if you want to request a signing certificate for the Registration Manager installed in the currently selected CMS instance.
- **SSL Server Certificate**—choose this option if you want to generate an SSL server certificate request for the CMS managers installed in the currently selected CMS instance.
- **SSL Server Certificate for Remote Admin**—choose this option if you want to generate a remote administration server certificate request for the CMS managers installed in the currently selected CMS instance.
- **Other**—choose this option if you want to generate a certificate request for a certificate that is not generated by a CMS manager by default. For example, in a Certificate Manager, you can use this option to request a CRL signing certificate (see "CRL Signing Key Pair and Certificate" on page 453) or a separate SSL client certificate exclusively for authenticating to the publishing directory. Be sure to specify the certificate type in the adjoining field. By

default only two certificate types are supported: `caCrlSigning` for the CRL signing certificate (see “CRL Signing Key Pair and Certificate” on page 453) and `client` for SSL client certificate (see “Getting an SSL Client Certificate for a Subsystem” on page 502)

Step 3. Specify the Key-Pair Information

Specify the key-pair information for the certificate to be requested.



You need to identify the following:

- The token that contains the key pair for generating the certificate request—the drop-down list shows the names of tokens currently installed for the selected CMS instance; these are the tokens you can use now.
 - The internal token is identified as *internal*. You should choose this option if the key pair for the certificate you chose in the previous step is stored in the local key database.
 - The names of external tokens vary, matching the names specified when the tokens were installed. You should choose this option if the key pair for the certificate you chose in the previous step is in an external cryptographic device. If you don't see the token you want to use, exit from the wizard, make sure the token is installed properly, restart the server, and repeat the process. For information on using or installing external tokens, see “Installing Level 2 External Tokens” on page 466.

- The key pair for generating the certificate request—you can choose to generate the certificate request based on an existing or a new key pair.
 - If you want to renew the certificate you selected in the previous step, use the existing key pair for generating the request. For example, you can extend the validity period of a certificate by renewing it.

To generate a certificate request based on an existing key pair, select the token that contains the key pair you want to use for generating the request. The wizard automatically selects the key pair that corresponds to the certificate you chose in the previous step.

- If you want a new certificate, use a new key pair for generating the request. For example, you may want to get a new SSL server certificate or may want to replace an existing certificate whose private key has been compromised.

To generate a certificate request based on a new key pair, select the token that can generate the key pair you want to use for generating the request. For example, if you want to generate the key pair using an external cryptographic device, such as a smart card, select that as the token. In addition, you will be required to indicate details, such as the key algorithm and size for the key pair.

- The type and length of the key pair—you are required to provide this information only if you chose to generate the certificate request based on a new key pair. For key type, you can choose RSA or DSA. Be sure to select a key type that the CA (to which you will later submit the request for signing) can certify.

For key length, enter the size in bits.

- If the key type is RSA, the key size can be 512, 1024, 2048, 4098, or custom.
- If the key type is DSA, the key size can be 512, 1024, or custom (must be in increments of 64 bit).

Keep in mind that generating a new key pair takes time—the longer the key length the longer the time the wizard takes to generate it.

Step 4. Specify the Subject Name for the Certificate

Specify the subject name, in distinguished name (DN) format, for the certificate to be requested. Note that you will see this screen only if you chose to generate the certificate for a new key pair.

You can either enter values for individual DN attributes required to build the subject DN or build the complete subject DN string yourself. If you enter values for individual DN attributes, the wizard constructs the subject DN string.

If you want to enter values for individual DN components, provide the following information:

- **Common name**—enter the name as appropriate. Except for the SSL server certificate, the common name format can be a descriptive name of up to 255 characters. For example, you can name the Certificate Manager’s signing certificate as “Root CA for ABC Corporation”; similarly, you can name the Registration Manager’s signing certificate as “Registration Authority for North America”. For a SSL server certificate, the name must be the fully qualified host name of Certificate Management System in this form:
`<machine_name>.<your_domain>.<domain>`

To determine the machine and domain names, go to iPlanet Console, and locate the CMS host in the navigation tree.

- **Organizational unit**—enter the organizational unit the server belongs to. For example, `Corporate Security`.
- **Organization**—enter a description that identifies your organization. For example, `Siroe Corporation`.
- **Locality**—enter the name of the city where your business is located. For example, `Mountain View`.
- **State or province**—enter the name of the state or province where your business is located. For example, `California`.
- **Country**—enter the name of the country where your business is located. For example, `US`.

Step 5. Specify the Validity Period

You need to complete this step only if you chose to generate a self-signed CA certificate request.

The screenshot shows a dialog box titled "Certificate Setup Wizard" with a sub-title "Validity Period for CA Signing Certificate". The main instruction is "Specify the validity period for the certificate:". Below this, there are two rows of date and time input fields. The first row is labeled "Begin on:" and the second row is labeled "Expire on:". Each row has six input boxes corresponding to the labels YYYY, MM, DD, HH, mm, and SS. The "Begin on:" row has values 1999, 01, 05, 10, 30, and 00. The "Expire on:" row has values 2002, 01, 05, 10, 30, and 00. At the bottom of the dialog box, there are four buttons: "<Back", "Next>", "Cancel", and "Help".

Specify the starting and ending dates of the validity period for the certificate request you want to generate. You can also specify the time at which the validity period should start and end on those dates.

The default validity period is two years.

Step 6. Specify Extensions

You need to complete this step only if you chose to generate a CA signing certificate request for a Certificate Manager (deployed as either the root CA or a subordinate CA).



This screen allows you to set the standard X.509 version 3 extensions and Netscape-defined extensions for the certificate to be requested. The required extensions are chosen by default. If you want to change the default choices, be sure to read the general guidelines explained in “Certificate and CRL Extensions” in Appendix C of *CMS Plug-Ins Guide*.

Also note that certificate extensions are required if you are setting up a hierarchy of certificate authorities (CAs). Subordinate CAs must have certificates that include the extension identifying them as either a subordinate SSL CA (which allows them to issue certificates for SSL) or a subordinate email CA (which allows them to issue certificates for secure email). If you disable certificate extensions, you will not be able to set up CA hierarchies. For more information on CA hierarchies, see “Certificate Hierarchies” in Appendix D of *Managing Servers with iPlanet Console*.

You can set the following extensions:

- **Basic constraints**—select this option if you want to set any of the basic constraints extension bits in the certificate you are requesting. When you select the option, the associated fields are enabled. You should select the ones you want to set.

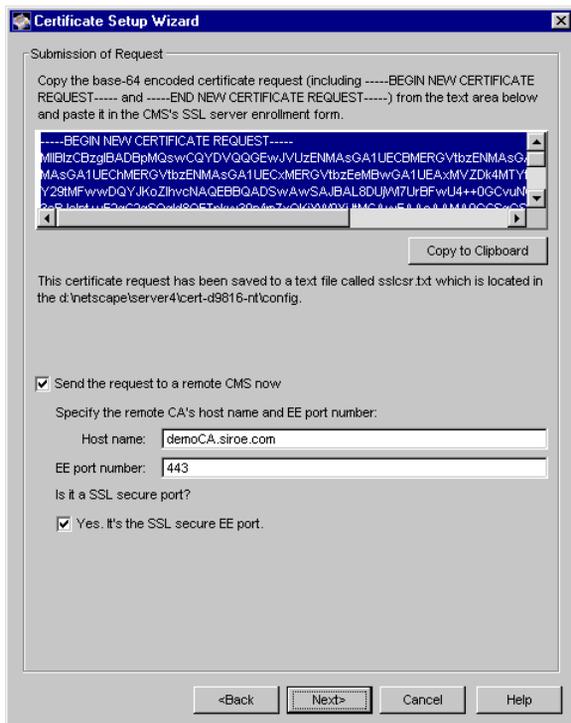
- Netscape certificate type—select this option if you want to set any of the Netscape Certificate Type extension bits in the certificate you are requesting. When you select the option, the associated fields are enabled. You should select the ones you want to set.
- Authority key identifier—select this option if you want to set the authority key identifier extension in the certificate you are requesting.
- Subject key identifier—select this option if you want to set the subject key identifier extension in the certificate you are requesting.
- Key usage—select this option if you want to set the key usage extension in the certificate you are requesting. If you choose this option, the digital signature (bit 0), non repudiation (bit 1), key Certificate Sign (bit 5), and CRL sign (bit 6) bits are set by default. The extension is marked critical as recommended by the PKIX standard and RFC 2459 (see <http://www.ietf.org/rfc/rfc2459.txt> for a description of the Key Usage extension).
- Extension in MIME 64 DER encoding—select this option if you want to specify any custom extension. When you select the option, the associated text field is enabled. You should paste your extension (in MIME 64 DER encoded format) into the text field.

Certificate Management System provides tools that generate MIME-64 encoded blobs for many standard extensions. You can use these tools for generating MIME-64 encoded blobs for any extensions that you may want to include in CA and other certificate requests. For details about these tools, check this directory in your CMS installation: `<server_root>/bin/cert/tools`

Note that the text field provided for pasting the extension in general accepts a single extension blob. If you want to add multiple extensions, you should use the `ExtJoiner` program, which is also provided in the tools directory mentioned above. For instructions to use the program, see Chapter 5, “Extension Joiner Tool” of *CMS Command-Line Tools Guide*.

Step 7. Copy the Certificate Signing Request

Based on the information you've entered in the previous steps, the wizard now displays the certificate signing request (CSR).



The request is in a base-64 encoded PKCS #10 format and is bounded by the marker lines -----BEGIN NEW CERTIFICATE REQUEST----- and -----END NEW CERTIFICATE REQUEST-----. An example is show below:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIICJzCCAZCgAwIBAgIBAzANBgkqhkiG9w0BAQQFADBC6SAwHgYDVQQKEXdOZXRz
Y2FwZSBDb21tdW5pY2F0aW9ucngjhnMVQ2VydG1maWNhdGUgQXV0aG9yaXR5MB4
XDTK4NDgyNzE5MDAwMFoXDTK5MDIyMzE5MDAwMnBjdGngYoxIDAeBgNVBAoTF05l
dHNjYXB1IEENvbW1lbmljYXRpb25zMQ8wDQYDVQQLEWZwZw9wbGUxZzAVBgoJkiaJ
kIsZAEBEwdzdXByaXlhMRcwFQYDVQQDEw5TdXByaXlhIFNoZXR0eTEjMCEGCSqGS
Ib3DbndgJARYUc3Vwcm15Yhvfggsvwryw4y7214vaOBgNVHQ8BAf8EBAMCBLAwFA
YJYIZIAyB4QgEBAQHBAQDAgCAMA0GCSqGSIb3DQEBAUAA4GBAFi9FzyJlLmS+kz
sue0kTXawbwamGdYql2w4hIBgdR+jWeLmD4CP4x
-----END NEW CERTIFICATE REQUEST-----
```

The wizard also copies the CSR to a text file it creates in the configuration directory, which is located at `<server_root>/cert-<instance_id>/config`. The name of the text file varies depending on for which key pair you generated the request. Table 14-1 lists them.

Table 14-1 Names of files created for certificate signing requests

Filename	Certificate Signing Request
<code>cacsr.txt</code>	Certificate Manager CA signing certificate
<code>ocspcsr.txt</code>	Certificate Manager OCSP signing certificate
<code>racsr.txt</code>	Registration Manager signing certificate
<code>kracsr.txt</code>	Data Recovery Manager transport certificate
<code>ocspcsr.txt</code>	Online Certificate Status Manager signing certificate
<code>sslcsr.txt</code>	SSL server certificate
<code>sslcsrradm.txt</code>	Remote administration server certificate
<code>othercsr.txt</code>	Other certificates, such as Certificate Manager CRL signing certificate or SSL client certificate

Do not modify the CSR; you must send it to the CA as it is. You can either submit the request automatically or copy the request and manually submit it to the CA by visiting the URL it provides for this purpose. Note that the wizard's auto-submission feature—a feature that enables you to send the request directly to a remote CA without having to manually copy the base-64 encoded certificate and paste the request in an enrollment form—can be used to submit requests to a remote Certificate Manager or Registration Manager (if that Certificate Manager is configured to receive requests via the Registration Manager) only. It can't be used for submitting the request to a third-party CA. For a third-party CA, you must manually copy the certificate request and paste it into the text area provided in the CA's enrollment form.

Sending the CSR Automatically to a CMS Manager

To send the certificate signing request (CSR) automatically to a Certificate Manager:

1. Type the appropriate values in the following fields:

Send the request to a remote CMS now. Select this option.

Host name. Type the fully-qualified host name (in the <machine_name>.<your_domain>.<domain> format) of the Certificate Manager to which you want to submit your request automatically. For example, CAmachine.siroe.com.

EE port number. Type the end-entity port number. For example, 80.

Yes, it's the SSL secure server port. Select this option if the end entity port number you specified is the SSL port for end entities.

2. Click Next to submit your request to the CA.

The Certificate Manager returns a request ID for your request. Note the request ID as you can use it later to get the certificate from the Certificate Manager to which you submitted the request.

The request you submitted gets queued for agent approval—an agent needs to process and approve the certificate request, which the CA signs then and delivers back to the email address specified in the request. You can contact the CA agent to find out when the certificate will be delivered to you. If you have agent privileges to the Certificate Manager, you can log in to its agent interface and approve the request yourself.

3. Once the certificate has been issued, you can use the request ID to import the certificate into the wizard. Alternatively, you can also install the certificate following the instructions in “Using the Wizard to Install a Certificate or Certificate Chain” on page 493.

Sending the CSR Manually to an Internal CA

The following instructions assume that your internally deployed CA is a Certificate Manager and that you are using the default HTML forms provided for end-entity enrollment. If you have customized these forms, you should follow the appropriate instructions.

To send the certificate signing request (CSR) manually to an internal CA:

1. Copy the CSR, including the marker lines -----BEGIN NEW CERTIFICATE REQUEST----- and -----END NEW CERTIFICATE REQUEST-----, to a text file. If you are running the wizard on a Windows NT system, you can also copy the CSR to the Windows clipboard. In a Unix system, you may have to open an application, such as Netscape Composer, with a clipboard.
2. Open a web browser window.

3. Enter the URL to the CA's home page.

By default, the CA's home page is the end entity services interface. Depending on the port at which the CA is listening to end-entity requests (see "End-Entity Ports" on page 383) the URL to the end entity services is one of the following:

```
http://<hostname>:<end_entity_port> or
https://<hostname>:<end_entity_port>, where <hostname> is in the form
<machine_name>.<your_domain>.<domain>
```

The end entity services interface appears.

4. Click the Enrollment tab.

5. In the menu list, click the appropriate link:

- If the CSR is for a subordinate CA certificate, in the Server section, click the Certificate Manager link.
- If the CSR is for a Registration Manager's signing certificate, in the Server section, click the Registration Manager link.
- If the CSR is for a Certificate Manager's OCSP signing certificate or Online Certificate Status Manager's signing certificate certificate, in the Server section, click the OCSP Responder link.
- If the CSR is for an SSL server certificate or remote administration server certificate, in the Server section, click the SSL Server link.

6. In the form that appears, enter the required information and paste the CSR from either the clipboard or text file.

For information on how a form works, click the Help button provided on the form. Be sure to include the marker lines, -----BEGIN NEW CERTIFICATE REQUEST----- and -----END NEW CERTIFICATE REQUEST-----.

7. Submit the request.

8. When the CA sends you a response, save the information in a text file for future reference or inquiry.

Note that the you submitted gets queued for agent approval—an agent needs to process and approve the certificate request, which the CA signs then and delivers back to the email address specified in the request. You can contact the CA agent to find out when the certificate will be delivered to you. If you have agent privileges to the Certificate Manager, you can approve the request yourself.

9. When you receive the certificate from the CA, you'll need to install it following the instructions in "Using the Wizard to Install a Certificate or Certificate Chain" on page 493.

Sending the CSR to an External CA

An external CA is any public or third-party CA. Before sending the CSR to a public CA, make sure that the CA can issue the certificate you want to request. Also, it is a good idea to read the policy statement published by a CA to see whether the CA imposes any restrictions on the validity period or usage of the certificate.

To send the CSR manually to an external or third-party CA:

1. Copy the CSR, including the marker lines `-----BEGIN NEW CERTIFICATE REQUEST-----` and `-----END NEW CERTIFICATE REQUEST-----`, to a text file. If you are running the wizard on a Windows NT system, you can also copy the CSR to the Windows clipboard. In a Unix system, you may have to open an application, such as Netscape Composer, with a clipboard.
2. Open a web browser window.
3. Navigate to the CA's home page by entering the appropriate URL in the browser window.
4. Locate the form that allows you to submit certificate requests for servers.
5. Enter the required information and paste the CSR from the text file.
6. Submit the request.
7. When the CA sends you a response, save the information in a text file for future reference or inquiry.
8. When you receive the certificate from the CA, install it following the instructions in "Using the Wizard to Install a Certificate or Certificate Chain" on page 493.

Step 8. Check the Certificate Request Status

The wizard now informs you of the status of the request.

- If you requested a self-signed CA certificate, the wizard automatically submits the CSR to the CA. If the CSR includes all the required information, the CA signs the certificate and returns it to the wizard, which then installs it in the appropriate token.
- If you requested any other certificate, you must get the certificate from the CA and install it using the process outlined in "Using the Wizard to Install a Certificate or Certificate Chain" on page 493.

Using the Wizard to Install a Certificate or Certificate Chain

The Certificate Setup Wizard allows you to install or import the following certificates into either an internal or external token used by the currently selected CMS instance:

- Any of the certificates used by a Certificate Manager, Registration Manager, Data Recovery Manager, and Online Certificate Status Manager
- Any other trusted CA certificates (certificates of CAs that you want to trust)
- Certificate chains

A certificate chain typically includes a collection of certificates: the subject certificate, the trusted root CA certificate, and any intermediate CA certificates needed to link the subject certificate to the trusted root. However, the certificate chain the wizard allows you to import must include only CA certificates; none of the certificates can be a user certificate.

In a certificate chain, each certificate in the chain is encoded as a separate DER-encoded object. When the wizard imports a certificate chain, it imports these objects one after the other, all the way up the chain to the last certificate, which may or may not be the root CA certificate. If any of the certificates in the chain already exist in the local certificate database, the wizard replaces them by the ones included in the chain. If the chain includes intermediate CA certificates, the wizard adds them to the certificate database as *untrusted* CA certificates.

The certificate or certificate chain you provide to the wizard for installation must be in one of the data formats supported by the wizard. This is explained in “Data Formats for Installing Certificates and Certificate Chains” on page 494.

Using the wizard to install a certificate or certificate chain involves the following steps, described in detail on page 495:

- Step 1. Select the Operation
- Step 2. Select the Certificate or Certificate Chain
- Step 3. Specify the Location of the Certificate
- Step 4. View the Certificate or Certificate Chain
- Step 5. Install the Certificate or Certificate Chain
- Step 6. Verify the Certificate Status

Data Formats for Installing Certificates and Certificate Chains

The wizard can accept certificates and certificate chains in several data formats. This section briefly explains the data formats recognized by the wizard.

Binary Formats

The wizard can recognize certificates and certificate chains in the following binary formats:

- DER-encoded certificate—This is a single binary DER-encoded certificate.
- PKCS #7 SignedData objects—This is a PKCS #7 SignedData object. The only significant field in the SignedData object is the certificate. In particular, the signature and the contents are ignored. The PKCS #7 format allows multiple certificates to be downloaded at once.
- DER-encoded certificates—These are DER-encoded certificates that may or may not be wrapped in a base-64 encoding package surrounded by the delimiters -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----.
- 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, while the content field is the following structure:

```
CertificateSequence ::= SEQUENCE OF Certificate
```

This format allows multiple certificates to be downloaded at once.

Text Formats

The wizard can also import certificates and certificate chains in text formats. Here's what you should be aware of when using the wizard to install a certificate or certificate chain in text format:

The text format must begin with the following line:

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

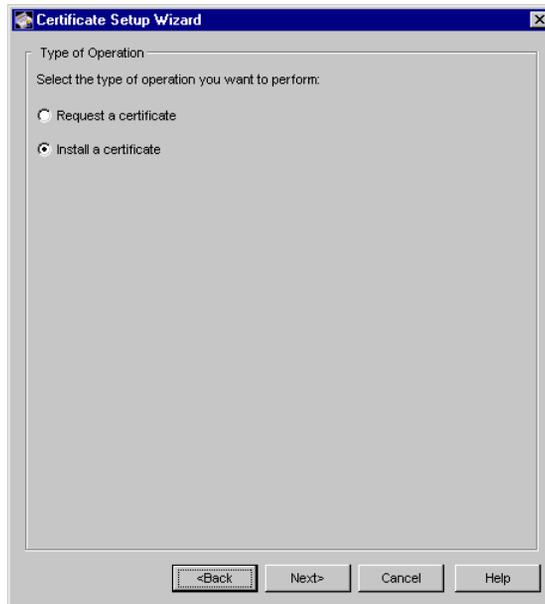
Following this line should be the certificate data, which can be in any of the binary formats described in “Binary Formats” on page 494. This data should be base-64 encoded as described by RFC 1113 (for details, see <http://www.scit.wlv.ac.uk/rfc/rfc11xx/RFC1113.html>).

Following the certificate data must be this line:

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

Step 1. Select the Operation

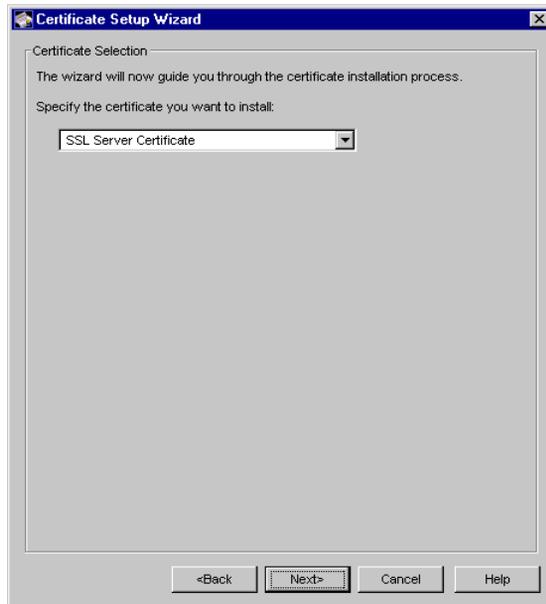
Indicate whether you want to request a certificate or install a certificate.



For the sake of completing the instructions that follow, assume that you chose to install a certificate.

Step 2. Select the Certificate or Certificate Chain

Select the certificate you want to install.



The drop-down list shows various options. Depending on whether you want to install a CMS certificate, any other trusted CA certificate, or a CA certificate chain, choose the appropriate option from the list box:

- **Certificate Manager Signing Certificate**—choose this option if you want to install a CA signing certificate for the Certificate Manager installed in the currently selected CMS instance.
- **OCSP Signing Certificate**—choose this option if you want to install an OCSP signing certificate for the Certificate Manager installed in the currently selected CMS instance.
- **Registration Manager Signing Certificate**—choose this option if you want to install a request signing certificate for the Registration Manager installed in the currently selected CMS instance.
- **Data Recovery Manager Transport Certificate**—choose this option if you want to install a transport certificate for the Data Recovery Manager installed in the currently selected CMS instance.
- **Online Certificate Status Manager Signing Certificate**—choose this option if you want to install a signing certificate for the Online Certificate Status Manager installed in the currently selected CMS instance.

- Keeping the certificate or certificate chain in a text file—the wizard can import a certificate or certificate chain from a text file in *text* as well as *binary* formats; see “Data Formats for Installing Certificates and Certificate Chains” on page 494. If you have copied the certificate or certificate chain to a text file, you will be required to provide the wizard with the absolute path to that file. The file must be located in the host system the wizard is running. If the file is located elsewhere, exit from the wizard, copy the file to the local disk, and restart the wizard.
- Copying the certificate or certificate chain to the text area on the wizard screen—you can paste the certificate or certificate chain into the text area provided by the wizard. This is a text input field, so you can paste the certificate or certificate chain in text format only. For example, if you are installing a certificate, its base-64 encoded certificate blob should look similar to this:

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

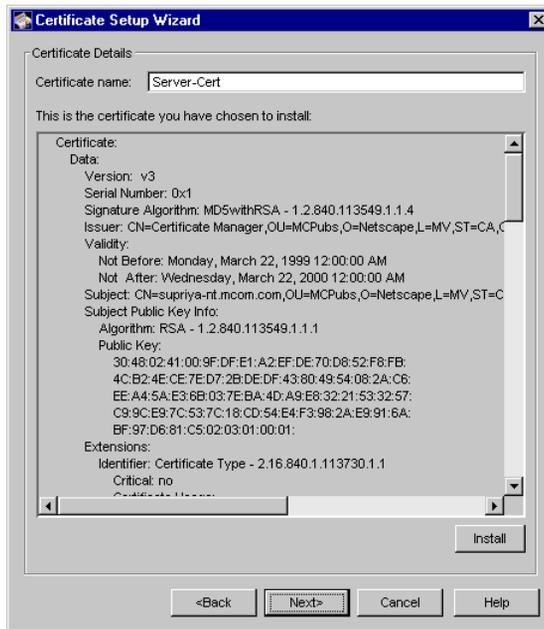
```
MIICKzCCAZSgAwIBAgIBAzANGkqkiG9w0BAQQFADA3MQswCQYDVQQGEwJVUzERMA
8GA1UEChMlTmV0c2NhcnVFTATBgNVBAsTDFN1cHJpeWEncyBDQTAeFw05NzEwMT
gwMTM2MjVaFw05OTEwMTgwMTM2MjVaMEgxCzAJBgNVBAYTAlVTMREwDwYDVQQKEw
hOZXRzY2FwZTENMA8GA1UECxEUHWcWZEXMBUGA1UEAxMOU3Vwcm15YSB0dH
kgwZ8wDQYJKoZIhdfNAQEBBQADgY0AMIGJAoGBAMr6eZiPGfjX3uRjgEjmKiQG7S
dATYzBcABu1AVyd7chRFOGD3wNktbf6hRo6EAmM5R1Askzf8AW7LiQZBcrXpc0k4
du+2j6xJu2MPm8WKuMOTuvzpo+SGXelmHVChEqooCwfdiZywyZNmgaMa2MS6pUkf
QVAgMBAAGjNjA0MBEGCWCGSAGG+EIBAQQEAwIAgD
```

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

- The certificate is at the CMS where your request was sent— if you have previously sent the certificate request to a remote Certificate Manager automatically and have noted the request ID that you received in return, you can use it to retrieve the certificate from the Certificate Manager.

Step 4. View the Certificate or Certificate Chain

The wizard displays the certificate or certificate chain you have chosen to install. Make sure you have chosen the right one; otherwise, use the Back button to go back and locate the right one. Specify a nickname for the certificate.



Step 5. Install the Certificate or Certificate Chain

The wizard shows the certificate or certificate chain information you have selected for installing. You should check the information to make sure that you have chosen the correct one for installing.

After verifying that the certificate you have chosen is the correct one, click the Install button. The wizard installs the certificate or the CA chain in the token you have chosen.

- If you installed a certificate that has been issued by CA whose certificate chain doesn't exist in the certificate database, you must add that CA's certificate chain to the database. To add the CA chain to the database, copy the CA chain to a text file, start the wizard again, and install the CA chain.

- If you installed (or imported) a certificate chain, the wizard adds (to the local trust database) the first certificate in the chain as a trusted CA certificate and any subsequent certificates as untrusted CA certificates. For more information on how the wizard installs a certificate chain, see “Using the Wizard to Install a Certificate or Certificate Chain” on page 493.

Step 6. Verify the Certificate Status

This step is applicable only if you installed a certificate chain.

After you install a certificate chain in the trust database of a CMS instance, check the trust status of each certificate that got installed, and make sure that the correct CA certificates are trusted. For instructions, see “Changing the Trust Settings of a CA Certificate” on page 526.

Configuring the Server's Security Preferences

Configuring a CMS manager's security preferences involves identifying the following:

- The SSL server certificates a server must use for authenticating to the end entity, agent, and administration interfaces. For details, see “Configuring the Server to Use Separate SSL Server Certificates” on page 500.
- The SSL client certificate a Certificate Manager must use for authenticating to the publishing directory (if the Certificate Manager is configured to publish certificates and CRLs to the directory). For details, see “Getting an SSL Client Certificate for a Subsystem” on page 502.
- The version of SSL that an instance of Certificate Management System must use during SSL communication. The latest version is SSL version 3, but many older clients use SSL version 2. Because client authentication is required for performing privileged operations, you must enable SSL version 3 ciphers supported by Certificate Management System. For details, see “Setting Up Cipher Preferences for SSL Communications” on page 504.

Configuring the Server to Use Separate SSL Server Certificates

You can configure a CMS instance to use separate SSL server certificates for authenticating to iPlanet Console, the Agent Services interface, and the end entity services interface.

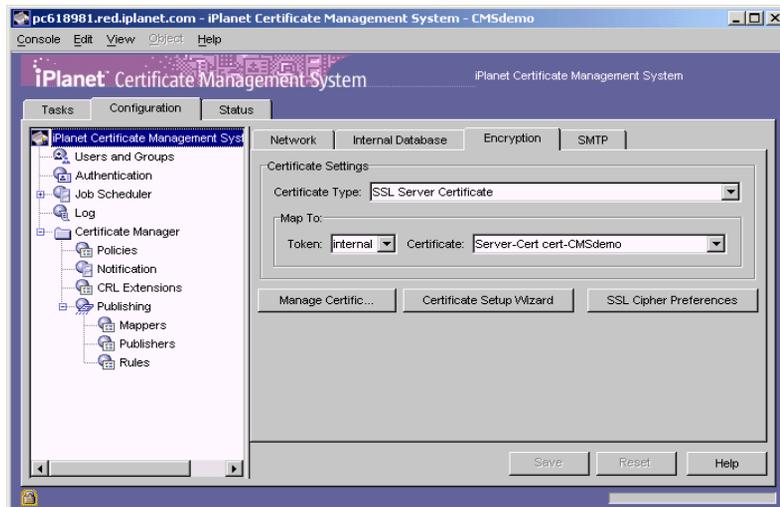
This configuration involves the following steps:

- Step 1. Get the Required SSL Server Certificates
- Step 2: Update the Configuration

Step 1. Get the Required SSL Server Certificates

You must first request and install the required number of SSL server certificates for the particular CMS instance. For instructions, see “Getting New Certificates for the Subsystems” on page 507.

Once you have installed the certificates, you should be able to see them in the list of SSL server certificates in the Encryption tab of the CMS window.



Step 2: Update the Configuration

After you verify that the certificates are installed, configure the server as follows:

1. Stop the CMS instance; see “Stopping Certificate Management System” on page 330.
2. Go to this directory: `<server_root>/cert-<instance_id>config`
3. Open the configuration file (`CMS.cf`) in a text editor.

4. Change the configuration:

- o To change the certificate used for authenticating to the Agent Services interface, locate the `agentGateway.https.nickName` parameter and change its value to the nickname of the new SSL server certificate. For example, if the nickname of the SSL server certificate is `ServerCert_agt`, the configuration should look like this:

```
agentGateway.https.nickName=ServerCert_agt cert-<instance_id>
```

- o To change the certificate used for authenticating to the end-entity services interface, locate the `eeGateway.https.nickName` parameter and change its value to the nickname of the new SSL server certificate. For example, if the nickname of the SSL server certificate is `ServerCert_ee`, the configuration should look like this:

```
eeGateway.https.nickName=ServerCert_ee cert-<instance_id>
```

- o To change the certificate used for authenticating to the administration interface, iPlanet Console, locate the `radm.https.nickName` parameter and change its value to the nickname of the new SSL server certificate. For example, if the nickname of the SSL server certificate is `ServerCert_admin`, the configuration should look like this:

```
radm.https.nickName=ServerCert_admin cert-<instance_id>
```

5. Save your changes.

6. Start the server; see “Starting Certificate Management System” on page 322.

Getting an SSL Client Certificate for a Subsystem

By default, the Certificate Manager uses its SSL server certificate for SSL client authentication to the publishing directory. For details about publishing certificates and CRLs to a directory, see Chapter 19, “Setting Up LDAP Publishing.”

If you want the Certificate Manager to use another certificate for authenticating to the publishing directory, you can do so. This section provides instructions for requesting and installing an SSL client certificate for a Certificate Manager and configuring it to use that certificate for SSL client authentication to the publishing directory.

1. Log in to iPlanet Console; see “Logging In to iPlanet Console” on page 344.

2. Locate the CMS instance for the Certificate Manager, make sure it's started, and then log in to the CMS window of the Certificate Manager.
3. Select the Configuration tab, and then select the Encryption tab.
4. Click the Certificate Setup Wizard button to launch the wizard, which is explained in "Certificate Setup Wizard" on page 478.
5. Select the option to request a certificate and then follow the on-screen prompts to generate a certificate request for the client certificate—in the Certificate Selection window, select `Other` and specify `client` as the certificate type in the associated text field.
6. Once you have the certificate request ready, submit it to a CA so that it can issue a certificate. For general instructions to use the wizard to request a certificate, see section "Using the Wizard to Request a Certificate" on page 479.
7. If you submitted the request to a Certificate Manager and if you have agent privileges for that Certificate Manager, log in to its Agent Services interface, locate the request, and check the request for required extensions. (If you submitted the request to any other CA, you must ask the person managing that CA to make the same changes to the request before approving it.)

Make sure that only the `SSL Client` option for certificate type is selected in the request. For certificates with no Netscape Certificate Type extensions, the Key Usage extension must be included with `Signing` and `Encryption` bits set.

8. Approve the request.
9. Once you have the certificate ready, restart the wizard and install the certificate in the Certificate Manager's database. For general instructions to use the wizard to add a certificate, see "Using the Wizard to Install a Certificate or Certificate Chain" on page 493.

Note that the default nickname for the certificate is `cr1SigningCert cert-<instance_id>`, where `<instance_id>` identifies the CMS instance in which the Certificate Manager is installed.

10. After you've installed the certificate successfully, go to the Tasks tab and stop the Certificate Manager.
11. Configure the Certificate Manager to use this certificate.

After you install the certificate, configure the Certificate Manager to use the new certificate for SSL client authentication to the publishing directory. For instructions, see "Step 5. Identify the Publishing Directory" on page 680.

Setting Up Cipher Preferences for SSL Communications

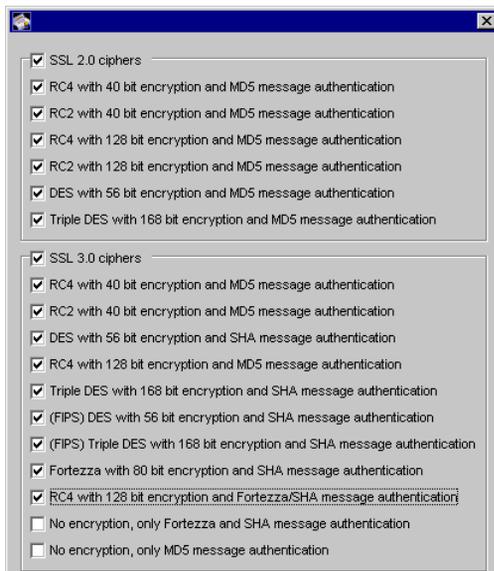
A *cipher* is the algorithm used in encryption. Some ciphers have *stronger* encryption capabilities than others. Generally speaking, the more bits a cipher uses during encryption, the harder it is to decrypt the data.

When a client initiates an SSL connection with Certificate Management System, it lets the server know what ciphers it prefers to use to encrypt information. In any two-way encryption process, both parties must use the same ciphers. A number of ciphers are available; your server needs to be able to use the most popular ones.

SSL Ciphers Supported in Certificate Management System

Figure 14-4 shows the ciphers supported by Certificate Management System (on the server side). The figure shows SSL 2.0 and 3.0 ciphers supported in the domestic (US and Canada) version of Certificate Management System. Note that Certificate Management System 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 Certificate Management System with the same encryption and cryptographic features available in the US and Canada. For more information, see Appendix C, “Export Control Information.”

Figure 14-4 SSL version 2.0 and 3.0 cipher suites supported (in the domestic version)



You can choose ciphers from the SSL 2.0 protocol, as well as from SSL 3.0. To specify which ciphers your server can use, check them in the list of ciphers to enable them. Unless you have a compelling reason not to use a specific cipher, you should check them all, except as noted in the warning that follows. For a detailed description of ciphers, see "Ciphers Used with SSL" in Appendix E of *Managing Servers with iPlanet Console*.

CAUTION You might not want to check the options that say “No Encryption, only MD5 message authentication” and “No Encryption, only Fortezza and SHA message authentication.” The reason for this is, if no other ciphers are available on the client side, the server will use these and no encryption will occur.

Previous US law prohibited the export of software with strong encryption, so most browsers still in use outside of the US and Canada do not support 128-bit encryption. Disabling all 40-bit ciphers will ensure that all connections use higher-grade security, but will prevent access to your service to many users outside of the US and Canada.

Note that Netscape Communicator too 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 Communicator with the same encryption and cryptographic features available in the US and Canada.

Prior to the retail status, international users of Netscape Communicator (with encryption capability restricted to 40-bit encryption) could use Netscape's International Step-Up program to *step up* to stronger encryption, 56-bit, 128-bit, or 168-bit. Step-up refers to the ability of export browsers to establish strong SSL sessions with domestic SSL servers, if they have the appropriate step-up certificates.

Because many of the features, such as issuance of dual certificates for dual key pairs and real-time verification of certificates using the OCSP protocol, supported in Certificate Management System require Communicator versions 4.7x or Netscape 6, it's recommended that you upgrade your browser. For information on downloading the browser, check this site:

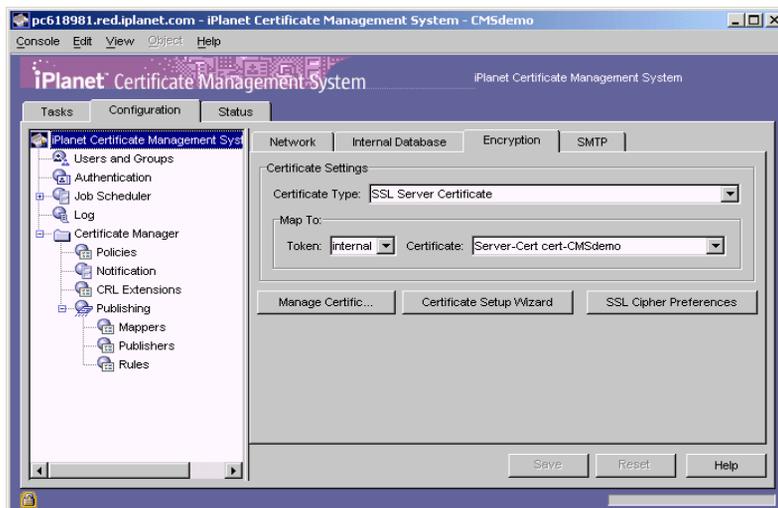
<http://home.netscape.com/download/>

Configuring the Server to Use Specific Ciphers

You can set a number of systemwide preferences for SSL by specifying the ciphers that Certificate Management System should recognize and use during SSL communication; the server applies the cipher settings you choose to all the SSL (HTTPS) ports it uses.

To change the cipher settings for a CMS instance:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab, and then in the right pane, select the Encryption tab.



3. Click SSL Cipher Preferences, and choose the appropriate options.

For details, see “Setting Up Cipher Preferences for SSL Communications” on page 504.

4. Click OK.

You are returned to the Encryption tab.

5. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Getting New Certificates for the Subsystems

You may need to get new certificates for the CMS managers installed in a CMS instance. Getting a new certificate means getting a certificate based on a new public and private key pair.

The sections that follow explain how to get new certificates for the Certificate Manager, Registration Manager, Data Recovery Manager, and Online Certificate Status Manager using the Certificate Setup Wizard. Alternatively, you can use the command-line utilities called the Key Database Tool and Certificate Database Tool. For details about these tools, check the *CMS Command-Line Tools Guide*. To locate an online version of this book, see “Where to Go for Related Information” on page 29.

Getting a new key pair and a corresponding certificate involves the following steps:

- Step 1. Plan for the New Certificate
- Step 2. Request the New Certificate
- Step 3. Install the New Certificate
- Step 4. Deploy the New Certificate

Step 1. Plan for the New Certificate

Getting a new certificate for a CMS manager requires careful planning. This section provides some guidelines that will help you request and install the new certificate.

Determine which certificate you want to get

You can get CA signing, OCSP signing, CRL signing, SSL server, and remote administration certificates for the Certificate Manager; signing, SSL server, and remote administration certificates for the Registration Manager; transport, SSL server, and remote administration certificates for the Data Recovery Manager; and signing, SSL server, and remote administration certificates for the Online Certificate Status Manager. For details about the certificates used by a CMS manager, see “Keys and Certificates for the Main Subsystems” on page 450.

- If you have deployed a Certificate Manager as your root CA and if you want to get a new self-signed CA certificate for that Certificate Manager, you must consider the possible effects on your PKI setup of changing the key pair of the root CA. If you reissue the Certificate Manager’s CA signing certificate with a new key material, none of the certificates issued or signed by the CA using its old key will work; the reason for this is, when you change the root CA key, all certificates that rely on the CA certificate for validation will no longer be

validated. For example, if the CA has issued certificates to subordinate Certificate Managers, Registration Managers, Data Recovery Managers, Online Certificate Status Manager, and agents, all those certificates will become invalid—the subsystems will fail to function, and agents will fail to access agent interfaces.

Before getting a new self-signed certificate for the Certificate Manager, therefore, you must address issues involved in deploying the new root CA certificate across your enterprise. It is beyond the scope of this document to explain how you should deploy the new CA certificate.

- If you have deployed a Certificate Manager as a subordinate CA (that's chained to a root CA) and if you want to get a new subordinate CA certificate for that Certificate Manager, you must consider the possible effects on your PKI setup of changing the key pair of the subordinate CA. When you change the subordinate CA key, all certificates that rely on the subordinate CA certificate for validation will no longer be validated. Before getting a new subordinate certificate, therefore, you must plan to address issues involved in deploying the new subordinate CA certificate across your enterprise.
- If you have deployed a Certificate Manager and if you have configured it to publish CRLs to a Online Certificate Status Manager, you will need to identify the Certificate Manager to the Online Certificate Status Manager again. For details, see “Step 3. Identify the CA to the OCSP Responder” on page 735.
- If you want to get a new signing certificate for a Registration Manager, check whether the Registration Manager has been set up as a trusted manager for a Certificate Manager and Data Recovery Manager—that is, you must identify the subsystems that have been configured to receive requests from this Registration Manager; see “Trusted Managers” on page 405. You will need to replace the existing signing certificate with the new one in all these subsystems.
- If you want to get a new transport certificate for a Data Recovery Manager, you must identify the end-entity interfaces or forms that have been set up for the archival of end users' encryption private keys; see “How Key Archival Works” on page 763. You will need to replace the existing transport certificate with the new one in all these forms.
- If you want to get a new SSL server certificate for a Certificate Manager, determine whether the Certificate Manager is used as a master CA in a cloned-CA setup; see “Cloning a Certificate Manager” on page 288. If it is, you'll have to update the clone CAs certificate databases with the new SSL server certificate.

Also determine whether the Certificate Manager is configured to publish certificates and CRLs to an LDAP directory and whether it uses the SSL server certificate for SSL client authentication to the directory. If it does, you will have to request the certificate with the appropriate extensions, and after installing the certificate you will have to configure the publishing directory to use this certificate.

- You can get any number of SSL server certificates.

Decide on the CA that will sign the certificate

If you want to get a new self-signed CA certificate, you don't have to make this decision, because the CA itself signs it. For all other certificates, you must decide on the CA that will sign the certificate.

If you want the certificate to be signed by an internally deployed CA, check to be sure (for example, the policy configuration) that the CA can issue the certificate you want request.

If you want the certificate to be signed by a public CA, find out the following:

- Does the public CA have a public policy statement? If one is available, read it; it may help you decide whether to request the certificate from this CA.
- Is the public CA's certificate already installed in the trusted CA in the trust database of Certificate Management System? If not, do you want to install it?
- Is the public CA a trusted CA in the trust database of Certificate Management System? If not, do you want to trust it?
- Can the public CA issue the certificate you want to request?
- Does the public CA impose any restrictions on certificates it issues? For example, if you are planning for requesting a subordinate CA certificate for a Certificate Manager, you may want to find out whether the public CA imposes any restrictions on the validity period, volume, or type of certificates your CA can issue. If you are planning for requesting a signing certificate for a Registration Manager, you may want to find out whether the public CA imposes any restrictions on the validity period or the number of certificate requests the Registration Manager can sign using the certificate. If you are planning for requesting a transport certificate for a Data Recovery Manager, you may want to find out whether the public CA imposes any restrictions on the validity period or the number of keys the Data Recovery Manager can archive using the certificate.
- What information does the public CA expects you to provide with the certificate request?

- How long will the public CA take to deliver the certificate, and how will the certificate be delivered to you? (The most common delivery mechanism is by email.)

Determine the token for generating the key pair

Identify the token, internal or external, that you want to use to generate the key pair for the certificate and to store the certificate. For details, see “Tokens for Storing CMS Keys and Certificates” on page 464. If you want to use an existing token, you must know the password that protects the token. If the token is external, make sure that the token is installed properly; for instructions, see “Installing Level 2 External Tokens” on page 466.

Determine certificate formulation information

Decide on the subject DN and nickname for the certificate you want generate. Also decide on details such as the key algorithm, key size, extensions, and validity period for the certificate.

Step 2. Request the New Certificate

Once you have all the information, go ahead and request the certificate. The Certificate Setup Wizard built into the CMS window automates the process of requesting certificates used by the CMS managers. You can use this wizard to generate a new certificate request and submit the request to any CA for signing. For instructions, see “Using the Wizard to Request a Certificate” on page 479.

Step 3. Install the New Certificate

When you receive the certificate from the CA, you must install it in the token that contains the key pair for this certificate; it must be the token you used to generate the request in Step 2 above.

The Certificate Setup Wizard automates the process of installing certificates used by the CMS managers. You can use this wizard to install the new certificate. For instructions, see “Using the Wizard to Install a Certificate or Certificate Chain” on page 493.

Step 4. Deploy the New Certificate

In this step, follow the instructions appropriate for the certificate you installed:

- If you installed a new CA signing certificate for a Certificate Manager, see “Deploying Certificate Manager’s CA Signing Certificate” on page 511.
- If you installed a new signing certificate for a Registration Manager, see “Deploying Registration Manager’s Signing Certificate” on page 512.
- If you installed a new transport certificate for a Data Recovery Manager, see “Deploying Data Recovery Manager’s Transport Certificate” on page 513.
- If you installed a new SSL server certificate, see “Deploying a Subsystem’s SSL Server Certificate” on page 514.

Deploying Certificate Manager’s CA Signing Certificate

If you reissued the Certificate Manager’s CA signing certificate with a new key material, none of the certificates issued by the CA using its old key will work. For example, if the CA has issued certificates to subordinate Certificate Managers, Registration Managers, Data Recovery Managers, Online Certificate Status Manager, and agents, all those certificates will become invalid—the subsystems will fail to function and agents will fail to access the agent interfaces.

To reinstate your PKI setup, first you should get an agent certificate from the new CA so that you can get access to the Certificate Manager’s agent interface. Once you have access to this interface, you will be able to approve new certificate requests from entities such as Registration Managers, Data Recovery Managers, Online Certificate Status Managers, and agents.

To request an agent certificate from the new CA:

1. Go to this directory: `<server_root>/cert-<instance_id>/config`
2. Open the configuration file, `CMS.cfg`, in a text editor.
3. Locate the `agentGateway.enableAdminEnroll` parameter and change its value from `false` to `true`. The modified parameter should look like this:


```
agentGateway.enableAdminEnroll=true
```
4. Save your changes and close the file.
5. Restart the server.
6. Open a web browser window.

7. Go to the Certificate Manager's agent interface.

The URL is in this format: `https://<hostname>:<agent_port>`

8. Enter all the information and request a new certificate.

If you need more information on getting the first agent certificate, see "Stage 3. Enrolling for Administrator/Agent Certificate" on page 277.

9. Once you get the certificate, install it in your browser.
10. Access the Certificate Manager's agent interface using your new certificate.

Deploying Registration Manager's Signing Certificate

If you installed a new Registration Manager signing certificate, you must also install this certificate in the certificate database of all subsystems (Certificate Manager, Registration Manager, and Data Recovery Manager) that trust this Registration Manager.

Here's what you must do:

1. Install the new signing certificate in the subsystems' certificate databases.

Because the Registration Manager uses its signing certificate for SSL client authentication to the subsystems, you must add the new signing certificate to the internal database of all subsystems that have been configured to receive requests from the Registration Manager.

To add the new certificate to a subsystem's internal database:

- o Note the instance ID and host name of the Registration Manager for which you got the new signing certificate; this information will help you to identify the Registration Manager in a subsystem's list of privileged users.
- o Copy the new signing certificate, in base-64 encoded format, to a text file.
- o Add the new certificate to the individual subsystem's internal database following the instructions in "Changing a Privileged User's Certificate" on page 445. Repeat this step for all subsystems that receive requests from this Registration Manager.

2. Ensure that the CA that signed the Registration Manager's certificate is in the certificate database of the subsystem.

When a Registration Manager does SSL client authentication using its new certificate, the subsystem, as a part of validating the certificate presented by the Registration Manager, checks its trust database for the CA (certificate) that signed the Registration Manager's new certificate. If the subsystem does not find the CA as a trusted CA in its trust database, it rejects the Registration Manager.

For instructions on checking the trust database of a subsystem, see "Viewing the Certificate Database Content" on page 523.

- If you don't find the CA certificate, add it to the database as a trusted CA. For instructions on adding a CA certificate to the trust database of a subsystem, see "Installing a New CA Certificate in the Certificate Database" on page 528.
- If you find the CA certificate, verify its trust status. If it is untrusted, change the status to trusted. For instructions on changing the trust setting of a CA certificate, see "Changing the Trust Settings of a CA Certificate" on page 526.

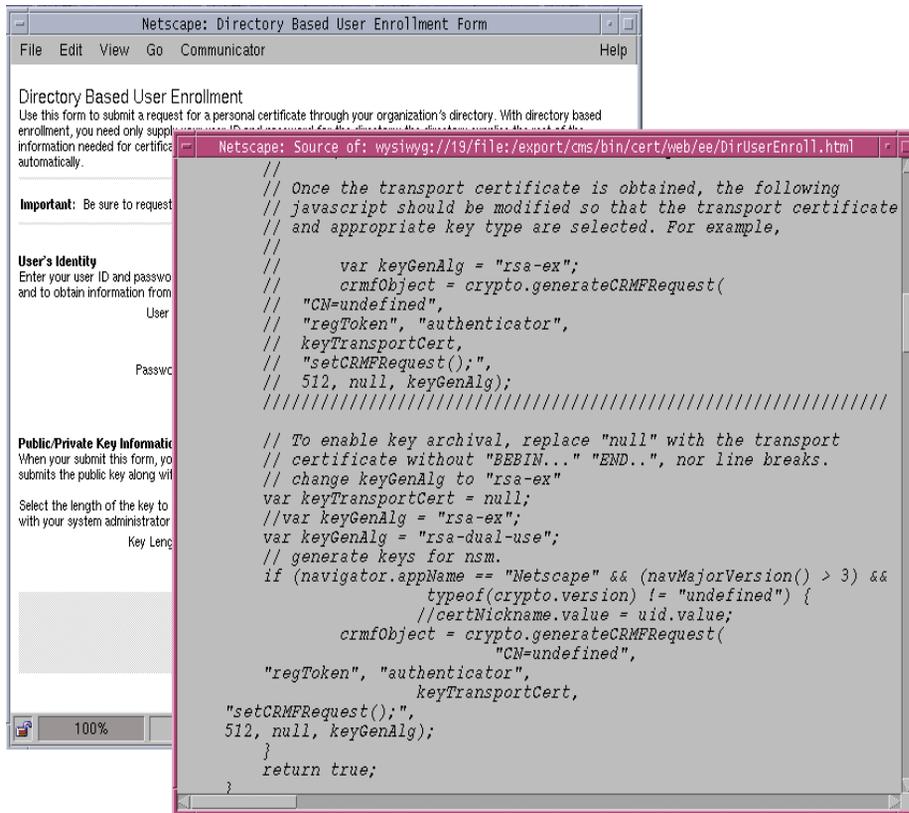
Deploying Data Recovery Manager's Transport Certificate

Because clients capable of generating dual key pairs use the transport certificate for encrypting end users' encryption private keys before sending them to the Data Recovery Manager, you must update the appropriate enrollment or key archival page to identify and use the new transport certificate. Otherwise, the Data Recovery Manager will fail to archive users' encryption private keys.

In general, here's what you need to do:

1. Locate the enrollment page that embeds the key archival feature.
2. View the HTML source, and identify the parameter that corresponds to the Data Recovery Manager's transport certificate.

The default enrollment forms for end users embed this feature. Figure 14-5 shows the default directory-based user enrollment form with the transport certificate-related information. (For more information, see "Step C. Customize the Certificate Enrollment Form" on page 777.)

Figure 14-5 Data Recovery Manager's transport certificate in the enrollment form

3. Replace the current MIME-64 string with the one for the new transport certificate.

To copy the MIME-64 string for the new transport certificate, locate the new transport certificate; the MIME-64 string for the certificate will be listed there.

4. Repeat steps 1 through 3 for any additional enrollment or key-archival pages.

Deploying a Subsystem's SSL Server Certificate

By default, the Certificate Manager and Registration Manager use a single SSL server certificate to do server-side authentication to all the CMS ports. If a Certificate Manager is configured for SSL-client-authenticated communication with the publishing directory, it also uses its SSL server certificate for authenticating to the publishing directory. Depending on the purpose for which you requested this certificate, you should configure the server appropriately.

- To configure the server to use this certificate for authenticating to one of the clients, see “Configuring the Server to Use Separate SSL Server Certificates” on page 500.
- To configure the Certificate Manager to use this certificate for authenticating to the publishing directory, see “Step 5. Identify the Publishing Directory” on page 680.
- If you’re using the Certificate Manager as a master CA, add the new SSL server certificate to the certificate databases of cloned Certificate Managers.

Renewing Certificates for the Subsystems

All certificates used by Certificate Management System have a validity period beyond which they are not usable, unless the validity period is extended. For Certificate Management System to function properly, you must renew the certificates used by the Certificate Manager, Registration Manager, Data Recovery Manager, or Online Certificate Status Manager before they expire. For example, if you generated these certificates during CMS installation with a validity period of two years, at the end of which they will all expire; you must consider renewing them well in advance, maybe two months in advance.

When you *renew* a certificate, you get a new certificate with the same subject name and public and private key material as that of the existing certificate, but with an extended validity period.

The sections that follow explain how to renew certificates for the Certificate Manager, Registration Manager, Data Recovery Manager, and Online Certificate Status Manager using the Certificate Setup Wizard. Alternatively, you use the command-line utility called the *Certificate Database Tool*, which is explained in *CMS Command-Line Tools Guide*.

Renewing an existing certificate involves the following:

- Step 1. Plan for Certificate Renewal
- Step 2. Renew the Existing Certificate
- Step 3. Install the Renewed Certificate
- Step 4. Deploy the Renewed Certificate
- Step 5. Restart the Server

Step 1. Plan for Certificate Renewal

Renewing a CMS manager's certificate requires careful planning. This section provides some guidelines that will help you renew the certificate smoothly.

Before renewing a certificate:

- Note the subject DN and nickname of the certificate you want to renew.

If you are planning on renewing the CA signing certificate of a Certificate Manager, make sure that the Certificate Manager has updated your LDAP directory, file, and OCSP responder with the most current certificate and CRL information. For details, see Chapter 19, Chapter 20, and, Chapter 21.

When you renew its CA signing certificate, the Certificate Manager automatically formulates a new certificate with the same public key and other details from the existing certificate, and publishes the new CA certificate to the configured LDAP directory.

- Identify the token, internal or external, that contains the keys for the certificate you want to renew. To use an existing token, you must know the password that protects the token. If the token is external, make sure that the token is installed properly; see "Installing Level 2 External Tokens" on page 466.
- Decide on the validity period of the renewed certificate.
- Decide on the CA that will sign the certificate. If you want the certificate to be signed by a public CA, find out what information you need to provide with the certificate request. If you want the certificate to be signed by an internally deployed CA, check to be sure it can issue the certificate you want to request and that it's configured to set the required extensions in the certificate.
- Find out how long the CA will take to deliver the certificate to you. Make sure the renewed certificate is delivered to you well in advance so that you have a buffer period for installing and testing the renewed certificate, before the current certificate expires.
- Find out how the certificate will be delivered to you; the most common delivery mechanism is email. Make appropriate arrangements to receive the certificate.
- If you want to renew a subordinate CA certificate, plan how you will deploy the renewed CA certificate to end entities that rely on this certificate for validation.
- If you want to renew a root CA certificate, plan how you will deploy the renewed root CA certificate in your enterprise.

Step 2. Renew the Existing Certificate

Once you have all the information, go ahead and renew the certificate. The Certificate Setup Wizard built into the CMS window automates the process of renewing certificates used by the CMS managers. The wizard can generate a certificate request based on the existing key pair and submit the request to a CA for signing. For instructions on using the wizard, see “Using the Wizard to Request a Certificate” on page 479.

NOTE When renewing a certificate, be sure to use the existing key pair to generate the certificate signing request.

Note that when you renew any of the CMS certificates using the wizard, it saves the old or previous certificate (in its base-64 encoded format) to a text file in the configuration directory, which is located here:

```
<server_root>/cert-<instance_id>/config
```

The names of the text files vary depending on the certificate you choose for renewal. Table 14-2 lists them.

Table 14-2 Names of backup files created for old CMS certificates

Filename	Renewed Certificate
prevCACert.txt.<timestamp>	Certificate Manager CA signing certificate
prevOCSPCert.txt.<timestamp>	Certificate Manager OCSP signing certificate
prevRACert.txt.<timestamp>	Registration Manager signing certificate
prevKRACert.txt.<timestamp>	Data Recovery Manager transport certificate
prevOCSPCert.txt.<timestamp>	Online Certificate Status Manager signing certificate
prevSSLCert.txt.<timestamp>	SSL server certificate
prevSSLRadmCert.txt.<timestamp>	Remote administration server certificate
prevOtherCert.txt.<timestamp>	Other certificates, such as Certificate Manager CRL signing certificate or SSL client certificate

Table 14-2 Names of backup files created for old CMS certificates

Filename	Renewed Certificate
prevClientCert.txt.<timestamp>	SSL client

The wizard also deletes the old certificate from the server's certificate database and adds the renewed certificate to the database, so that the server is able to use the renewed certificate upon restart. This feature restricts you to set the value of the `notBefore` attribute of the renewed certificate to either the current time or any time in the past, but not in the future.

If you set the validity period of the renewed certificate to begin on a future date and time, the server fails to use the certificate for its intended purposes. If this happens, you may either reinstall the old certificate (saved to the text file mentioned above) or renew the certificate again with an appropriate validity period.

Step 3. Install the Renewed Certificate

When you receive the renewed certificate from the CA, you must install it in the token that contains the key pair for the certificate; this is the token you used to generate the request in Step 2.

The Certificate Setup Wizard automates the process of installing certificates used by the CMS managers. For instructions on using the wizard, see "Using the Wizard to Install a Certificate or Certificate Chain" on page 493.

Step 4. Deploy the Renewed Certificate

Follow the instructions appropriate for the certificate you installed:

- If you installed a renewed CA signing certificate for a Certificate Manager, see section "Deploying Certificate Manager's Renewed CA Signing Certificate" on page 519.
- If you installed a renewed signing certificate for a Registration Manager, see section "Deploying Registration Manager's Renewed Signing Certificate" on page 519.

- If you installed a renewed transport certificate for a Data Recovery Manager, see section “Deploying Data Recovery Manager’s Renewed Transport Certificate” on page 520.
- If you installed a renewed SSL server certificate, see section “Deploying a Subsystem’s Renewed SSL Server Certificate” on page 522.

For all certificates, make sure that the CA-chain verification takes place smoothly. For example, if you requested the certificate from a different CA, be sure to import a CA certificate into the certificate database of the subsystem using the Certificate Setup Wizard. For instructions, see “Using the Wizard to Install a Certificate or Certificate Chain” on page 493. After you install the CA certificate, you can follow the instructions in see “Changing the Trust Settings of a CA Certificate” on page 526 to trust the CA certificate you imported.

Deploying Certificate Manager’s Renewed CA Signing Certificate

If you renewed a CA signing certificate, deploy it in the PKI environment that depends on this certificate for validation. For example, you’ll need to add the renewed CA certificate to the certificate databases of clients that trust this CA. Similarly, if you have configured the Certificate Manager to publish CRLs to a Online Certificate Status Manager, you will need to identify the Certificate Manager to the Online Certificate Status Manager again. For details, see “Step 3. Identify the CA to the OCSP Responder” on page 735.

You might also need to get a new agent certificate. For instructions, see the procedure outlined in “Deploying Certificate Manager’s CA Signing Certificate” on page 511.

It is beyond the scope of this book to explain how you should deploy the new CA certificate.

Deploying Registration Manager’s Renewed Signing Certificate

Here’s what you must do:

1. Install the renewed signing certificate in the subsystem’s certificate database.

Because the Registration Manager uses its signing certificate for SSL client authentication to the subsystems, you must add the renewed signing certificate to the internal database of all subsystems that have been configured to receive requests from the Registration Manager.

To add the renewed certificate to a subsystem's internal database:

- a. Note the instance ID and host name of the Registration Manager for which you got the signing certificate; this information will help you to identify the Registration Manager in a subsystem's list of privileged users.
 - b. Copy the renewed signing certificate, in its base-64 encoded format, to a text file.
 - c. Add the renewed certificate to the individual subsystem's internal database following the instructions in "Changing a Privileged User's Certificate" on page 445. Repeat this step for all subsystems that receive requests from this Registration Manager.
2. Ensure that the CA that signed the Registration Manager's certificate is in the trust database of the subsystem.

When a Registration Manager does SSL client authentication using its renewed certificate, the subsystem, as a part of validating the certificate presented by the Registration Manager, checks its trust database for the CA (certificate) that signed the Registration Manager's renewed certificate. If the subsystem does not find the CA as a trusted CA in its trust database, it rejects the Registration Manager.

For instructions on checking the trust database of a subsystem, see "Viewing the Certificate Database Content" on page 523.

- o If you don't find the CA certificate, add it to the database as a trusted CA. For instructions on adding a CA certificate to the trust database of a subsystem, see "Installing a New CA Certificate in the Certificate Database" on page 528.
- o If you find the CA certificate, verify its trust status. If it is untrusted, change the status to trusted. For instructions on changing the trust setting of a CA certificate, see "Changing the Trust Settings of a CA Certificate" on page 526.

Deploying Data Recovery Manager's Renewed Transport Certificate

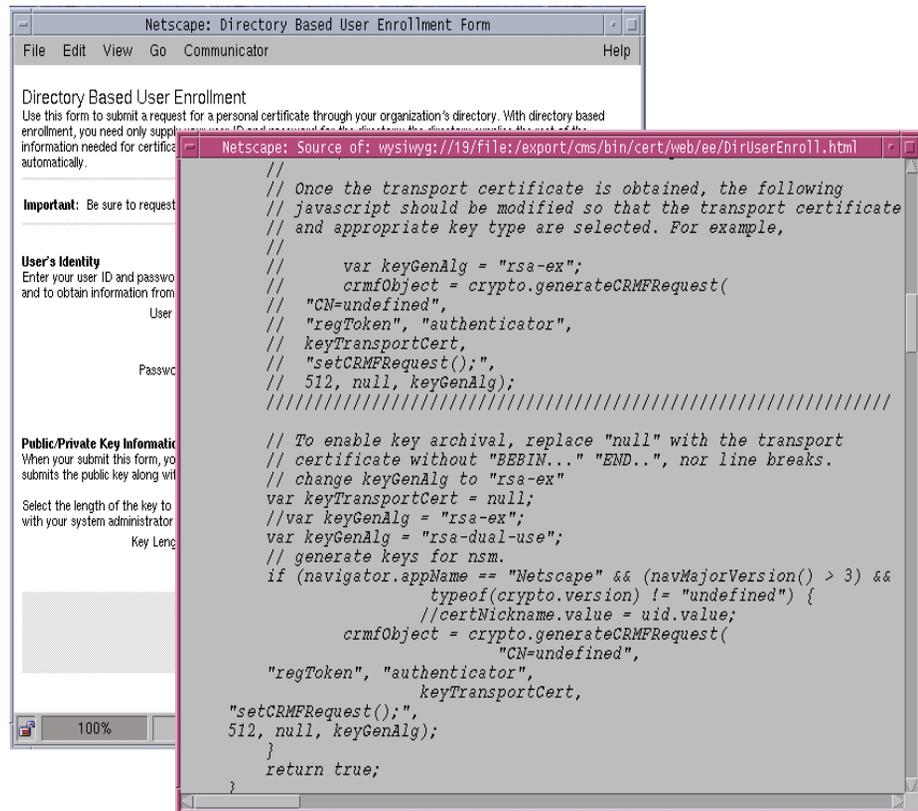
Because clients capable of generating dual key pairs use the transport certificate for encrypting end users' encryption private keys before sending them to the Data Recovery Manager, you must update the appropriate enrollment or key archival page to identify and use the renewed transport certificate. Otherwise, the Data Recovery Manager will fail to archive users' encryption private keys.

In general, here's what you need to do:

1. Locate the page that embeds the key archival feature.
2. View the HTML source, and identify the parameter that corresponds to the Data Recovery Manager's transport certificate.

The default enrollment forms for end users embed this feature. Figure 14-6 shows the default directory-based user enrollment form with the transport certificate-related information. (For more information, see "Step C. Customize the Certificate Enrollment Form" on page 777.)

Figure 14-6 Data Recovery Manager's transport certificate in the enrollment form



3. Replace the current MIME-64 string with the one for the renewed transport certificate.

To copy the MIME-64 string for the renewed transport certificate, locate the certificate; the MIME-64 string for the certificate will be listed there.

4. Repeat steps 1 through 3 for any additional key archival or enrollment pages.

Deploying a Subsystem's Renewed SSL Server Certificate

If you renewed the SSL server certificate of a Certificate Manager and if the Certificate Manager is used as a master CA in a cloned-CA setup (see “Cloning a Certificate Manager” on page 288), you should add the renewed SSL server certificate to the certificate databases of the clone Certificate Managers.

By default, the Certificate Manager and Registration Manager use a single SSL server certificate to do server-side authentication to all the CMS ports. If a Certificate Manager is configured for SSL client authenticated communication with the publishing directory, it also uses the SSL server certificate for authenticating to the publishing directory. The Certificate Manager, if configured to function as a trusted manager to a Data Recovery Manager, also uses its SSL server certificate for SSL client authentication to the Data Recovery Manager. Depending on the purpose for which the certificate being renewed is used currently, you should configure the server appropriately.

- To configure the server to use this certificate for authenticating to one of the clients, see “Configuring the Server to Use Separate SSL Server Certificates” on page 500.
- To configure the Certificate Manager to use this certificate for authenticating to the publishing directory, see “Step 5. Identify the Publishing Directory” on page 680.

Step 5. Restart the Server

After you renew any of the CMS certificates using the wizard, you must restart the server. For instructions, see “Restarting Certificate Management System” on page 332.

Managing the Certificate Database

Each CMS instance has a certificate database (`cert7.db`), which is maintained in its internal token. This database contains certificates belonging to the subsystems installed in the CMS instance (see “Keys and Certificates for the Main Subsystems” on page 450) and various CA certificates the subsystems use for validating the certificates they receive.

Whether you use an internal token or an external token for generating and storing key pairs, Certificate Management System always maintains its list of trusted and untrusted CA certificates in its internal token.

You may need to add new certificates to the database, remove unwanted certificates from the database, or change the trust settings of CA certificates in the database. This section explains how to view the contents of the certificate database, delete unwanted certificates, and change the trust settings of CA certificates installed in the database using the CMS window. For information on adding certificates to the database, see “Certificate Setup Wizard” on page 478.

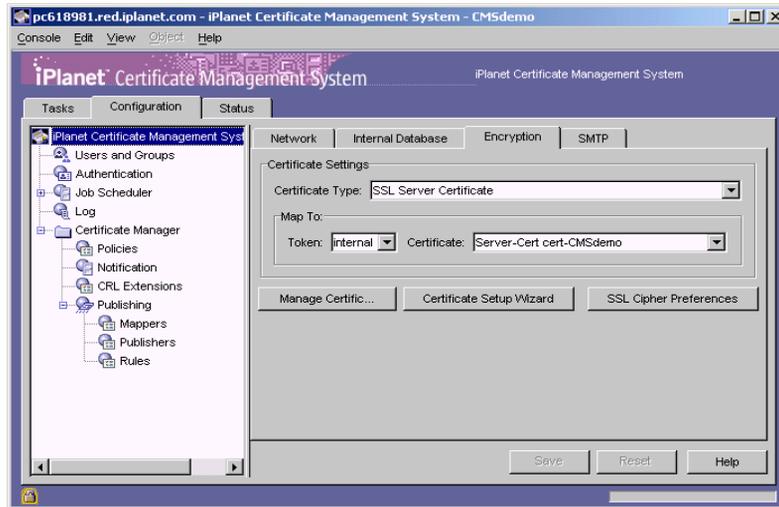
NOTE Certificate Management System also provides a command-line utility called `certutil` for managing its certificate database. For details, check *CMS Command-Line Tools Guide*.

Viewing the Certificate Database Content

Each CMS instance has a certificate database that contains the list of certificates the server uses. To view the contents of the database:

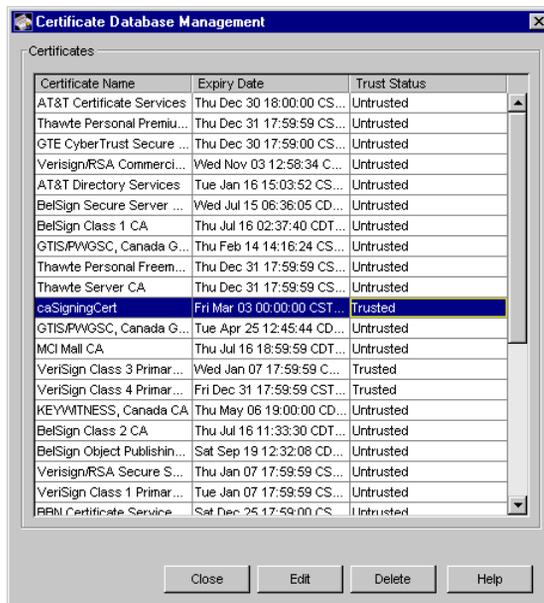
1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).

2. Select the Configuration tab, and then in the right pane, select the Encryption tab.



3. Click Manage Certificate.

The Certificate Database Management window appears.



The window lists the certificates in a table, with each certificate occupying a row. The certificates are listed in alphabetical order. If the database contains multiple certificates with the same nickname, they are sorted by their validity periods; the most recently requested certificate is placed at the top.

For each certificate, you see the following information:

Certificate Name. Specifies the nickname of the certificate.

Expiry Date. Specifies the date (and time) on which the certificate expires.

Trust Status. Specifies whether the CA is trusted or untrusted. To change the trust setting, see “Changing the Trust Settings of a CA Certificate” on page 526.

Deleting a Certificate From the Certificate Database

By default, the CMS certificate database includes a few public or third-party CA certificates. As an administrator, you should periodically check the contents of the certificate database and make sure that it doesn't include any unwanted CA certificates. For example, if the database includes CA certificates that you don't ever want to trust in your PKI setup, you should delete them.

Removing unwanted certificates also reduces the size of the certificate database.

NOTE When deleting CA certificates from the certificate database, be careful not to delete the *intermediate CA certificates*, which help a subsystem chain up to the trusted CA certificate. If in doubt, leave the certificates in the database as *untrusted CA certificates*; see “Changing the Trust Settings of a CA Certificate” on page 526.

To delete a CA certificate from the certificate database:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab, and then in the right pane, select the Encryption tab.

3. Click **Manage Certificate**.

The Certificate Database Management window appears. The window lists all the certificates for the selected instance of Certificate Management System; the list is a table, with each certificate occupying a row.

4. Select the CA certificate you want to delete, and click **Delete**.
5. When prompted, confirm the delete action.
6. Click **Close**.

You are returned to the **Encryption** tab.

7. To save your changes, click **Save**.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Changing the Trust Settings of a CA Certificate

Certificate Management System relies on the CA certificates in its certificate database for validating certificates it receives during an SSL-enabled communication. For example, when a Certificate Manager is authenticating a Registration Manager that has sent a certificate signing request, the Certificate Manager checks its certificate database to see whether the CA that has signed the certificate presented by the Registration Manager is included in the database as a *trusted* CA.

You may need to change the status of a currently trusted CA to untrusted (or vice versa) temporarily or permanently. For example, you may be notified that a CA is experiencing technical difficulty that prevents certificate authentication. By making the CA certificate untrusted, you can prevent entities whose certificates have been signed by that CA from successfully authenticating to Certificate Management System. You can then return the trust option to *trusted* when the CA notifies you that the problem has been resolved.

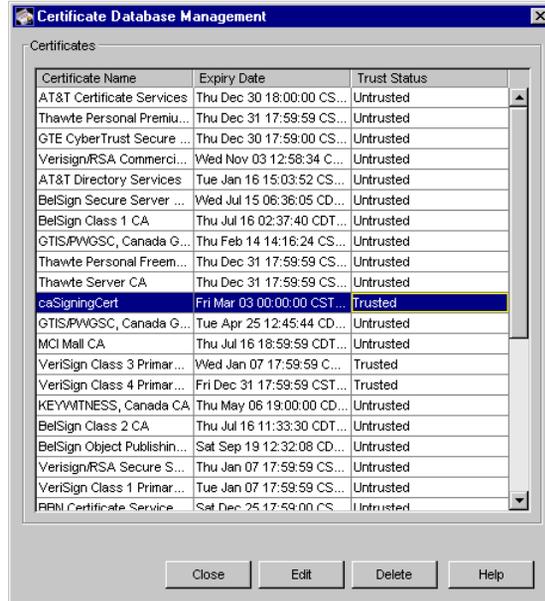
If you want to untrust a CA permanently, you should consider removing its certificate from the trust database altogether. For instructions, see “Deleting a Certificate From the Certificate Database” on page 525.

Changing the trust setting changes the trust flag (or bit) in the CA certificate. To change the trust setting of a CA certificate:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).

2. Select the Configuration tab, and then in the right pane, select the Encryption tab.
3. Click Manage Certificate.

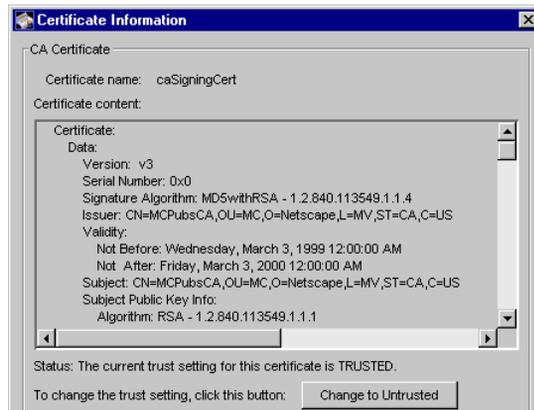
The Certificate Database Management window appears.



The window lists the certificates currently installed for the selected CMS instance; the list is a table, with each certificate occupying a row.

4. Select the CA certificate whose trust setting you want to modify, and click Edit.

The Certificate Information window appears.



The window shows detailed information about the selected certificate, including serial number, validity period, subject name, issuer name, certificate fingerprint, and trust status.

If the certificate you selected is currently trusted, the window shows a button named “Change to Untrusted.” If it is untrusted, the window shows a button named “Change to Trusted.”

5. Click “Change to Untrusted” or “Change to Trusted,” as appropriate.
6. Click Close.

You are returned to the Certificate Database Management window. The certificate now shows a different trust status.

7. Click Close.

You are returned to the Encryption tab.

8. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Installing a New CA Certificate in the Certificate Database

You may need to install new trusted CA certificates in the certificate database of a CMS instance. For example, assume that you renewed the signing certificate of a Registration Manager. Also assume that the CA that signed the Registration Manager’s certificate is not included in the trust database of the Certificate Manager that has been configured to sign certificate requests from this Registration Manager.

When the Registration Manager attempts to request a service from the Certificate Manager (using the renewed certificate for SSL client authentication), the Certificate Manager fails to authenticate the Registration Manager. This happens because, as a part of validating the certificate presented by the Registration Manager, the Certificate Manager checks its certificate database for the CA that signed the Registration Manager’s certificate. The Certificate Manager does not find the CA listed in its trust database as a trusted CA, so it rejects the Registration Manager’s service request.

The Certificate Setup Wizard built into the CMS window automates the process of installing trusted CA certificates in the certificate database. For instructions on using the wizard, see “Using the Wizard to Install a Certificate or Certificate Chain” on page 493.

NOTE Be sure to choose the “Other Trusted CAs” option in Step 2 of the wizard process.

Installing a CA Certificate Chain in the Certificate Database

Any client or server software that supports certificates maintains a collection of trusted CA certificates in its certificate database. These CA certificates determine which other certificates the software can validate—in other words, which issuers of certificates the software can trust. In the simplest case, the software can validate only certificates issued by one of the CAs for which it has a certificate. It’s also possible for a trusted CA certificate to be part of a chain of CA certificates, each issued by the CA above it in a certificate hierarchy; for details on certificate hierarchies and certificate chains, see “How CA Certificates Are Used to Establish Trust” in Appendix D of *Managing Servers with iPlanet Console*.

Setting Up End-User Authentication

iPlanet Certificate Management Server (CMS) provides a customizable authentication component that supports various methods for authenticating end users. This chapter provides an introduction to various parts of Certificate Management System that require authentication and explains how to configure a Certificate Manager and Registration Manager to use specific authentication plug-in modules for authenticating end users during certificate enrollment.

The chapter has the following sections:

- Introduction to Authentication (page 531)
- Configuring Authentication for End-User Enrollment (page 545)
- Managing Authentication Instances (page 569)
- Managing Authentication Plug-in Modules (page 572)

Introduction to Authentication

Authentication is the process of verifying the identity of a user that is requesting a service from iPlanet Certificate Management Server (CMS). More specifically, authentication involves acquiring and verifying the values of the configured attributes of the user. For example, the user might be prompted to log in with a user name and password, and then the server would look in a preconfigured database to verify that the user's password was correct.

Service requests to Certificate Management System come from any of the following users:

- End entities—general users or applications that make certificate issuance, renewal, and revocation requests

- Administrators—privileged users who connect to the server to do system or server administration tasks
- Agents—privileged users who connect to the server to do agent operations

This section explains how Certificate Management System identifies and authenticates these users, and it provides details about the various authentication methods supported by the server.

This section has the following sections:

- Privileged-User Authentication
- End-Entity Authentication

Privileged-User Authentication

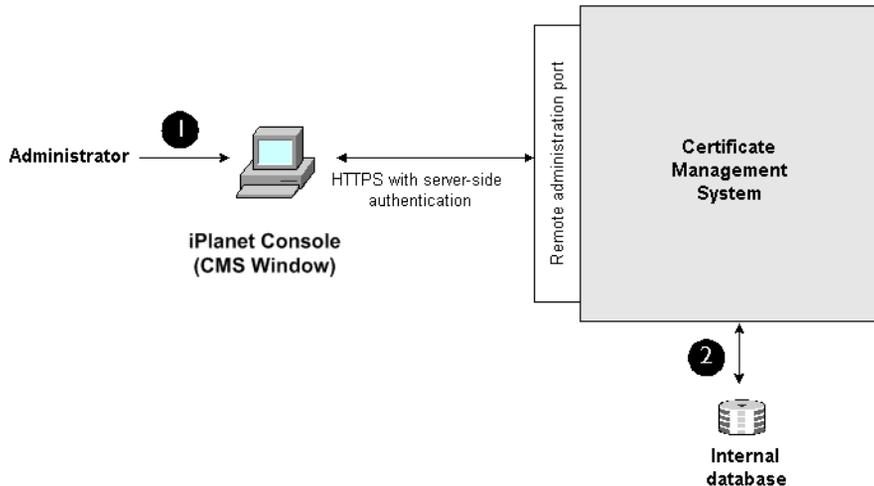
For authenticating privileged users, such as administrators and agents, Certificate Management System uses built-in authentication mechanisms.

Authentication of Administrators

When an administrator makes an administrative request to Certificate Management System (from the CMS window within iPlanet Console or from any command-line tool), the server needs to authenticate the administrator before processing the request. To facilitate this, Certificate Management System supports an authentication method that includes user ID- and password-based authentication from the client and SSL server authentication from the server.

Certificate Management System identifies and authenticates users with *administrator* privileges by checking their user IDs and passwords in its internal database. These are the user IDs and passwords you entered in the internal database when you created these user entries. For details, see “Setting Up Administrators” on page 413.

Figure 15-1 illustrates the authentication process.

Figure 15-1 CMS authentication of a user with *administrator* privileges

These are the steps shown in Figure 15-1:

1. An administrator opens iPlanet Console and attempts to log in to the CMS window by entering the user ID and password at the login prompt. The server takes the administrator's user ID and password and binds them to privileged-user entries in its internal database.
2. If the user ID and password bind successfully to a user entry, authentication succeeds; otherwise, it fails.
 - If authentication fails, the server logs an error message and sends a rejection notification. See Chapter , “.”
 - If authentication succeeds, the server checks the user's access rights (based on group memberships) to determine whether the user is authorized to perform the requested operation.

If both authentication and authorization succeed, the server services the request. Otherwise, it rejects the request and logs the reason for the rejection.

NOTE Authentication for administrators is hardcoded; it is not configurable.

Authentication of Agents

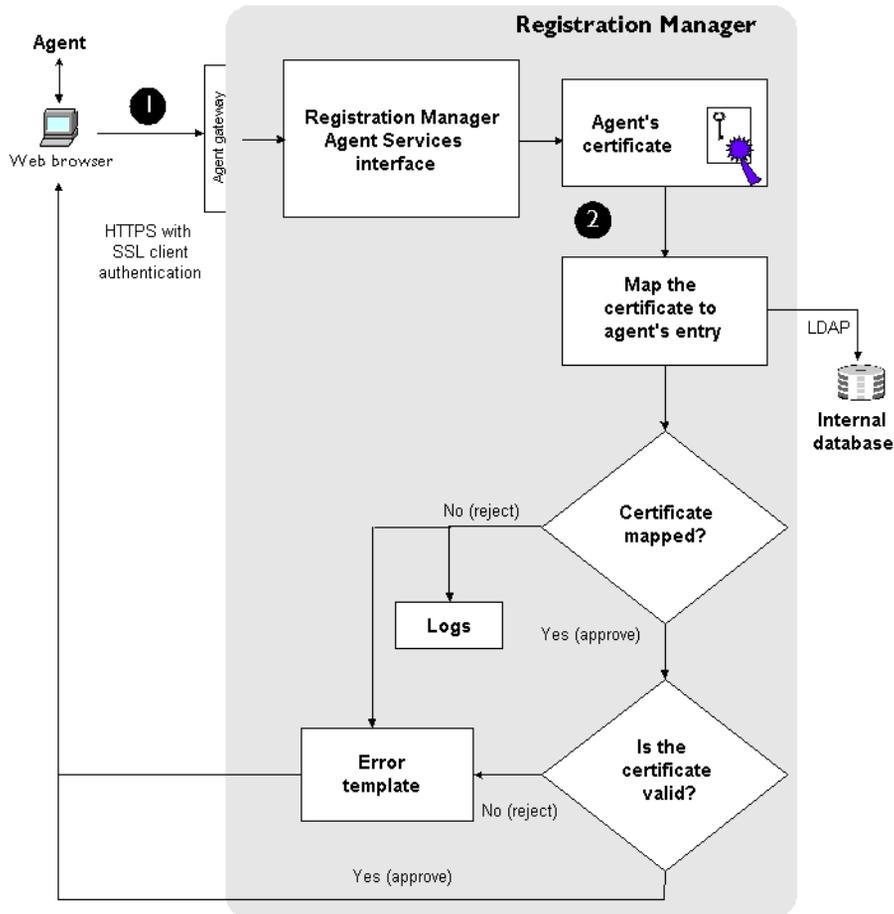
When an agent makes a request to Certificate Management System (from the appropriate Agent Services interface), the server needs to authenticate the agent before processing the request. To facilitate this, Certificate Management System supports a certificate-based authentication method.

Certificate Management System identifies and authenticates a user with *agent* privileges by checking the user's SSL client certificate in its internal database. The certificates it checks are the ones you imported and stored in the internal database while creating or modifying the user entry. You create agent users for a CMS instance by adding their client certificates into the internal database and associating them with the corresponding users' identification information; for details, see "Setting Up Agents" on page 416.

When an agent makes a request to perform a privileged operation, the server requests SSL client authentication from the client that the agent has used to connect to the server. The server then uses the successfully SSL client-authenticated certificate to map to internal user entries for further checks. The server checks the certificate's subject name and issuer name against the list of privileged-user certificates stored in its internal database. If the certificate belongs to a privileged user who is authorized (based on group membership) to perform agent operations, the server allows the user to perform the requested operation. Otherwise, the server rejects the request and logs an appropriate message; for details, see , "Managing CMS Logs."

NOTE Authentication for agents is hardcoded; it is not configurable.

Figure 15-2 shows how a Registration Manager authenticates and authorizes a Registration Manager agent.

Figure 15-2 Registration Manager authentication of a user with Registration Manager agent privileges

This example shows these steps:

1. An agent opens a web browser and enters the URL to the Registration Manager Agent Services interface hosted by the Registration Manager. The server requests the client for SSL client authentication. The client in turn prompts the agent to specify the certificate that it should present to the server for authentication. The successfully SSL client authenticated certificate is presented to the Registration Manager.

2. Upon receiving the certificate, the Registration Manager performs the following authentication and authorization process:
 - o First, it verifies that the certificate exists in its internal database. Next, it verifies that the certificate is a valid client certificate. If the certificate is valid, the Registration Manager proceeds. Otherwise (for example, if the certificate has expired or been revoked or was signed by an *untrusted* authority), the Registration Manager rejects the request, sends an error message to the agent, and logs a reason for the rejection.

Note that the Registration Manager verifies the revocation status of the agent certificate if it has been issued by the Certificate Manager to which the Registration Manager is connected to; the Certificate Manager keeps a record of all the certificates it has issued and their current status in its internal database. However, if the agent certificate is issued by any other CA, the Registration Manager cannot verify the revocation status of the certificate; it can only verify that the certificate is valid and that it has been issued by a CA that the Registration Manager trusts. For details on configuring the Certificate Manager or Registration Manager to check the revocation status of its agents' certificates, see "Revocation Status Checking of Agent Certificates" on page 402.

If the internal database contains an invalid certificate for an agent, the server rejects all requests from that agent. For the server to accept requests from that agent, you would have to replace the agent's invalid certificate in the internal database with a valid one. For details on how to do this, see "Changing a Privileged User's Certificate" on page 445.

- o The Registration Manager reads the user's subject name (in DN form) and the issuer name from the certificate. This combination is unique. It then finds the login name corresponding to this unique combination in its privileged-users list, which is stored in the internal database. If a login name is associated with the certificate, the Registration Manager proceeds. Otherwise, it rejects the request.

The Registration Manager then checks the group memberships of the login name and the corresponding access rights to determine whether the user is authorized to perform the requested service.

If both authentication and authorization succeed, the Registration Manager services the request. Otherwise, it rejects the request and logs a reason for the rejection.

End-Entity Authentication

This section provides an overview of how Certificate Management System authenticates end entities during certificate enrollment, renewal, and revocation processes.

Authentication of End Entities During Certificate Enrollment

When an end entity submits a certificate request, a Certificate Manager or Registration Manager's first task is to identify and authenticate the end entity. The server must perform this task before it can register the end entity for certificate issuance. This task includes verifying the end entity's identity based on information the end entity provides and returning enough information about the end entity so that the subject name for the certificate can be constructed.

To cater to a variety of end-entity enrollment scenarios, Certificate Management System supports both manual and automated certificate issuance. For detailed description of authentication methods supported by the Certificate Manager and Registration Manager, see Chapter 1, "Authentication Plug-in Modules" of *CMS Plug-Ins Guide*. To locate an online version of this guide, open the `<server_root>/manual/index.html` file.

Authentication of End Users During Certificate Renewal

When an end user submits a certificate renewal request, the first step in the renewal process is for the Certificate Manager or Registration Manager to identify and authenticate the end user. This step includes making sure that the end user's current certificate is either "valid" or "expired" ("revoked" is not acceptable).

Certificate Management System verifies the authenticity of a certificate renewal request by mapping the subject name in the certificate being presented for renewal to certificates in its internal database. The server renews the certificate only if the subject name maps successfully to a certificate in its internal database. If the internal database contains more than one certificate with matching subject name as that the one presented by the end entity for client authentication, the server lists all the matching certificates and expects the end entity to pick one for renewal.

Here are a few things to keep in mind about certificate renewal:

- The certificate being presented by the end user for renewal must be issued by a Certificate Manager.
- If the renewal request is processed by a Registration Manager, the end-user certificate presented must be issued by a Certificate Manager that the Registration Manager knows and is connected to; the Registration Manager forwards certificate requests to this Certificate Manager for signing.

- The certificate being presented by the end user for renewal must be currently valid or must have expired; it cannot have been revoked.
- The validity period of a renewed certificate is determined by the policy rule explained in “RenewalValidityConstraints Plug-in Module” of *CMS Plug-Ins Guide*. If the renewal lead time does not permit renewing, the server rejects the renewal request. Also, if the policy is disabled, renewal of certificates fails.
- If the certificate being presented by the end user has already been renewed, the server displays the URL for downloading the certificate.

This situation may occur if the end user forgets to download the renewed certificate. It can also happen if the end user maintains two identical certificate databases on two machines, renews the certificate from one machine, and then tries to renew the same certificate from the other machine.

Challenge Password-Based Renewal

A challenge password is a unique, alphanumeric string that the end user specifies when requesting a certificate; the user is expected to keep this password confidential and use it to authenticate to the server when renewing the certificate. When the server issues the certificate, it associates the password with the certificate, stores both the certificate and password in its internal database, and uses them later for authenticating any renewal requests.

If a challenge phrase was set during certificate enrollment, a user can renew the certificate even though he may no longer have access to the actual certificate. This may be useful when, for example, the certificate was stored on a disk that failed. Another example is when the certificate is a server or object-signing certificate which cannot be used for SSL client authentication. In order to renew the certificate, the user must either present the certificate to the server (the web browser will do this automatically if the certificate is installed in it) or he must know the secret challenge phrase and the certificate serial number.

The server verifies the authenticity of a renewal request by mapping the serial number to the list of certificates in its internal database followed by mapping the challenge password specified to the one associated with the matching certificate it detects in the internal database.

The server renews the certificate only if the certificate maps successfully to one or more valid or expired certificates in its internal database. If the server detects only one valid or expired certificate with a matching serial number and challenge password, it automatically renews the certificate. If the server detects more than one valid or expired certificates with matching serial numbers, it lists all those certificates. The user can then select the certificate to be renewed, or renew all certificates in the list.

Here are a few things, to keep in mind about the challenge-password-based renewal:

- The certificate being presented by the user for renewal must be issued by a Certificate Manager.
- The user must have requested the certificate using the *manual enrollment method*—only the default manual enrollment form includes fields for entering the challenge password when requesting a certificate.
- The user can renew only those certificates that contain the specified serial number with the corresponding challenge password. For example, if there is a mismatch between the challenge password and serial number, the server rejects the renewal request.

Certificate Renewal Form

The End Entity Services interface of the Certificate Manager and Registration Manager includes a default HTML form for renewing end users' certificates. The form is accessible from the Renewal tab as shown in Figure 15-3.

Figure 15-3 Certificate renewal form for end users

The screenshot shows the iPlanet Certificate Management System interface. At the top, it says "iPlanet Certificate Management System" and "Certificate Manager". Below this is a navigation bar with four tabs: "Enrollment", "Renewal", "Revocation", and "Retrieval". The "Renewal" tab is selected. On the left side, there is a sidebar with "User Certificate" and "Certificate (challenge phrase based)". The main content area is titled "User Certificate Renewal" and contains the following text:

Use this form to renew your certificate automatically.

After you click the Submit button, a window will pop up with a list of certificates you can send to the server. Select the certificate you want to renew from this window.

Important: Be sure to make this request on the same computer on which you plan to use your renewed certificate.

At the bottom right of the form, there are two buttons: "Submit" and "Help".

The default renewal form is preconfigured for SSL client authentication, enabling end users to renew their personal or client certificates by presenting valid or expired certificates.

If you want to change the form content to suit your organization's requirements, edit the following file:

```
<server_root>/cert-<instance_id>/web/ee/UserRenewal.html
```

For details on individual form elements, see the online help available by clicking the Help button on the form. For more information on customizing the form, see *CMS Customization Guide*. To locate an online version of this guide, open the `<server_root>/manual/index.html` file.

Authentication of End Users During Certificate Revocation

Certificates can be revoked by administrators, agents, and end users. When an end user submits a certificate revocation request, the first step in the revocation process is for the Certificate Manager or Registration Manager to identify and authenticate the end user. The reason for this is when an end user attempts to revoke a certificate, the server needs to verify that the user is attempting to revoke his or her own certificate, not a certificate belonging to someone else.

Both Certificate Manager and Registration Manager support the following methods of revocation:

- SSL client authenticated revocation

This method requires an end user to present a valid or revoked certificate that has the same subject name as the one he or she wants to revoke. Without the certificate, the user won't be able to revoke the certificate.

- Challenge-password-based revocation

This method requires an end user to enroll for a personal certificate using the *manual enrollment* method. The reason for this is, by default, only the manual enrollment form includes fields for entering the challenge password when requesting a certificate. None of the other enrollment forms, for example directory-based or NIS server-based forms, by default allow end users to specify a challenge password.

You can use the manual-enrollment form (`ManUserEnroll.html`) as a model and introduce the input fields for entering the challenge password in any of the other end user enrollment forms. Keep in mind that this feature is available for end-user certificates only; the feature is not available for other types of certificates.

Revoking a certificate using the challenge password is useful in certain situations. For example, if you issue a single certificate to a user and the user is unable to use the certificate due to loss of corresponding key pair, it's not possible for the user to revoke his or her own certificate using the SSL client authenticated revocation method. If the user has a challenge password for the certificate, he or she can use it to revoke the certificate the server maintains in its database.

Forms for both methods are available through the End Entity Services interface (HTTPS only) of the Certificate Manager and Registration Manager; see "Certificate Revocation Forms" on page 543.

Here are a few common points to keep in mind about the automated revocation of end users' certificates:

- A Certificate Manager can revoke only those certificates that it has issued; it cannot revoke certificates issued by other CAs.
- If the revocation request is processed by a Registration Manager, it must be connected as a *trusted manager* to the Certificate Manager that has issued the certificate the user is attempting to revoke; the Registration Manager forwards certificate revocation requests to this Certificate Manager. For information on trusted managers, see "Trusted Managers" on page 405.
- The certificate the user attempts to revoke must be currently valid or must have expired; it cannot have been already revoked.
- At the time of revocation, the user can also specify additional details, such as the date of revocation and revocation reason, for each certificate or for the list as a whole.

SSL Client Authenticated Revocation

In an SSL client authenticated revocation method, the server expects the end user to present a certificate that has the same subject name as the one he or she wants to revoke and uses that for authentication purposes. The server verifies the authenticity of a revocation request by mapping the subject name in the certificate being presented for client authentication to certificates in its internal database. The server revokes the certificate only if the certificate maps successfully to one or more valid or expired certificates in its internal database.

After successful authentication, if the server detects only one valid or expired certificate with matching subject name as that of the one presented for client authentication, it revokes the certificate. If the server detects more than one valid or expired certificate with matching subject name, it lists all those certificates. The user can then either select the certificate to be revoked or revoke all certificates in the list.

Here are a few things, in addition to the ones listed on page 541, to keep in mind about SSL client authenticated revocation:

- The certificate being presented by the user for revocation must be issued by a Certificate Manager.
- If the revocation request is processed by a Registration Manager, the certificate presented for SSL client authentication must be issued by a Certificate Manager that the Registration Manager knows about and is connect to (the Registration Manager forwards certificate requests to this Certificate Manager for signing).
- The certificate being presented by the user for revocation must be currently valid or must have expired; it cannot have been already revoked.
- The user can revoke only certificates that contain the same subject name as the one in the certificate presented for authentication.

Challenge-Password-Based Revocation

A challenge password is a unique, alphanumeric string that the end user specifies when requesting a certificate; the user is expected to keep this password confidential and use it to authenticate to the server when revoking the certificate. When the server issues the certificate, it associates the password with the certificate, stores both the certificate and password in its internal database, and uses them later for authenticating any revocation requests.

In the challenge-password-based revocation method, the server expects the end user to specify the serial number of the certificate the user wants to revoke and the challenge password associated with the certificate. The server verifies the authenticity of a revocation request by mapping the serial number to the list of certificates in its internal database followed by mapping the challenge password specified to the one associated with the matching certificate it detects in the internal database.

The server revokes the certificate only if the certificate maps successfully to one or more valid or expired certificates in its internal database. If the server detects only one valid or expired certificate with a matching serial number and challenge password, it automatically revokes the certificate. If the server detects more than one valid or expired certificates with matching serial numbers, it lists all those certificates. The user can then select the certificate to be revoked or revoke all certificates in the list.

Here are a few things, in addition to the ones listed on page 541, to keep in mind about the challenge-password-based revocation:

- The certificate being presented by the user for revocation must be issued by a Certificate Manager.
- The user must have requested the certificate using the *manual enrollment method*—only the default manual enrollment form includes fields for entering the challenge password when requesting a certificate.
- The user can revoke only those certificates that contain the specified serial number with the corresponding challenge password. For example, if there is a mismatch between the challenge password and serial number, the server rejects the revocation request.

Certificate Revocation Forms

The End Entity Services interface of the Certificate Manager and Registration Manager includes default HTML forms for both the SSL client authenticated revocation and challenge-password-based revocation. The forms are accessible from the Revocation tab. Figure 15-4 shows the form that enables end users to revoke their certificates using a challenge password. You can view the form that enables SSL client authenticated revocation by clicking the User Certificate link.

Figure 15-4 Form for SSL client authenticated certificate revocation

iPlanet®
Certificate Management
System

Certificate Manager

Enrollment Renewal **Revocation** Retrieval

User Certificate
Certificate
(challenge
phrase based)

User Certificate Revocation

Use this form to revoke your certificate automatically.

After you click the submit button, a window will pop up with a list of certificates you can send to the server. Select the certificate you want to revoke from this window.

Important: This is an irreversible operation. If you still want to continue, be sure to request revocation on the computer where the private key and certificate to be revoked are stored.

Revocation Reason
Select a revocation reason

- Unspecified
- Key Compromise
- Cessation of Operation
- Affiliation Changed
- Superseded

Submit Reset Help

If you want to change the forms to suit your organization's requirements, you can edit the following files:

- `ChallengeRevoke1.html` (the form that allows challenge password based revocation of client or personal certificates)
- `UserRevocation.html` (the form that allows SSL client authenticated revocation of client or personal certificates)

Both the files are located here: `<server_root>/cert-<instance_id>/web/ee`

For details on individual form elements, see the online help available by clicking the Help button on the form. For more information on customizing the form, see *CMS Customization Guide*.

Configuring Authentication for End-User Enrollment

To set up a Certificate Manager or Registration Manager to authenticate end users based on a specific criteria, follow these steps:

- Step 1. Before You Begin
- Step 2. Set Up the Directory for PIN-Based Enrollment (required for PIN-based enrollment only)
- Step 3. Enable the AttributePresentConstraints Policy (required for PIN-based enrollment only)
- Step 4: Add an Authentication Instance
- Step 5. Set Up the Enrollment Interface
- Step 6. Enable End-Entity Interaction
- Step 7. Turn on Automated Notification
- Step 8. Test Your Authentication Setup
- Step 9. Deliver PINs to End Users (required for PIN-based enrollment only)

NOTE If you do not configure a Certificate Manager or Registration Manager to use any of the registered authentication plug-in modules, the server uses manual authentication for end-user enrollment. This means that all end-user enrollment requests are queued for agent approval. For more information, see section “Manual Authentication” in Chapter 1, “Authentication Plug-in Modules” of *CMS Plug-Ins Guide*.

Step 1. Before You Begin

Before setting up a Certificate Manager or Registration Manager to use a specific authentication method:

- Determine the authentication module you want to use. To find out about the modules that are installed with the server, see Chapter 1, “Authentication Plug-in Modules” of *CMS Plug-Ins Guide*. If you want to develop and use a custom plug-in module, be sure to check the tutorials provided in this directory: `<server_root>/cms_sdk/cms_jdk/samples/authentication`

- If you decided to use the directory-based authentication module, note the authentication directory credentials, such as the host name, port number, base DN, the user entry to bind as and the corresponding password, the DN pattern to retrieve from the directory to construct certificate subject names, LDAP version number, and minimum and maximum number of connections permitted.
- If you decided to use the directory- and PIN-based authentication module, note the authentication directory credentials, such as the host name, port number, based DN, the user entry to bind as and the corresponding password, LDAP version number, and minimum and maximum number of connections permitted.

Next, read Chapter 4 , “PIN Generator Tool” of *CMS Command-Line Tools Guide*. Determine the options you want to use to generate PINs and construct the command for generating the PINs. Note that the `optfile` option enables you to put all the arguments in a file (instead of typing the arguments at the command prompt) and then point the tool to read arguments from the file.

- If you decided to use the NIS server-based authentication module, note the NIS server host name and domain name. If you have an LDAP directory deployed and want to use that for formulating the certificate subject names, note the directory-specific information.
- If you decided to use the portal authentication module, note the LDAP directory-specific information.
- Determine the enrollment form you want your users to use. Decide whether you want to customize it.

The next step depends on the authentication module you chose:

- If you decided to use the directory- and PIN-based authentication module, go to the next step, “Step 2. Set Up the Directory for PIN-Based Enrollment” on page 547.
- For all other modules, skip to “Step 4: Add an Authentication Instance” on page 553.

Step 2. Set Up the Directory for PIN-Based Enrollment

Complete this step only if you want to configure the server to use the directory- and PIN-based authentication method (with or without PIN removal). Otherwise, skip to the next step.

To set up a directory for PIN-based authentication method:

- Step A. Check the Directory for User Entries
- Step B. Update the Directory
- Step C. Prepare the Input File
- Step D. Run the Command Without the Write Option
- Step E. Check the Output File
- Step F. Run the Command Again with the Write Option

Step A. Check the Directory for User Entries

Before you run the PIN Generator tool, make sure that the directory you intend to use for generating PINs has been populated with end-user entries that require PINs. It is also a good idea to confirm with that directory's administrator that all pending directory requests have been processed before the PIN Generator starts its operation.

Step B. Update the Directory

By default, the PIN Generator modifies the `pin` attribute in a directory's user entry. Because this attribute is not part of the standard `organizationalPerson`, it's likely that the user entries in your directory do not contain the `pin` attribute. This means, before you run the PIN Generator, you'll need to add the `pin` attribute to the user entries in your directory—that is, you'll need to create a new object class (named `pinPerson`) in your authentication directory's schema.

In general, you'll need to update the `slapd.user_at.conf` file to include the `pin` attribute and the `slapd.user_oc.conf` file to include the object-class definition. The modified schema should look similar to this:

```
attribute pin bin
objectclass pinPerson
    superior organizationalPerson
    allows
        pin
```

In addition, if you want to make use of the PIN-removal feature—that is, remove a user's PIN from the directory after Certificate Management System successfully authenticates that user and thus prevents the user from enrolling for another certificate—ACIs must be set up on the directory to prevent end users from creating new PINs for themselves. To do this, you'll need to create an entry for a *PIN manager* user with read-write permission to the `pin` attribute.

For your convenience, the PIN Generator tool comes with a configuration file, named `setpin.conf`, which enables you to automate the process of updating the authentication directory with changes required for setting up PIN-based authentication. The configuration file is located in this directory:

```
<server_root>/bin/cert/tools
```

To make the required schema changes and add an entry for the PIN manager user (using the configuration file):

1. Go to this directory: `<server_root>/bin/cert/tools`
2. Open the `setpin.conf` file in a text editor.
3. Follow the instructions outlined in the file and make the appropriate changes.

Typically, you will need to update the Directory Server's host name, Directory Manager's bind password, and PIN manager's password.

4. Run the `setpin` command with its `optfile` option pointing to the `setpin.conf` file (`setpin optfile=setpin.conf`).

The tool modifies the schema with a new attribute (by default, `pin`) and a new object class (by default, `pinPerson`), creates a `pinmanager` user, and sets the ACI to allow only the `pinmanager` user to modify the `pin` attribute.

If the tool is able to update the directory with all the changes, you should see a status-update message, similar to the sample shown below:

```

Adding attribute: ( pin-oid NAME 'pin' DESC 'User Defined
Attribute' SYNTAX '1.3.6.1.4.1.1466.115.121.1.5' SINGLE-VALUE )
.. successful

Adding objectclass: ( pinPerson-oid NAME 'pinPerson' DESC 'User
Defined ObjectClass' SUP 'top' MUST ( objectclass ) MAY ( aci $
pin )
.. successful

Adding user: cn=pinmanager,o=siroe.com
.. successful

modifying ACI for: ou=people,o=siroe.com
.. successful

```

Step C. Prepare the Input File

This step is optional.

If you want to generate PINs for specific user entries or want to provide your own PINs, use an input file (to provide the tool with such information). For information on constructing an input file, check the PIN Generator documentation.

Step D. Run the Command Without the Write Option

Run the `setpin` command without the `write` option. Be sure to include the `output` option and bind to the directory as the PIN manager user.

The tool will write PINs to the specified output file; no changes are made to the directory yet. This will give you the opportunity to check the PINs (by looking at the output file) before updating the directory.

To run the command:

1. Open a terminal window.
2. Go to this directory: `<server_root>/bin/cert/tools`
3. Run the `setpin` command with the appropriate arguments; be sure to point the `outfile` option to the file you've created (and not to the `setpin.conf` file).

Step E. Check the Output File

Check the output file to be sure it contains PINs for your users; the output should look similar to the one specified in PIN Generator documentation.

Next, verify that the tool has assigned PINs to the correct users and that the PINs conform to the length and character-set restrictions you specified. If the output isn't what you want, run the command again with appropriate arguments. Repeat the process until the output file shows the results you want.

Step F. Run the Command Again with the Write Option

When you are sure about the results, run the command again (with exactly the same arguments) with the `write` option and the `output` option. The tool stores the hashed PINs in the directory. For information on how PINs are stored in the directory, see section “How PINs Are Stored in the Directory” of the PIN Generator tool documentation.

Use the output file for delivering PINs to users after you complete setting up the required authentication method; see “Step 9. Deliver PINs to End Users” on page 568.

Step 3. Enable the AttributePresentConstraints Policy

This step is required for PIN-based enrollment with PIN removal only in certain deployment scenarios. Here's some information that will help you decide whether you need to enable this policy.

In the password and PIN-based enrollment method, users enroll for a certificate using their directory user ID, password and PIN. After a PIN has been used to successfully authenticate a user, the Certificate Manager calls the `PinRemovalListener` module. This module removes the PIN from the authentication directory when the Certificate Manager issues the requested certificate.

Note that listeners in Certificate Management System are objects which register themselves as interested in knowing about certain events—for example, change in the state of a request—and carry out a specific task. For more information on listeners, check the samples directory:

```
<server_root>/cms_sdk/cms_jdk/samples/listeners
```

Once the PIN is removed from the authentication directory, it prevents the user from enrolling for another certificate.

The above mentioned process works smoothly if a Certificate Manager or Registration Manager is configured to use the master directory for authenticating users. The process may not work smoothly in deployment scenarios that involve replicated directories. In these scenarios, you need to use the Attribute Present Constraints policy to verify that the PIN has been removed from the directory. Here's an example of such a scenario:

A Registration Manager acts as an enrollment authority, passing authenticated certificate requests to a Certificate Manager; the users have no direct interaction with the Certificate Manager. The Certificate Manager (CA) and the master corporate directory are behind the firewall. The Registration Manager and a replica of the corporate directory are outside the firewall. The Certificate Manager is configured to communicate with the master corporate directory. The Registration Manager has read-only permission to the replicated corporate directory and it uses the directory for authenticating end entities. Both the Certificate Manager and Registration Manager are configured for password and PIN-based enrollment with the PIN removal feature turned on. The master corporate directory is configured to update its replica (outside the firewall) every 10 minutes.

When a user enrolls for a certificate using the End Entity Services interface of the Registration Manager, it authenticates the user against the replica of the corporate directory. If the user presents a valid user ID, password, and PIN, the Registration Manager authenticates the user successfully and forwards the request to the Certificate Manager. As the Registration Manager is configured for PIN-based enrollment with PIN removal, it attempts to remove the PIN from the replicated directory, but it can't as it has no write permission to the replicated directory; the PIN is still around.

When the Certificate Manager processes the request forwarded by the Registration Manager, it calls the `PinRemovalListener` module, which in turn removes the PIN from the master corporate directory when the Certificate Manager issues the certificate. (The Certificate Manager sends the certificate to the Registration Manager, which in turn sends it to the user.)

Although the Certificate Manager has removed the PIN from the master directory, the replicated directory still has the PIN, because the update hasn't occurred. In the meantime, the user may enroll again successfully (from the Registration Manager) for another certificate and receive it from the Certificate Manager.

The Attribute Present Constraints policy enables you to prevent users from successfully enrolling for multiple certificates from the Registration Manager during the time interval between directory updates. If you configure the Certificate Manager to use this policy to check the master directory for PINs before issuing

certificates, successive certificate requests would fail because the PIN has been removed from the master directory. This way, even if the Registration Manager authenticates successive requests, the Certificate Manager rejects them, thus ensuring that a user has only one certificate.

If you are not familiar with the Attribute Present Constraints policy, see section “AttributePresentConstraints Plug-in Module” in Chapter 3, “Constraints Policy Plug-in Modules” of *CMS Plug-Ins Guide*.

Note that unlike some of the other policy rules, Certificate Management System does not create an instance of the Attribute Present Constraints policy rule during installation. If you created this rule after installation, you can configure the server to use that rule. The instructions below explain how to create a new rule:

1. Log in to the CMS window for the Certificate Manager (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select Certificate Manager, and then select Policies.

The Policy Rules Management tab appears. It lists currently configured policy rules.

4. Click Add.

The Policy Plugin Implementation window appears.

5. Select `AttributePresentConstraints` and click Next.

The Policy Rule Editor window appears. It lists the configuration information required for this policy rule.

The screenshot shows the Policy Rule Editor window with the following configuration:

- Policy Rule ID: `PinCheckForClientCerts`
- Policy Plugin ID: `AttributePresentConstraints`
- enable:
- predicate: `HTTP_PARAMS.certType==client`
- * ldap.ldapconn.host: `corDirectory.siroe.com`
- * ldap.ldapconn.port: `389`
- ldap.ldapconn.secureConn:
- * ldap.ldapconn.version: `3`
- ldap.ldapauth.bindDN: `CN=pinmanager`
- password: `*****`
- ldap.ldapauth.clientCertNickname: (empty)
- * ldap.ldapauth.authType: `BasicAuth`
- * ldap.ldapconn.basedn: `O=siroe.com`
- ldap.ldapconn.minConns: `1`
- ldap.ldapconn.maxConns: `5`
- * attribute: `pin`
- value: (empty)

At the bottom, there is a text field labeled "Ldap attribute to check presence of (default pin)" which is currently empty. The window has OK, Cancel, and Help buttons.

6. Enter the appropriate information.
7. Click OK to save your configuration.

You are returned to the Policy Rules Management tab. If required, click the Reorder button and order the rules as appropriate. For details, see “Step 5. Reorder Policy Rules” on page 623.

Step 4: Add an Authentication Instance

Adding an authentication instance to the CMS configuration involves creating a new instance of an already registered plug-in module, assigning a unique name or ID to the instance, and entering appropriate values for the parameters that define the plug-in you want to create an instance of.

When naming an authentication instance (or rule), be sure to formulate the name using any combination of letters (aA to zZ), digits (0 to 9), an underscore (_), and a hyphen (-); other characters and spaces are not allowed. For example, you can type `My_Auth_Rule` or `MyAuthRule` as the instance name, but not `My Auth Rule`.

Also note that when you add an authentication instance, the CMS configuration is updated with authentication-specific information only. The server does not associate the authentication instance you added with any of the end-user enrollment forms—that is, the end-user servlets that should use this authentication instance are not configured yet. You make this association by manually embedding the authentication instance name in the enrollment forms.

By default, the enrollment forms include authentication instance names listed in Table 15-1. Note that the authentication instances are not created by default; only the instance names are embedded in the forms for your convenience. If you create authentication instances with the default names, you can skip the step (“Step A. Associate the Authentication Instance With the Enrollment Form” on page 559) that explains how to update an enrollment form to associate it with the name of an authentication instance.

Table 15-1 Default authentication instance names embedded in end user enrollment forms

Enrollment form (filename)	Authentication instance name
Certificate-based enrollment for dual certificates (<code>CertBasedDualEnroll.html</code>)	<code>UserDirEnrollment</code>
Certificate-based enrollment for encryption certificates (<code>CertBasedEncryptionEnroll.html</code>)	<code>UserDirEnrollment</code>
Certificate-based enrollment for single certificates (<code>CertBasedSingleEnroll.html</code>)	<code>UserDirEnrollment</code>
Directory-based enrollment (<code>DirUserEnroll.html</code>)	<code>UserDirEnrollment</code>
Directory- and PIN-based enrollment (<code>DirPinUserEnroll.html</code>)	<code>PinDirEnrollment</code>
NIS server-based enrollment (<code>NISUserEnroll.html</code>)	<code>NISAuth</code>
Portal enrollment (<code>PortalEnrollment.html</code>)	<code>PortalEnrollment</code>

Figure 15-5 shows the default directory-based enrollment form configured to use an authentication instance named `UserDirEnrollment`.

Figure 15-5 Authentication information in the default directory-based enrollment form

```

VIM - /export/cms/bin/cert/web/ee/DirUserEnroll.html
Window Edit Options Help
    (navigator.appName == "") {
    document.writeln(
      '<input type="submit" value="Submit" '+
      'name="Send" width="72">');
    }
    else {
      // alert('nsm');
      document.writeln(
        '<input type="button" value="Submit" '+
        'name="submitbutton" '+
        'onclick="validate(form)" width="72">');
    }
    document.write('' +
      '<input type="reset" value="Reset" name="reset" width="72">' +
      '' +
      '<input type="button" value="Help" '+
      'onclick="help(' +
      '"/manual/ee_guide/h1ptxt.htm#Directory-Based User Enrollment" +
      ')"' +
      'name="button" width="72">' +
      '<input type="hidden" name="certType" value="client">' +
      '<input type="hidden" name="authenticator" '+
      'value="UserDirEnrollment">');

    if (navigator.appName == 'Netscape') {
      if ((navMajorVersion() > 3) &&
        (navigator.userAgent.indexOf("MSIE") == -1)) {

```

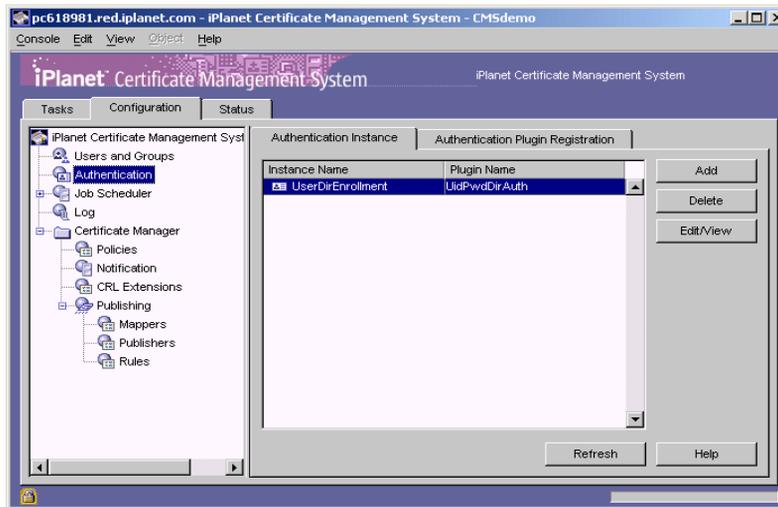
For information on locating and customizing the default end-entity forms, see *CMS Customization Guide*.

To add an authentication instance to the CMS configuration:

1. In the CMS window, select the Configuration tab.

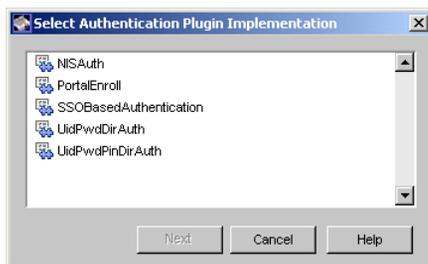
2. In the navigation tree, select Authentication.

The right pane shows the Authentication Instance tab, which lists any currently configured authentication instances.



3. Click Add.

The Select Authentication Plugin Implementation window appears. It lists the currently registered authentication plug-in modules.



4. Select a plug-in module.

The following choices are the ones provided by default with Certificate Management System. If you have registered any custom authentication plug-in modules, they too will be available for selection.

UidPwDirAuth. Select this if you want to use the directory-based authentication module.

UidPwdPinDirAuth. Select this if you want to use the directory- and PIN-based authentication module (with or without PIN removal). To configure Certificate Management System for PIN-based enrollment method, you must have completed “Step 2. Set Up the Directory for PIN-Based Enrollment” on page 547.

NISAuth. Select this if you want to use the NIS server-based authentication module.

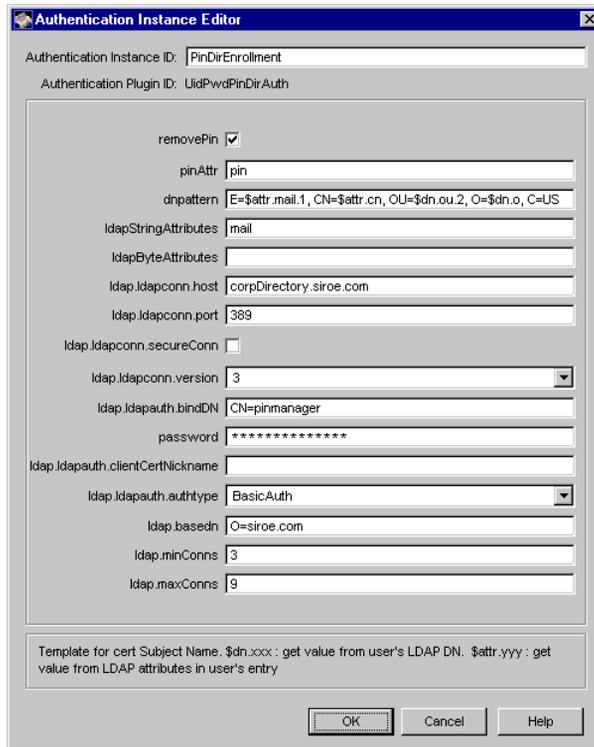
SSOBasedAuthentication. Select this if you want to use the Single Sign-On authentication token that come with Directory Server Access Management Edition (DSAME) 6.0.

PortalEnroll. Select this if you want to use the portal authentication module.

For the purposes of this instruction, assume that you selected `UidPwdPinDirAuth`.

5. Click Next.

The Authentication Instance Editor window appears. The Authentication Instance ID field shows the default instance name embedded in the associated enrollment form (see Table 15-1 on page 554). The Authentication Plugin ID field shows the module you've chosen. The remaining fields list the configuration information required for this authentication instance.



6. If you don't want to use the default instance name, in the Authentication Instance ID field, type a unique name for this instance that will help you identify it.

For the name, be sure to use an alphanumeric string with no spaces. If you chose to use a different name, be sure to edit the default name in the enrollment form in the next step, "Step 5. Set Up the Enrollment Interface" on page 559.

7. Fill in values for the remaining parameters.

8. Click OK.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. Don't restart the server yet; you can do this after you've made all the required changes.

Step 5. Set Up the Enrollment Interface

This step explains how to customize the end-entity interface for the enrollment method you've chosen for your users.

- Step A. Associate the Authentication Instance With the Enrollment Form
- Step B. Customize the Form
- Step C. Hook Up the Certificate-Based Enrollment Form (optional)
- Step D. Remove Unwanted Enrollment Options

Step A. Associate the Authentication Instance With the Enrollment Form

You can skip this step if, in the previous step, you chose the default instance name suggested in Table 15-1 on page 554. Otherwise, you must edit the enrollment form to associate the instance you added because Certificate Management System relies on the authentication instance name embedded in the enrollment form to determine the authentication method.

For the new authentication instance to work with end-entity enrollment forms, you must update the appropriate forms, as follows:

1. In the CMS host system, go to this directory:

```
<server_root>/cert-<instance_id>/web/ee
```
2. Locate the file that corresponds to the authentication module you chose in "Step 4: Add an Authentication Instance" on page 553; use Table 15-1 for guidance.
3. Open the file in a text editor.
4. Locate the attribute that associates the authentication instance with the enrollment form.

In the default enrollment forms, locate this line:

```
<INPUT TYPE="HIDDEN" NAME="authenticator" VALUE="myAuthMgr">
```

In the custom enrollment form, be sure to include the following line, and replace `myAuthMgr` with the name of the authentication instance you added.

```
<INPUT TYPE="HIDDEN" NAME="authenticator" VALUE="myAuthMgr">
```

5. Check the value assigned to `authenticator` attribute (the `VALUE=` field). Make sure that it is same as the name or ID you assigned to the authentication instance you created in Step 5. If it is different, replace it with the name of the authentication instance. For example, if you used `test_auth` as the instance name, the edited line should look like this:

```
<INPUT TYPE="HIDDEN" NAME="authenticator" VALUE="test_auth">
```

6. Save your changes and close the file.

Step B. Customize the Form

You can make any other changes to meet your organization's requirements.

Step C. Hook Up the Certificate-Based Enrollment Form

This step is required only if you want to use any of the certificate-based enrollment forms.

As explained in the “Certificate-Based Enrollment” section in Chapter 1, “Authentication Plug-in Modules” of *CMS Plug-Ins Guide*, Certificate Management System provides three forms for certificate-based enrollment:

- `CertBasedDualEnroll.html`
- `CertBasedEncryptionEnroll.html`
- `CertBasedSingleEnroll.html`

By default, the form named `CertBasedDualEnroll.html` is hooked up to the Enrollment tab of the end-entity interface. You can replace this form with either of the other two forms, `CertBasedEncryptionEnroll.html` and `CertBasedSingleEnroll.html`; you can do this by uncommenting the script relevant to either of the forms in the index file and by commenting out the script for `CertBasedDualEnroll.html`—thus, effectively unhook the old one and hook the new one.

To enable any of the certificate-based enrollment forms, follow these steps:

1. In the CMS host system, go to this directory:

```
<server_root>/cert-<instance_id>/web/ee
```

2. Locate the `index.html` file.
3. Open the file in a text editor.

4. Follow instructions as appropriate:

If you want to enable the `CertBasedDualEnroll.html` form, search for `CertBasedDualEnroll`. You should find a block of script like the following:

```
count++;
}
if (http != 'true') {
    // this one is directory based, cert-based
    if ( isAuthMgrEnabled("UidPwdDirAuth") ) {
        item = 'certBasedDualEnroll';
        menuItems[count] = top.EnrollMenu[count] =
            new menuItem(item, 'CertBasedDualEnroll.html', 'Certificate');
```

If you want to enable the `CertBasedEncryptionEnroll.html` form, search for `CertBasedEncryption`. You should find a block of script like the following:

```
count++;
}
// item = 'certBasedEncEnroll';
// menuItems[count] = top.EnrollMenu[count] =
// new menuItem(item, 'CertBasedEncryptionEnroll.html',
// 'Certificate');
```

Uncomment the lines and then add lines for using the automated enrollment module you configured the server with. Your edited lines should look similar to this:

```
count++;
}
if (http != 'true') {
    // this one is directory based cert-based
    if ( isAuthMgrEnabled("UidPwdDirAuth") ) {
        item = 'certBasedEncEnroll';
        menuItems[count] = top.EnrollMenu[count] =
            new menuItem(item, 'CertBasedEncryptionEnroll.html',
                'Certificate');
```

If you want to enable the `CertBasedSingleEnroll.html` form, search for `CertBasedSingle`. You should find a block of script similar to this:

```

count++;
}
//      item = 'certBasedSingleEnroll';
//      menuItems[count] = top.EnrollMenu[count] =
//          new menuItem(item, 'CertBasedSingleEnroll.html',
//              'Certificate');

```

Uncomment the lines and then add lines for using the automated enrollment module you configured the server with. Your edited lines should look like these:

```

count++;
}
if (http != 'true') {
    // this one is directory based cert-based
    if ( isAuthMgrEnabled("UidPwdDirAuth") ) {
        item = 'certBasedSingleEnroll';
        menuItems[count] = top.EnrollMenu[count] =
            new menuItem(item, 'CertBasedSingleEnroll.html',
                'Certificate');
    }
}

```

5. Make sure to comment out lines for any unused options.
6. If you're using any of the default modules, except for the one provided for the directory-based enrollment (`UidPwdDirAuth`), edit the following line to replace `UidPwdDirAuth` with the name of the module you're using.

```

if ( isAuthMgrEnabled("UidPwdDirAuth") ) {

```

7. By default, a link named `Certificate` will be created under the **Browser** section. If you want to rename the link, replace `Certificate` in the following line with the new name:

```

new menuItem(item, 'CertBasedDualEnroll.html', 'Certificate');

```

8. Save your changes and close the file.

Step D. Remove Unwanted Enrollment Options

This step is optional.

By default, the Enrollment tab of the end-entity interface shows just one link, named `Manual`, under the **Browser** section. The `Manual` link opens a form that enables end users to request certificates manually. When you enable automated enrollment, that is, when you create an instance of any of the automated

enrollment modules, a link for the corresponding form is automatically created under the Browser section. For example, if you create an instance of the directory-based authentication module, you will notice a new link named `Directory` under the Browser section.

If you don't want your end users to see the manual enrollment option, you can remove it from the Enrollment tab altogether. The steps below explain how to remove the Manual link from the Browser section of the Enrollment tab:

1. In the CMS host system, go to this directory:

```
<server_root>/cert-<instance_id>/web/ee
```

2. Locate the `index.html` file.
3. Open the file in a text editor.
4. Locate the `initEnrollMenu` function (the function `initEnrollMenu()` line).
5. Remove or comment out lines that correspond to the Manual enrollment form.

```
count++;
item = 'manuser';
menuItems[count] = top.EnrollMenu[count] =
new menuItem(item, 'ManUserEnroll.html', 'Manual');
```

6. Save your changes and close the file.
7. Open a browser window.
8. Go to the end-entity interface and verify your changes.

Step 6. Enable End-Entity Interaction

You can configure end-entity interaction with a Certificate Manager or a Registration Manager, or with both. End entities cannot interact with a Data Recovery Manager directly; they must interact through a Certificate Manager or Registration Manager. By default, the Certificate Manager is configured for end-entity interaction; the Registration Manager is not configured for end-entity interaction.

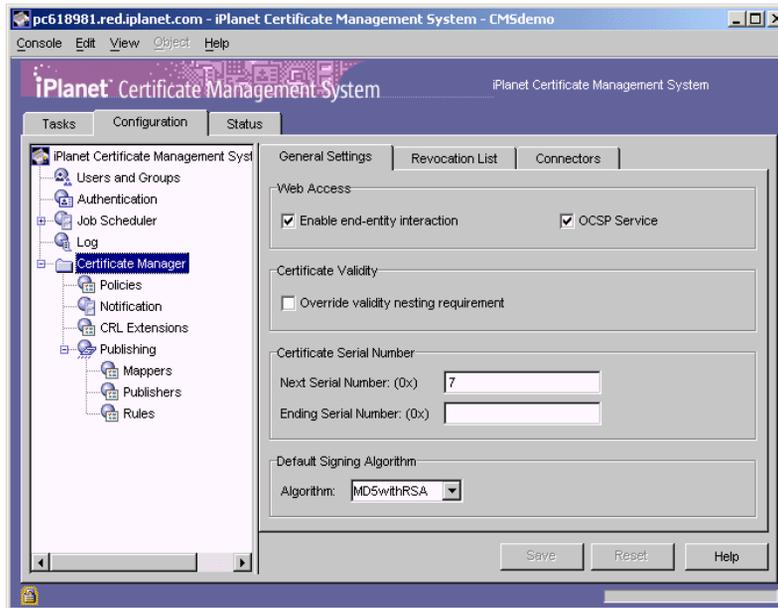
Depending on the subsystem you're configuring, follow the instructions in "Enabling End-Entity Interaction with a Certificate Manager" on page 563 or in "Enabling End-Entity Interaction with a Registration Manager" on page 566.

Enabling End-Entity Interaction with a Certificate Manager

To enable end-entity interaction with a Certificate Manager:

1. In the CMS window, select the Configuration tab.
2. In the navigation tree, select Certificate Manager.

The General Setting tab appears.



3. In the Web Access section, check the “Enable end-entity interaction” option if you want end entities to be able to interact with the selected Certificate Manager via the HTTPS port; leave it unchecked to disable end-entity interaction with the server.

Note that if you disable end-entity interaction, the Network tab still shows the HTTPS port and allows you to configure it (see “Configuring Port Numbers” on page 384). However, you should know that the server ignores this port.

4. In the Certificate Validity section, check the “Override validity nesting requirement” option, if you want the Certificate Manager to issue certificates with validity periods beyond that of its *CA signing certificate*; see “CA Signing Key Pair and Certificate” on page 451).

If you leave the box unchecked and if the Certificate Manager (CA) finds a request with validity period extending beyond that of its CA signing certificate, it automatically truncates the validity period to end on the day the CA signing certificate expires. For example, if the CA signing certificate expires on June 10, 2004, any enrollment or renewal request with validity period beyond June 10, 2004 will have validity period truncated to end on June 10, 2004.

Validity periods of certificates during enrollment is determined by the policy explained in `ValidityConstraints` plug-in module. Similarly, validity periods of certificates during renewal is determined by the policy explained in `RenewalValidityConstraints` plug-in module. Both the modules are explained in *CMS Plug-Ins Guide*.

5. In the Certificate Serial Number section, specify the serial number range for certificates issued by this Certificate Manager. The server assigns the serial number you enter in the “Next serial number” to the next certificate it issues and the number you enter in the “Ending serial number” to the last certificate it issues.

The serial number range enables you to deploy multiple CAs, balancing the number of certificates each CA issues. Note that the combination of an issuer name and a serial number uniquely identifies a certificate. To ensure that two distinct certificates issued by the same authority doesn’t contain the same serial number, make sure the serial number range does not overlap among cloned CAs. (For information on cloning CAs, “Cloning a Certificate Manager” on page 288.)

Also note that when a CA exhausts all its serial numbers, you can revive it by changing the values in the “Next serial number” and “Ending serial number” fields, followed by restarting the Certificate Manager.

6. In the Default Signing Algorithm section, select the signing algorithm the Certificate Manager should use for signing certificates. The choices are “MD2 with RSA,” “MD5 with RSA,” and “SHA1 with RSA,” if the CA’s signing key type is RSA and “SHA1 with DSA,” if the CA’s signing key type is DSA.

Note that the signing algorithm specified in the Certificate Manager's policy configuration overrides the algorithm you select here. For information on a Certificate Manager's policy configuration, see *SigningAlgorithmConstraints* policy plug-in module in *CMS Plug-Ins Guide*.

7. To save your changes, click Save.

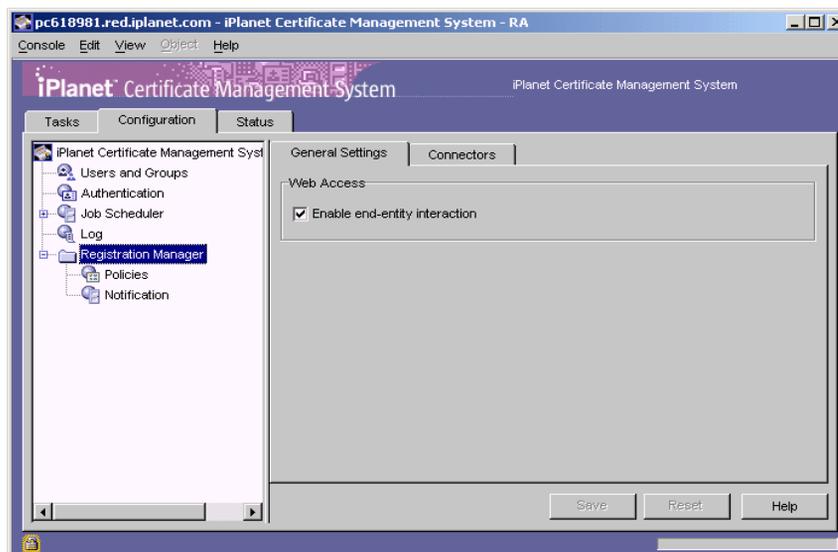
The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Enabling End-Entity Interaction with a Registration Manager

To enable end-entity interaction with a Registration Manager:

1. In the CMS window, select the Configuration tab.
2. In the navigation tree, select Registration Manager.

The General Setting tab appears.



3. In the Web Access section, check the “Enable end-entity interaction” option if you want end entities to be able to interact with the selected Registration Manager via the HTTPS port; leave it unchecked to disable end-entity interaction with the server.

Note that if you disable end-entity interaction, the Network tab still shows the HTTPS port and allows you to configure it (see “Configuring Port Numbers” on page 384). However, you should know that the server ignores this port.

4. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Step 7. Turn on Automated Notification

Both the Certificate Manager and the Registration Manager can send certificate-issuance notification to end users. For details on turning this feature on, see , “Setting Up Automated Notifications.”

Step 8. Test Your Authentication Setup

To test whether your end users can successfully enroll for a certificate using the authentication method you’ve set up:

1. Open a web browser window.
2. Go to the end-entity interface for the enrollment authority you configured.

The default URL is as follows:

```
https://<hostname>:<end_entity_HTTPS_port> or
http://<hostname>:<end_entity_HTTP_port>
```

3. In the Enrollment tab, open the enrollment form you customized.
4. Fill in all the values and submit the request.
5. The client prompts you to enter the password for your key database.
6. When you enter the correct password, the client generates the key pair.

Do not interrupt the key-generation process. Upon completion of the key generation, the request is submitted to the server for certificate issuance. The server subjects the request to the currently configured policy rules and issues the certificate only if the request passes all the policy rules.

Upon receipt of a notification about the certificate issuance, install the certificate in your browser.

7. Verify that the certificate is installed in the browser's certificate database; for example, in Communicator you can open the Security Info window and verify that the certificate is listed in there.
8. If you've set up the directory- and PIN-based authentication with PIN removal, reenroll for another certificate using the same PIN. Your request should get rejected.
9. If you've set up the portal enrollment, verify that an entry for the user is created in the directory. For example, you can point your browser to the portal directory and find out if an entry for the user for whom you requested the certificate exists.

In the URL field, type

```
ldap://<host_name>:<port>/<base_dn>??sub?(uid=<user_id>),  
substituting <host_name> with the fully qualified host name of the Directory  
Server, <port_number> with the port number at which the Directory Server is  
listening to authentication requests from the Certificate Manager <base_dn>  
with the DN to start searching for the user's entry, and <user_id> with the ID  
of the user for whom you requested the certificate.
```

For example, if the directory host name is corpDirectory, port number is 389, base DN is o=siroe.com, and user's ID is jdoe, the URL would look like this:

```
ldap://corpDirectory:389/o=siroe.com??sub?(uid=jdoe)
```

In the resulting page, look for the user's credentials and verify that they match what you specified in the enrollment form. If you've configured Certificate Management System to publish certificates to the same directory (, "Setting Up LDAP Publishing"), you will be able to see the certificate-related information; it typically includes information such as the owner of the certificate, the CA that has issued the certificate, the serial number, the validity period, and the certificate fingerprint.

Step 9. Deliver PINs to End Users

This step is applicable for directory- and PIN-based authentication with or without PIN removal.

After you have confirmed that the PIN-based enrollment works (as it should), deliver the PINs to users so they can use them during enrollment. To protect the privacy of PINs, be sure to use a secure, out-of-band method for delivery. Here are a few suggested delivery methods:

- Encrypted email (S/MIME)—if your company has S/MIME mail set up, you can deliver PINs to users by encrypted mail.
- Mail—you can mail PINs to users, for example along with their pay stubs or slips.
- Personal delivery—you can arrange a secure means of delivering the password to the user, or ask the user to collect it from you in person.

Managing Authentication Instances

This section explains how to use the CMS window to do the following:

- Configuring Authentication for End-User Enrollment
- Deleting an Authentication Instance
- Modifying an Authentication Instance

For information on adding or changing authentication-specific information in the configuration file, see “Changing the Configuration by Editing the Configuration File” on page 359.

Deleting an Authentication Instance

You can delete an authentication instance that you no longer need from the CMS configuration. If you delete an authentication instance, end users will fail to enroll for certificates using the associated enrollment form. If you want the form to work with another authentication instance, you must make the appropriate changes to the form; see “Step 5. Set Up the Enrollment Interface” on page 559.

To delete an authentication instance from the CMS configuration:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, click Authentication.

The right pane shows the Authentication Instance tab, which lists currently configured authentication instances.

4. In the Instance Name list, select the instance you want to delete and click Delete.

5. When prompted, confirm the delete action.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Modifying an Authentication Instance

You can modify an authentication instance by editing its configuration parameter values; you cannot edit the name of an instance. To change the name of an instance, you need to create a new instance exactly like the instance you want to rename, except with a new name, and delete the old instance. For details, see “Step 4: Add an Authentication Instance” on page 553.

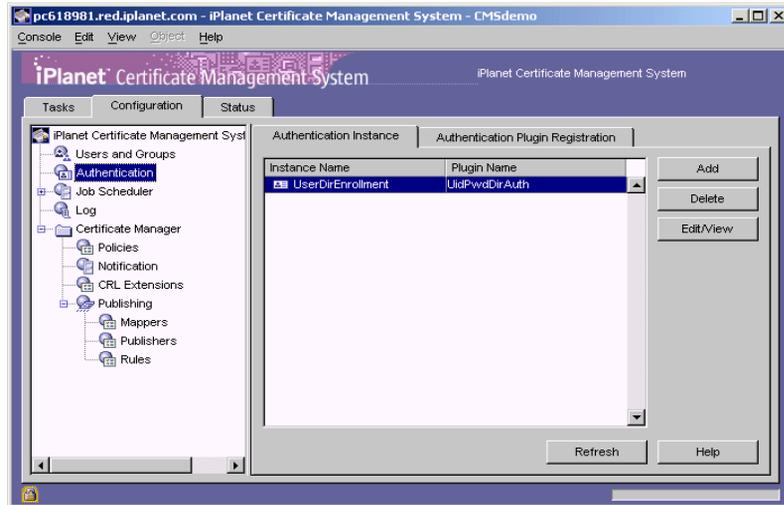
When you modify an authentication instance, the CMS configuration is updated to include the modifications. Because you are not changing the name of the authentication instance, you do not have to make any changes to the end-user servlet configuration.

To modify an authentication instance in the CMS configuration:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.

3. In the navigation tree, click Authentication.

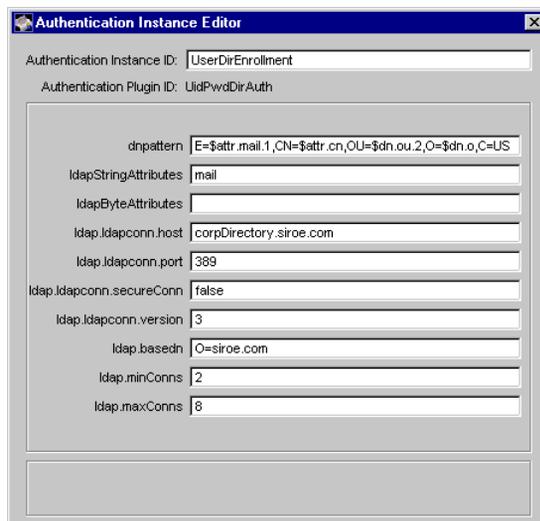
The right pane shows the Authentication Instance tab, which lists configured authentication instances.



4. In the Instance Name list, select the instance you want to modify and click Edit.

The Configure Authentication Instance Parameters window appears, showing the current configuration of this instance.

For the purposes of completing these instructions, assume you selected UserDirEnrollment.



5. Make changes as appropriate. If you need description for any of the parameters, click the Help button or check the *CMS Plug-Ins Guide*.
6. Click OK.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Managing Authentication Plug-in Modules

This section explains how to use the CMS window to do the following:

- Registering an Authentication Module
- Deleting an Authentication Module

For information on adding or changing authentication-specific information in the configuration file, see “Changing the Configuration by Editing the Configuration File” on page 359.

Registering an Authentication Module

You can register custom authentication plug-in modules from the CMS window. Registering a new authentication module involves specifying the name of the module and the full name of the Java class that implements the authentication interface. For example, you can add a Java class, named as follows, that implements an authentication module:

```
com.iplanet.certsrv.authentication.ssnAuth
```

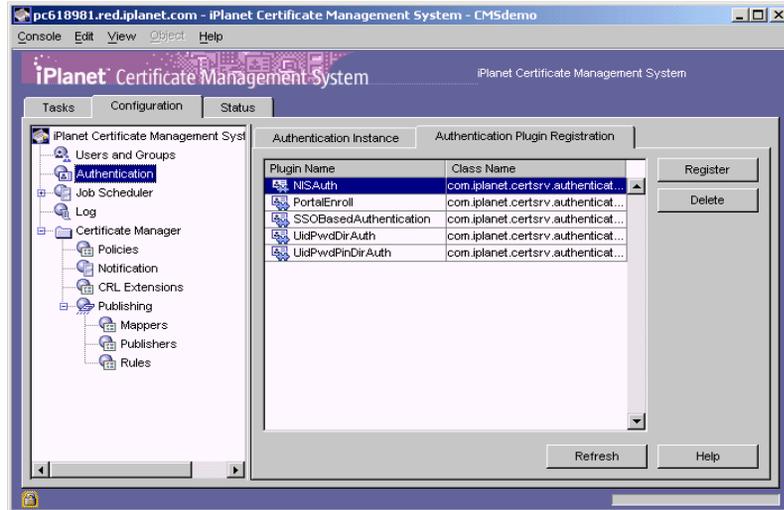
Before registering an authentication module, be sure to put the Java class for the module in the `classes` directory. If you need help, check the tutorials installed in this directory: `<server_root>/cms_sdk/cms_jdk/samples/authentication`

To register an authentication module in the CMS authentication framework:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.

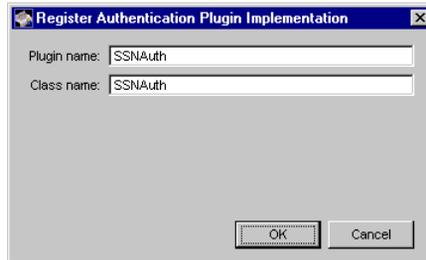
3. In the navigation tree, click Authentication, and in the right pane, click the Authentication Plugin Registration tab.

The tab lists modules that are already registered.



4. Click Register.

The Register Authentication Plugin Implementation window appears.



5. Specify which module you want to register:

Plugin name. Type a name for the module.

Class name. Type the full name of the class for this module—that is, the path to the implementing Java class. If this class is part of a package, be sure to include the package name. For example, if you are registering a class named `NISAuth` and if this class is in a package named `com.mycompany`, type `com.mycompany.NISAuth`.

6. Click OK.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Deleting an Authentication Module

You can delete an authentication plug-in module that you no longer need by using the CMS window. Before deleting a module, be sure to delete all the instances that are based on this module; see “Deleting an Authentication Instance” on page 569. You should also update the appropriate end-entity enrollment forms.

To delete an authentication module from the CMS authentication framework:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, click Authentication, and in the right pane, click the Authentication Plugin Registration tab.

The tab lists the currently registered modules.

4. In the Plugin Name list, select the module you want to delete and click Delete.
5. When prompted, confirm the delete action.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Setting Up Automated Notifications

iPlanet Certificate Management Server can send email notifications automatically when certain events occur. Unlike jobs that are executed on a preconfigured schedule, these notifications are event-driven—that is, whenever an event occurs, the server notifies the user. Notifiable events include certificate issuance and pending requests in an agent queue.

This chapter describes event-driven notifications supported by Certificate Management System and explains how to turn them on. The chapter has the following sections:

- Automated Notifications (page 575)
- Customizing Notification Messages (page 578)
- Configuring a Subsystem to Send Notifications (page 583)

Automated Notifications

You can configure the Certificate Manager and Registration Manager to send automated email notifications to end entities, agents, or administrators when events relevant to these users occur. The email notifications are event-driven—that is, whenever an event occurs, the subsystem generates the notification message and sends it to the user. Event-driven notification involves a listener class in the subsystem that registers an interest in an appropriate event, for example, successful completion of an enrollment request.

Notifiable events include the following:

- Notifications of Certificate Issuance to End Entities—end entities are notified by email that a requested certificate has been issued.

- Notification of New Request in Queue—agents are notified by email that a request has been added to the request queue. Alternatively (or in addition) a schedulable job can notify agents at regular intervals of the current state of the request queue; see “Configuring a Subsystem to Run Automated Jobs” on page 589.

Notifications of Certificate Issuance to End Entities

You can configure the Certificate Manager or Registration Manager to send a notification message to users who have been issued certificates in response to enrollment requests. This message normally includes information about the issued certificate and instructions for importing the certificate into the user’s client.

This kind of notification involves a listener class in the subsystem that registers an interest in an appropriate event, in this case successful completion of an enrollment request. In the CMS configuration, this listener class for a Certificate Manager is defined as `ca.notification.certIssued` and for the Registration Manager it is defined as `ra.notification.certIssued`.

For more information on listeners, check the `samples` directory:

```
<server_root>/cms_sdk/cms_jdk/samples/listeners
```

When a certificate is issued, the listener builds a notification message based on a configured template and sends it to an email address that it determines by using an email resolver. By default the email is obtained from the email address entered in the request or from the certificate.

- The email resolver first checks the request for the email address and if doesn’t find one, it checks the subject name of the certificate for the email address; if the subject name doesn’t include the email address, the resolver checks the certificate for the Subject Alternative Name extension to see whether it specifies the email address. For specifying an email address in the Subject Alternative Name extension, see “Configuring Policy Rules for a Subsystem” on page 613.
- In the absence of an email address, the notification is sent to the email address specified in the “Sender’s Email Address” field, instead of the requestor, as an undeliverable notification. There’ll also be a message to this effect in the logs; see “Monitoring CMS Logs” on page 803.

Note that you can customize the email resolver using the `ReqCertSANNameEmailResolver.java` class included as a sample at this location:

```
<server_root>/cms_sdk/cms_jdk/samples/resolvers
```

The template that the listener uses to construct the email notification message is located in the configured directory. This directory has the following default location: `<server_root>/cert-<instance_id>/emails`

You can configure both the path and filename of the template file. You can also modify the template to customize the contents and appearance of the messages; see “Customizing Message Templates” on page 580.

Messages can be sent as HTML or plain text.

For the `certIssued` listener, you can configure the following:

- Whether the listener is enabled.
- The sender of the notification messages (who will be notified of any delivery problems).
- The location of the notification email template.
- The subject line of the notification message.

Notification of New Request in Queue

When a *deferred* end-entity request enters the request queue of a Certificate Manager or Registration Manager, agents assigned to manage the queue must review the request and reject or accept it. To help ensure that an agent processes the request in a timely manner, you can configure the subsystem to notify agents whenever a new request gets added to the request queue.

This kind of notification involves a *listener* class in the subsystem that registers an interest in an appropriate event, in this case the addition of a request to the request queue. In the CMS configuration, this listener class is identified as follows:

```
ca.notification.requestInQ
```

For more information on listener classes, check the samples directory:

```
<server_root>/cms_sdk/cms_jdk/samples/listeners
```

When a request is added to the queue, the listener builds a notification message based on a configured template and sends it to one or more agents’ email addresses as configured.

The template that the listener uses to construct the email notification message is located in the configured directory. This directory has the following default location: `<server_root>/cert-<instance_id>/emails`

You can configure both the path and filename of the template file. You can also modify the template to customize the contents and appearance of the messages; see “Customizing Message Templates” on page 580.

For the `requestInQ` event listener, you can configure the following:

- Whether the listener is enabled.
- The sender of the notification messages (who will be notified of any delivery problems).
- The location of the notification email template.
- The subject line of the notification message.
- The email addresses of message recipients; these should be subsystem agents whose task is to review *deferred* enrollment requests.

Customizing Notification Messages

Notification and summary email messages are constructed using templates located in the `emails` directory of a CMS instance. This directory has the following default location: `<server_root>/cert-<instance_id>/emails`

Both text and HTML templates are included by default. They are listed in Table 16-1.

Templates for Event-Triggered Notifications

Table 16-1 lists the default template files provided for formulating event-triggered-notification messages. You can customize certain aspects of these templates, such as the subject of the email message and the location and name of the template file, using the CMS window.

Table 16-1 Default templates for event-triggered notifications

Filename	Description
<code>certIssued_CA</code>	Template for the Certificate Manager to send plain-text notifications to end entities upon issuance of certificates.

Table 16-1 Default templates for event-triggered notifications *(Continued)*

Filename	Description
<code>certIssued_CA.html</code>	Template for the Certificate Manager to send HTML-based notifications to end entities upon issuance of certificates.
<code>certIssued_RA</code>	Template for the Registration Manager to send plain-text notifications to end entities upon issuance of certificates.
<code>certIssued_RA.html</code>	Template for the Registration Manager to send HTML-based notifications to end entities upon issuance of certificates.
<code>certRequestRejected_CA</code>	Template for the Certificate Manager to send plain-text notifications to end entities when rejecting their certificate requests.
<code>certRequestRejected_CA.html</code>	Template for the Certificate Manager to send HTML-based notifications to end entities when rejecting their certificate requests.
<code>certRequestRejected_RA</code>	Template for the Registration Manager to send plain-text notifications to end entities when rejecting their certificate requests.
<code>certRequestRejected_RA.html</code>	Template for the Registration Manager to send HTML-based notifications to end entities when rejecting their certificate requests.
<code>reqInQueue</code>	Template for the Certificate Manager or Registration Manager to send plain-text notifications to agents when a request enters the queue.
<code>reqInQueue.html</code>	Template for the Certificate Manager or Registration Manager to send plain-text notifications to agents when a request enters the queue.

Note that the email notification that a certificate has been issued is based on a template file whose default name begins with `certIssued`. Similarly, the email notification that a certificate has been rejected is based on a template file whose name begins with `certRequestRejected`. This template file must be located in the same directory as the certificate-issuance template. Unlike the certificate-issuance template, the filename of the certificate-rejection template (`certRequestRejected`) cannot be changed. However, the file extension for the `certRequestRejected` file can be changed, as long as it exactly matches the file extension specified for the certificate issuance template file. For example, if the certificate issuance template

file is named `certIssued_CA.htm`, the `certRequestRejected` file must be named `certRequestRejected.htm`. The HTML file extensions permitted are `.htm`, `.html`, `.HTM`, and `.HTML`. Template files with any other extension (or no extension) are treated as text files.

If you change the name of any of these files, be sure to make the appropriate changes to the configuration (see the “Content template file” field on page 585 and page 586). In the CMS configuration, template files for event-triggered notifications are identified as follows:

```
<subsystem>.notification.<notification_name>.emailTemplate=  
<template_file_path>
```

```
<subsystem>.notification.<notification_name>.emailTemplate=  
<template_file_path>
```

`<subsystem>` specifies the prefix that identifies the subsystem to which the configuration parameter belongs—`ca` for the Certificate Manager and `ra` for the Registration Manager.

`<notification_name>` specifies the name of the event-triggered notification—`certIssued` for the certificate issuance notifications to end entities and `requestInQ` for the request in queue notifications to agents.

`<template_file_path>` specifies the path, including the filename, to the directory that contains the template to be used for formulating the message content.

Tokens, which you can use as variables in the body of the message, are also defined for each template, enabling you to customize the message; a token is replaced by its current variable value in the constructed message. For details, see “Customizing Message Templates” on page 580.

Customizing Message Templates

You can modify the templates to customize the contents and appearance of messages. The message body can contain HTML or plain text. In the body of the message, you can use tokens or keywords as variables. A token is indicated by the dollar character (\$) and is replaced by its current variable value in the constructed message. Different tokens are available for each job or notification class. These are listed in “Tokens Available in Message Templates” on page 581.

For example, a certificate-issuance-notification message can make use of tokens as follows:

```

-----
CERTIFICATE ISSUANCE NOTIFICATION
-----

Your certificate request ($RequestId) has been processed
successfully. Details of your certificate are as follows:

Serial Number= $SerialNumber
SubjectDN= $SubjectDN
IssuerDN= $IssuerDN
Validity Period= $NotBefore - $NotAfter

To get your certificate, please follow this URL:

https://$HttpHost:$HttpPort/getCertFromRequest?requestId=$Request
Id

If you have any questions or problems, please send an email to
cert_central@siroe.com.

Thank you.

```

Tokens Available in Message Templates

This section explains the tokens provided in the templates used by the default job plug-in and event-triggered notification modules to formulate notification messages.

- Tokens for Certificate Issuance Notifications to End Entities
- Tokens for Rejection Notifications to End Entities
- Tokens for Request In Queue Notification Messages

Tokens for Certificate Issuance Notifications to End Entities

Table 16-2 lists the tokens that are available in the message templates provided for formulating the content of email notifications to end entities; a Certificate Manager or Registration Manager can send these notifications upon issuance of the certificates they requested.

Table 16-2 Tokens defined in templates used for certificate-issuance notifications

Token	Description
\$HttpHost	Specifies the fully qualified host name of the Certificate Manager or Registration Manager to which end entities should connect to retrieve their certificates. (This token enables you to construct the URL from which end entities can download their certificates; see the example in “Customizing Message Templates” on page 580.)
\$HttpPort	Specifies the port number at which the Certificate Manager or Registration Manager is listening to end-entity requests. (This token enables you to construct the URL from which end entities can download their certificates; see the example in “Customizing Message Templates” on page 580.)
\$InstanceID	Specifies the ID assigned to the subsystem that sent this notification. <ul style="list-style-type: none"> • If the notification is sent by a Certificate Manager, this will be <code>ca</code>. • If the notification is sent by a Registration Manager, this will be <code>ra</code>.
\$IssuerDN	Specifies the distinguished name of the CA that issued the certificate.
\$NotAfter	Specifies the <code>NotAfter</code> attribute.
\$NotBefore	Specifies the <code>NotBefore</code> attribute.
\$RecipientEmail	Specifies the email address of the recipient (the address resolved from the email resolver explained in “Notifications of Certificate Issuance to End Entities” on page 576).
\$RequestId	Specifies the request ID.
\$SerialNumber	Specifies the serial number of the certificate that has been issued; the serial number will be displayed as a hexadecimal value in the resulting message.
\$SenderEmail	Specifies the email address of the sender (it is the same as the one you specify in the <code>Sender's E-mail Address</code> field in “Step 2. Turn On Certificate-Issuance Notification” on page 584).
\$SubjectDN	Specifies the distinguished name of the certificate subject.

Tokens for Rejection Notifications to End Entities

Table 16-3 lists tokens that are available in the message templates provided for formulating the content of email notifications to end entities; a Certificate Manager or Registration Manager can send these notifications to end entities when rejecting certificate requests.

Table 16-3 Tokens defined in templates used for request-rejection notifications

Token	Description
<code>\$InstanceID</code>	Specifies the ID assigned to the subsystem that sent this notification. <ul style="list-style-type: none"> • If the notification is sent by a Certificate Manager, this will be <code>ca</code>. • If the notification is sent by a Registration Manager, this will be <code>ra</code>.
<code>\$RequestId</code>	Specifies the request ID.

Tokens for Request In Queue Notification Messages

Table 16-4 lists the tokens that you can use for formulating the content of the `RequestInQueueJob` job's summary report.

Table 16-4 Tokens for the request-in-queue job's summary report

Token	Description
<code>\$InstanceID</code>	Specifies the ID assigned to the subsystem that sent this notification. <ul style="list-style-type: none"> • If the notification is sent by a Certificate Manager, this will be <code>ca</code>. • If the notification is sent by a Registration Manager, this will be <code>ra</code>.
<code>\$ExecutionTime</code>	Specifies the time the job (instance) was run.
<code>\$RecipientEmail</code>	Specifies the email address of the recipient.
<code>\$SenderEmail</code>	Specifies the email address of the sender (it is the same as the one you specify in the <code>Sender's E-mail Address</code> field in "Step 3. Turn on Request in Queue Notification" on page 585).
<code>\$SummaryTotalNum</code>	Specifies the total number of items (certificate requests that are pending in the queue) in the summary report.

Configuring a Subsystem to Send Notifications

To configure a Certificate Manager or Registration Manager to send event-driven notifications, follow these steps:

- Step 1. Before You Begin
- Step 2. Turn On Certificate-Issuance Notification

- Step 3. Turn on Request in Queue Notification
- Step 4. Verify Mail Server Settings
- Step 5. Test Your Configuration

Step 1. Before You Begin

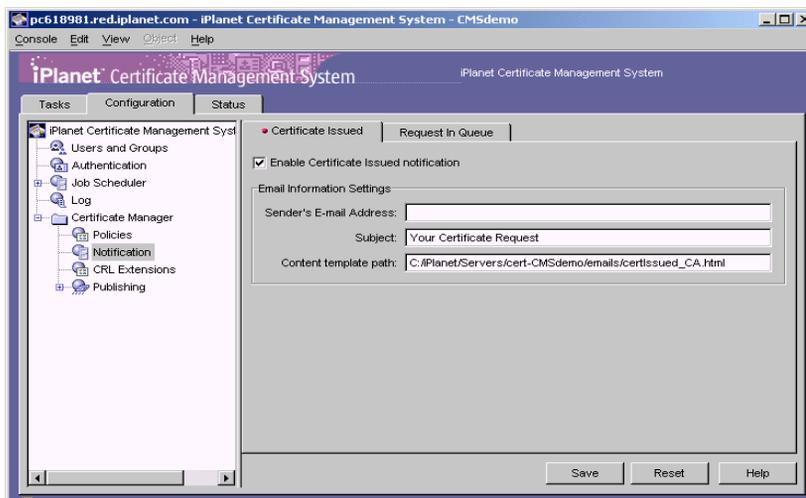
- Read section “Automated Notifications” on page 575 and decide which of the two notification features you want to turn on.
- Read section “Customizing Notification Messages” on page 578 and customize the message templates for the notifications you want to turn on.

Step 2. Turn On Certificate-Issuance Notification

Skip to the next step if you don’t want to turn this feature on.

To configure a Certificate Manager or Registration Manager to send certificate-issuance notifications to end entities:

1. Access the CMS window (see “Logging In to the CMS Window” on page 351).
2. Click the Configuration tab.
3. In the navigation tree, select the subsystem, then click Notification. (The figure below shows the Certificate Manager’s notification feature; the Registration Manager also has a similar feature.)



- To enable the notification feature, check the “Enable Certificate Issued notification” option.
- In the Email Information Settings section, enter information as appropriate:

Sender’s E-mail Address. Type the sender’s full email address (this is the person who should be notified of any delivery problems).

Subject. Type the subject title for the notification.

Content template path. Type the path, including the filename, to the directory that contains the template to be used for formulating the message content.

- To save your changes, click Save.

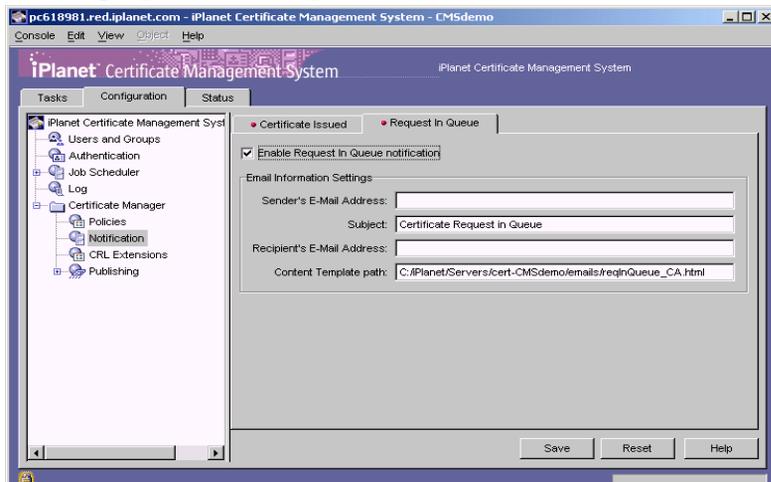
The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. Don’t restart the server yet.

Step 3. Turn on Request in Queue Notification

Skip to the next step if you don’t want to turn this feature on.

To configure a Certificate Manager or Registration Manager to send email notifications to its agents:

- In the navigation tree of CMS window, select the subsystem, and then click Notification. (The figure below shows the Certificate Manager’s notification feature; the Registration Manager also has a similar feature.)
- In the right pane, click Request In Queue.



3. To enable the notification feature, check the “Enable Request In Queue notification” option.

4. Enter information as appropriate:

Sender’s E-Mail Address. Type the sender’s full email address (this is the person who should be notified of any delivery problems).

Subject. Type the subject title for the notification—for example, “End Entity Request in Queue.”

Recipient’s E-Mail Address. Type the recipient’s full email address (this is the person who will check the queue). You can specify more than one recipient; separate email addresses by commas.

Content template path. Type the path, including the filename, to the directory that contains the template to be used for formulating the message content.

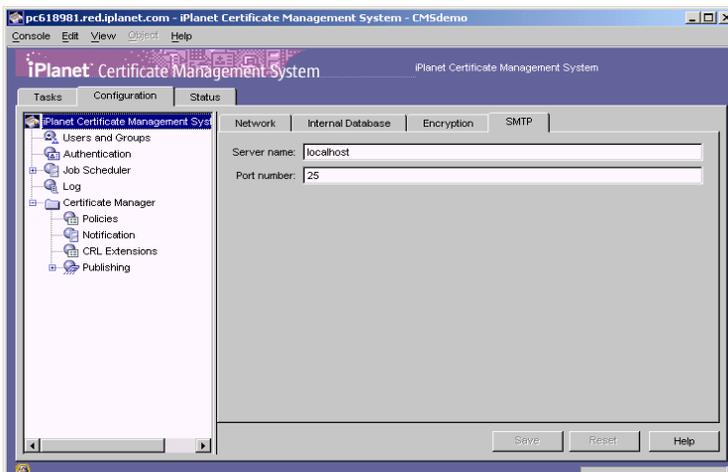
5. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. Don’t restart the server yet.

Step 4. Verify Mail Server Settings

To identify the mail server that the Certificate Manager or Registration Manager should use for routing email notifications:

1. In the CMS window, select the Configuration tab, and then in the right pane, select the SMTP tab.



2. Identify the mail server by providing the following details:

Server name. Make sure the field shows the correct host name for your mail server. Otherwise, type the full host name of the machine on which your mail server is installed. Certificate Management System uses this name to access the mail server. The format for the host name is as follows:

```
<machine_name>.<your_domain>.<domain>
```

By default, the host name of the mail server is shown as `localhost` instead of the actual host name (for example, `mail.siroe.com`).

Port number. Make sure that the field shows the correct port number at which the mail server is listening for requests. Otherwise type the port number.

3. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Step 5. Test Your Configuration

To test whether the subsystem you configured sends email notifications:

1. Change the email addresses in the notification configuration to your email address.
2. Go to the end-entity interface and request a certificate using the manual enrollment form. When the request gets queued for agent approval, you should get notified. Check the message to see if has the correct information. Otherwise, correct the message template.
3. Next, login to the agent interface and approve the request. When the server issues a certificate, you should get an email notification.

Scheduling Automated Jobs

iPlanet Certificate Management Server (CMS) provides a customizable Job Scheduler component that supports various mechanisms for scheduling `cron` jobs. This chapter explains how to configure Certificate Management System to use specific job plug-in modules for accomplishing jobs. The chapter also shows how plug-in implementations and configured instances for various job items appear in the configuration file.

The chapter has the following sections:

- Configuring a Subsystem to Run Automated Jobs (page 589)
- Managing Job Plug-in Modules (page 599)

Configuring a Subsystem to Run Automated Jobs

You can configure the Certificate Manager and Registration Manager to run automated jobs, that is execute specific jobs at specified times. The steps are as follows:

- Step 1. Before You Begin
- Step 2. Modify Existing Jobs
- Step 3. Delete Unwanted Jobs
- Step 4. Add New Jobs
- Step 5. Schedule the Frequency
- Step 6. Verify Mail Server Settings
- Step 7. Test Your Configuration

For information on adding or changing job-specific information in the configuration file, see “Changing the Configuration by Editing the Configuration File” on page 359.

Step 1. Before You Begin

Before configuring a Certificate Manager or Registration Manager to run jobs, be sure to do the following:

- Read Chapter 2, “Job Plug-in Modules” of *CMS Plug-Ins Guide*, and determine the jobs you want the server to run. To locate an online version of this guide, see “Where to Go for Related Information” on page 29.

Jobs that you might want to schedule include email notifications of timed events (such as the expiration of a certificate) that require action on the part of users, and periodic activities such as removing expired certificates from the publishing directory.

- Each job uses templates for formulating the notification-message and summary-message contents. Be sure to read the “Customizing Notification Messages” section to get familiar with the templates the server uses for formulating notification messages. If you want to customize them, do that before you start configuring a job plug-in; check the list of tokens to see if they’re useful for customizing your message body.

Step 2. Modify Existing Jobs

Modifying a job involves changing the configuration parameter values of the job instance; you cannot change the name of a job. To change the name of a job, create a new job using the same job plug-in module (that you used to create the job you want to rename) with the same parameter values, and delete the old one.

As a part of modifying a job, you can change its status from enabled to disabled or vice versa by checking or unchecking the `enable` parameter. A subsystem executes only those jobs that are enabled.

During installation, the Certificate Manager and Registration Manager automatically create a set of jobs (that you would most likely want to use) using the job plug-in modules registered by default. Figure 17-1 shows the jobs created for a Certificate Manager. The Registration Manager also has a similar list. Table 17-1 summarizes the default jobs created for both Certificate Manager and Registration Manager.

After installation, you must verify whether you want to use these jobs, check how these jobs are configured, and make the appropriate configuration changes. If you don't want to use a job, delete it from the configuration following the instructions in “Step 3. Delete Unwanted Jobs” on page 593; alternatively, you may keep it in the disabled state. If you want to create a new job, follow the instructions in “Step 4. Add New Jobs” on page 593.

Figure 17-1 Default jobs created for a Certificate Manager

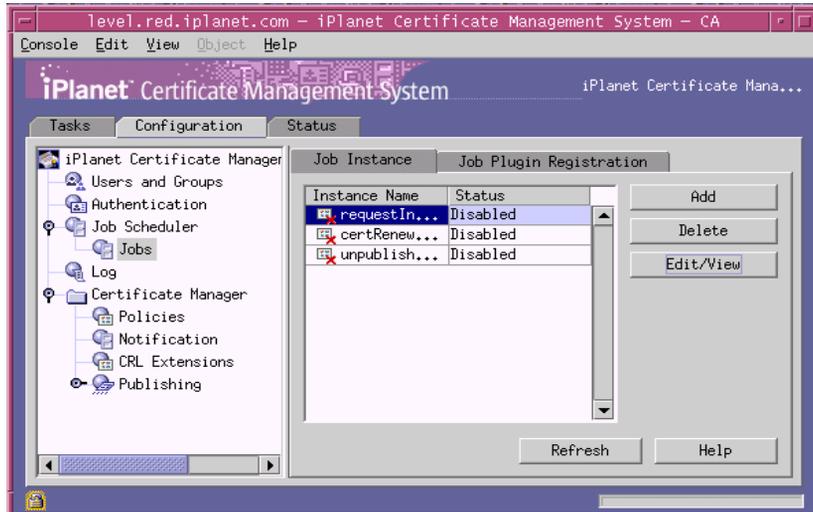


Table 17-1 Default jobs created for a Certificate Manager and Registration Manager

Job name	Certificate Manager	Registration Manager
certRenewalNotifier	Yes	Yes
requestInQueueNotifier	Yes	Yes
unpublishExpiredCerts	Yes	No

To modify a configured job in the CMS configuration:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.

3. In the navigation tree, select Job Scheduler, then select Jobs.

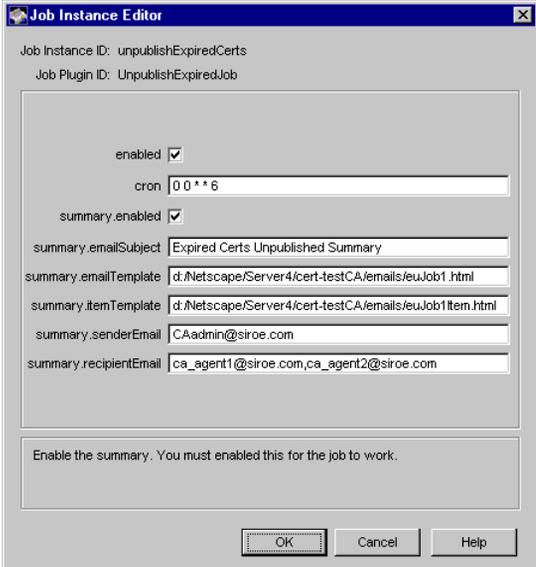
The Job Instance tab appears (Figure 17-1) showing the default jobs.

4. In the Instance Name list, select a job that you want to modify.

For the purposes of this instruction, assume that you selected the job named `unpublishExpiredCerts`.

5. Click Edit/View.

The Job Instance Editor window appears, showing how this job is currently configured. An example is shown below.



The screenshot shows a window titled "Job Instance Editor" with a close button in the top right corner. The window displays the following configuration for the job "unpublishExpiredCerts":

- Job Instance ID: unpublishExpiredCerts
- Job Plugin ID: UnpublishExpiredJob
- enabled:
- cron: 0 0 * * 6
- summary.enabled:
- summary_emailSubject: Expired Certs Unpublished Summary
- summary_emailTemplate: d:\Netscape\Server4\cert-testCA\emails\euJob1.html
- summary_itemTemplate: d:\Netscape\Server4\cert-testCA\emails\euJob1Item.html
- summary_senderEmail: CAAdmin@siroe.com
- summary_recipientEmail: ca_agent1@siroe.com,ca_agent2@siroe.com

At the bottom of the window, there is a message: "Enable the summary. You must enabled this for the job to work." Below this message are three buttons: "OK", "Cancel", and "Help".

6. Make the necessary changes and click OK.
7. Repeat steps 4 through 6 for the remaining jobs.
8. Click Refresh.

Step 3. Delete Unwanted Jobs

You can delete unwanted jobs from the CMS configuration, by using the CMS window. If you think you might need a job in the future, instead of deleting it from the configuration you should disable it by setting the `enable` parameter value to `false`. In this way, you can avoid re-creating the job in the future. Because Certificate Management System executes only those jobs that are currently enabled, keeping unwanted jobs in a disabled state in the configuration does not affect the server's functioning.

To delete a job from the CMS configuration:

1. In the Job Instance tab, select the job you want to delete and click Delete.
2. When prompted, confirm the delete action.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. Don't restart the server yet; you can do so after you've made all the required changes.

Step 4. Add New Jobs

Typically, you don't need to create a new job because jobs for all the default plug-ins are created for you during installation. However, in certain circumstances, for example, if you deleted a default instance, you might have to create a new job.

Adding a job to the CMS configuration involves creating a new instance of an already registered plug-in module, assigning a unique name (an alphanumeric string with no spaces) for the instance, and entering appropriate values for the parameters that define the plug-in module you want to create an instance of. When you add a job, the CMS configuration is updated with the appropriate information.

When naming a job, be sure to formulate the name using any combination of letters (aA to zZ), digits (0 to 9), an underscore (`_`), and a hyphen (`-`); other characters and spaces are not allowed. For example, you can type `My_Job` or `MyJob` as the instance name, but not `My Job`.

Figure 17-2 shows the job modules registered with a Certificate Manager. The Registration Manager also has a similar list. Table 17-2 summarizes the default modules registered with both Certificate Manager and Registration Manager. If you have registered any custom job modules (see "Registering a Job Module" on page 600), they too will be available for selection.

Figure 17-2 Default job modules registered with a Certificate Manager

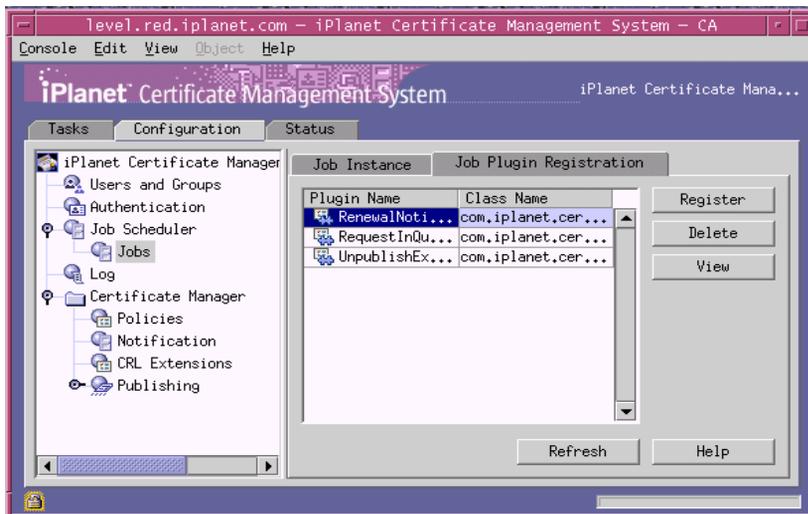


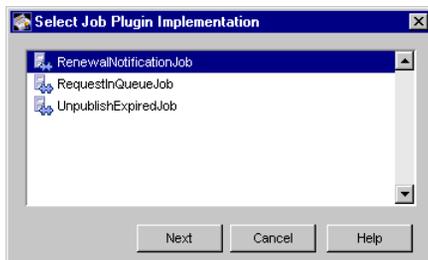
Table 17-2 Job modules registered with a Certificate Manager and Registration Manager

Job plug-in module name	Provided with Certificate Manager	Provided with Registration Manager
RenewalNotificationJob	Yes	Yes
RequestInQueueJob	Yes	Yes
UnpublishExpiredJob	Yes	No

To add a job to the CMS configuration:

1. In the Job Instance tab, click Add.

The Select Job Plugin Implementation window appears.



2. Select a module.

For the purposes of this instruction, assume that you selected the `RenewalNotificationJob` module.

3. Click Next.

The Configure Job Instance Parameters window appears. It lists the configuration information required for this job.

4. Enter the appropriate information.

Job Instance ID. Type a unique name that will help you identify the job. Be sure to formulate the name using any combination of letters (aA to zZ), digits (0 to 9), an underscore (`_`), and a hyphen (`-`). For example, you can type `My_Job` or `MyJob` as the instance name, but not `My Job`.

enabled. To enable the job, type `true`; to disable the job, type `false`.

cron. Specifies the cron specification for when this job should be run. In other words, it specifies the time at which the Job Scheduler daemon thread should check the certificates for sending renewal notifications. For example, `0 3 * * 1-5`.

notifyTriggerOffset. Type the number of days before certificate expiration the first notification should be sent. For example, if you want the server to send renewal notifications to users 30 days before their certificates expire, type 30.

notifyEndOffset. Type the number of days after the certificate expire notifications will continue to be sent, if the certificate is not renewed. For example, if you want the server to continue sending renewal notifications to users (if they don't renew their certificates) 30 days after their certificates expire, type 30.

senderEmail. Type the complete email address to which the server should send notifications regarding any delivery problems. For example, `CertCentral@siroe.com`.

emailSubject. Type the subject line of the notification message; the subject line must be an alphanumeric string of up to 255 characters. For example, `Certificate Renewal Notification`.

emailTemplate. Type the path, including the filename, to the directory that contains the template to be used for formulating the message content. For example: `C:/iplanet/servers/cert-testCA/emails/renewJob.txt`.

summary.enabled. Type `true` if you want the server to compile a summary report of renewal notifications and send. Type `false` if you don't want the server to compile a summary report of renewal notifications.

summary.recipientEmail. Type the email addresses of recipients of the summary report; when specifying multiple recipients, separate addresses by commas. These can be, for example, agents who need to know the status of user certificates. For example, `ca_agent1@siroe.com`, `ca_agent2@siroe.com`.

summary.senderEmail. Type the full email address of the sender (of the summary message); in case of a delivery problem, the server will send a notification to this address. For example, `CAadmin@siroe.com`.

summary.emailSubject. Type the subject line of the summary message; the subject line must be an alphanumeric string of up to 255 characters. For example, `Certificate Renewal Notification Summary`.

summary.itemTemplate. Type the path, including the filename, to the directory that contains the template to be used for formulating the content and format of each item to be collected for the summary report (see the `summary.emailTemplate` parameter below). For example, `C:/iplanet/servers/cert-testCA/emails/renewJobItem.txt`.

summary.emailTemplate. Type the path, including the filename, to the directory that contains the template to be used for formulating the summary report. For example, `C:/iplanet/servers/cert-testCA/emails/renewJobSummary.txt`.

5. Click OK.

You are returned to the Policy Rules Management tab.

6. Repeat steps 1 through 5 and create additional rules, if required.

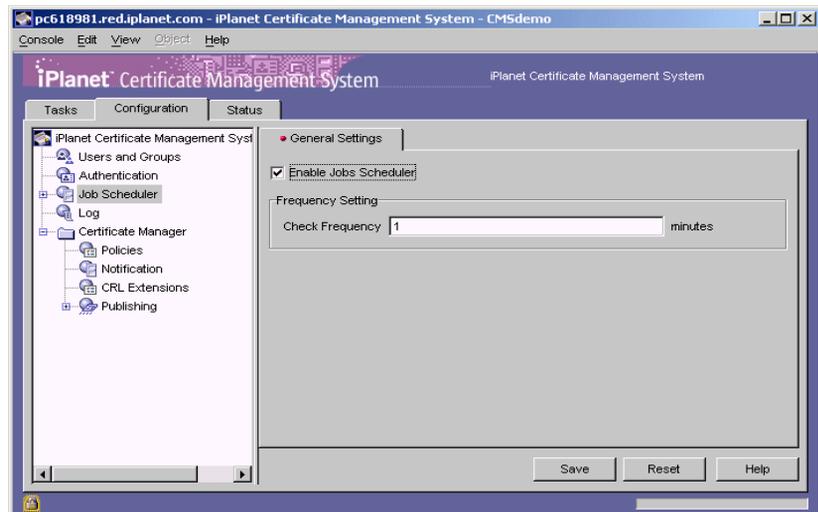
Step 5. Schedule the Frequency

The Certificate Manager and Registration Manager can execute a job only if the Job Scheduler is turned on (or enabled). As a part of turning the Job Scheduler on, you also specify the frequency at which the Job Scheduler daemon should check if any of the configured jobs need to be executed.

To schedule the interval for executing the job:

1. In the navigation tree, click Job Scheduler.

The General Settings tab appears. It shows whether the Job Scheduler component is currently enabled or disabled.



2. Enter information as appropriate:

Enable Job Scheduler. Check this option to enable the Job Scheduler. To disable the Job Scheduler uncheck the option; disabling turns off all the jobs.

Check Frequency. Type the frequency at which the Job Scheduler daemon thread should wake up and call the configured jobs that meet the cron specification. By default, it is set to one minute.

3. To save your changes, click Save.

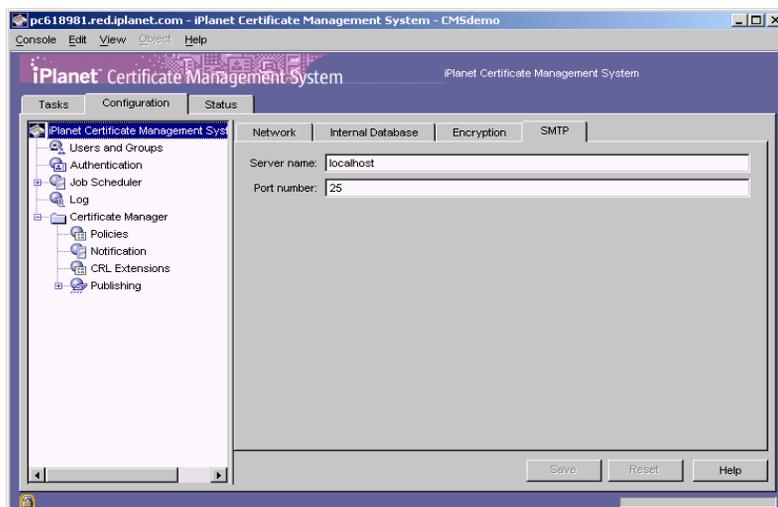
The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Step 6. Verify Mail Server Settings

The Certificate Manager and Registration Manager use the mail server specified in the SMTP tab (of the CMS window) for routing or sending email notifications automatically. If you've already set it up, you should verify that the host name and port number of the mail server are accurate. Otherwise, follow the procedure below to specify the mail server details.

To identify the mail server that the Certificate Manager or Registration Manager should use for routing email notifications:

1. In the CMS window, select the Configuration tab, and then in the right pane, select the SMTP tab.



2. Identify the mail server by providing the following details:

Server name. Make sure the field shows the correct host name for your mail server. Otherwise, type the full host name of the machine on which your mail server is installed. Certificate Management System uses this name to access the mail server. The format for the host name is as follows:

```
<machine_name>.<your_domain>.<domain>
```

By default, the host name of the mail server is shown as `localhost` instead of the actual host name (for example, `mail.siroe.com`).

Port number. Make sure that the field shows the correct port number at which the mail server is listening for requests. Otherwise type the port number.

3. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Step 7. Test Your Configuration

- Change the email addresses in the notification configuration to your email address.
- Go to the end-entity interface and request a certificate using the manual enrollment form. When the request gets queued for agent approval, you should get notified. Check the message to see if has the correct information. Otherwise, correct the message template.
- Next, login to the agent interface and approve the request. Whren the server issues a certificate, you should get a notification.

Managing Job Plug-in Modules

This section explains how to use the CMS window to perform the following operations:

- Registering a Job Module
- Deleting a Job Module

For information on adding or changing job-specific information in the configuration file, see “Changing the Configuration by Editing the Configuration File” on page 359.

Registering a Job Module

You can register custom job plug-in modules from the CMS window. Registering a new module involves specifying the name of the module and the full name of the Java class that implements the module. For example, you can add a job implementation named as follows:

```
com.iplanet.jobscheduler.unpublishUserCert
```

Before registering a module, be sure to put the Java class for the module in the `classes` directory (the implementation must be on the class path).

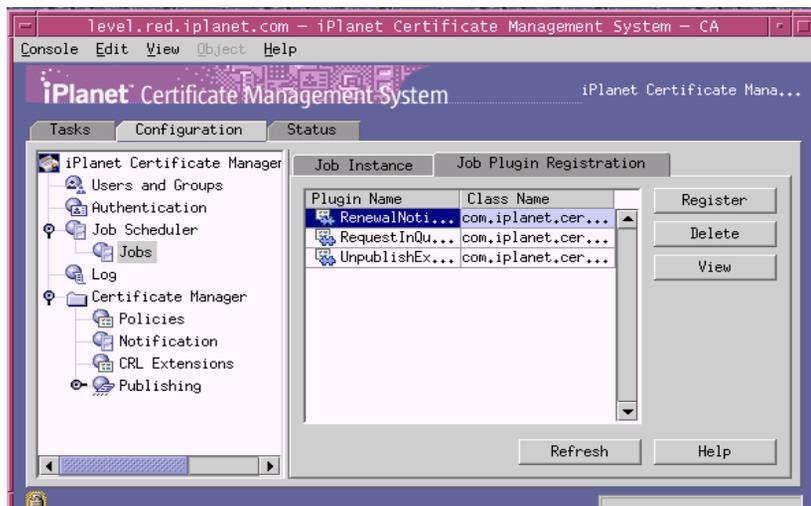
To register a job module in the CMS framework:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select Job Scheduler, then select Jobs.

The Job Instance tab appears. It lists any currently configured jobs.

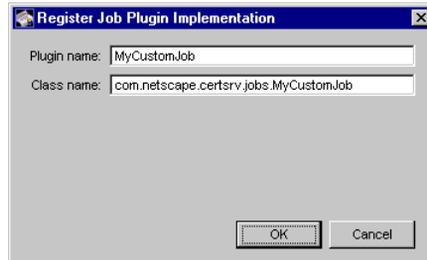
4. Select the Job Plugin Registration tab.

The Job Plugin Registration tab appears.



5. Click Register.

The Register Job Scheduler Plugin Implementation window appears.



6. Specify information as appropriate:

Plugin name. Type a name for the plug-in module.

Class name. Type the full name of the class for this module—that is, the path to the implementing Java class. If this class is part of a package, be sure to include the package name. For example, if you are registering a class named `myJob` and if this class is in a package named `com.myCompany`, type `com.myCompany.myJob`.

7. Click OK.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Deleting a Job Module

You can delete unwanted job plug-in modules by using the CMS window. Before deleting a module, be sure to delete all the instances that are based on this module; for instructions, see “Step 3. Delete Unwanted Jobs” on page 593.

To delete a job module from the CMS framework:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select Job Scheduler, then select Jobs.

The Job Instance tab appears. It lists any currently configured instances.

4. Select the Job Plugin Registration tab.

The Job Plugin Registration tab appears. It lists currently registered job modules.

5. In the Plugin Name list, select the module you want to delete and click Delete.
6. When prompted, confirm the delete action.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Setting Up Policies

iPlanet Certificate Management Server (CMS) provides a customizable policy framework for its main subsystems, the Certificate Manager, Registration Manager, and Data Recovery Manager. This chapter explains how to configure these subsystems to apply organizational and other policies on incoming certificate and key-related requests.

The chapter has the following sections:

- Introduction to Policy (page 603)
- Configuring Policy Rules for a Subsystem (page 613)
- Using JavaScript for Policies (page 626)
- Managing Policy Plug-in Modules (page 626)

Introduction to Policy

You can configure the main subsystems of iPlanet Certificate Management Server (CMS)—the Certificate Manager, Registration Manager, and Data Recovery Manager—to apply certain organizational policies on an end entity's certificate enrollment and management requests before servicing them. For example, some of the policies you might want a Certificate Manager to impose on these requests may include setting a minimum and maximum limit on validity period and key length of certificates, setting extensions based on the end entity's role within an organization, setting signing algorithms, and so on.

This section provides an overview of policy in general. Topics include:

- What Is Policy?
- Policy Rules
- Policy Processor

What Is Policy?

Policy refers to a set of rules that Certificate Management System uses to evaluate or verify an incoming request from an end entity and to determine the outcome; the incoming requests that are governed by policies include certificate issuance, certificate renewal, certificate revocation, key archival, and key recovery requests. For example, in the case of a certificate issuance request, the outcome would be the certificate content.

- A Certificate Manager's policy can include rules for evaluating certificate formulation, signing, renewal, and revocation requests. For example, you can configure a Certificate Manager's policy to impose restrictions on validity length, key type, key length, subject name, extensions, and signing algorithm during certificate issuance.
- A Registration Manager's policy can include rules for verifying incoming certificate issuance, renewal, and revocation requests from end entities in order to formulate the certificate content before forwarding the requests to a Certificate Manager for signing. For example, you can configure a Registration Manager's policy to impose restrictions on validity period, key length, subject name, and extensions. In general, policies for Registration Manager are largely the same as for Certificate Manager.
- A Data Recovery Manager's policy can include rules for verifying users' encryption private key archival and recovery requests.

Using policies, you can configure Certificate Management System to perform one or more of the following operations on each certificate issuance or management request it receives:

- Screen the request for specific content, and modify, reject, or defer (for agent approval) it accordingly. For example, the request might be checked for the inclusion of organizational constraints, such as key algorithm, key size, validity period, or a particular signing algorithm; if it did not meet the requirement, the subsystem would modify the request or return an error, depending on the severity of the problem.
- Set common attributes, such as extensions for user and server certificate requests.

Policy Rules

A policy rule refers to a uniquely configured instance of any policy plug-in implementation. For example, you can use the plug-in module provided for setting validity periods on certificates to configure a policy rule that forces validity periods for all client certificates issued by a Certificate Manager to fall within a predetermined range, say between 6 and 24 months. A subsystem's policy configuration can consist of one or more policy rules, each performing one or more of the following operations:

- Validate the request content by comparing it with configured criteria; reject, modify, or defer (for agent approval) the request if any of the request parameters are invalid.
- Build certificate content—for example, set common extensions and the validity period.
- Enforce organizational constraints, such as subject name, key algorithm, key size, and validity period.
- Determine whether the private key should be archived.

Keep in mind that the server applies the rules when processing end-entity requests and after agent approval (for deferred requests).

Types of Policy Rules

Certificate Management System supports distinct policy rules for each of the operations that end entities perform—certificate enrollment, renewal, and revocation, and key archival and recovery. Consequently, there are five broad categories of policies, corresponding to these types of operations:

- Enrollment policies
- Renewal policies
- Revocation policies
- Key-archival policies
- Key-recovery policies

To facilitate this classification, Certificate Management System supports a parent interface for a generic policy rule and other operation-specific interfaces that extend the parent interface. Check the CMS SDK, available in the form of Java Docs at this location: `<server_root>/cms_sdk/cms_jdk/javadocs`

For general guidelines on developing custom policy modules and adding them to the CMS policy framework, take a look at the samples installed at this location:

```
<server_root>/cms_sdk/cms_jdk/samples/policy
```

Using Predicates in Policy Rules

You can use predicates in a policy rule. A predicate indicates whether the rule that contains the predicate applies to a request. If you specify a predicate as part of the rule configuration, the policy rule applies that predicate based on request attributes to determine whether the rule is applicable for a request.

The policy predicate is a logical expression. You form the expression using variables and relational operators (**AND** or **OR**). For example, you could set up a predicate to put the CRL Distribution Point extension only in SSL client certificates, or set different validity dates for certificates for users in different groups.

The following are sample predicates:

```
HTTP_PARAMS.certType==client AND HTTP_PARAMS.ou==Engineering
```

```
HTTP_PARAMS.certType==server AND HTTP_PARAMS.o==Siroe OR
HTTP_PARAMS.certType==ca
```

Expression Support for Predicates

You form an expression using an attribute, its value, and one or more of the operators listed in Table 18-1. For a list of attributes, see “Attributes for Predicates” on page 608.

Table 18-1 Predicates in policy: supported comparison and logical operators

Operator	Description
==	Equal to
!=	Not equal to
AND	Logical operator AND
OR	Logical operator OR

Note that the expression parsing support currently supports only two comparison operators (**=**, **!=**) and two relational operators (**AND**, **OR**).

Policy expressions are formed with the following rules:

PrimitiveExpression | AndExpression | OrExpression

- PrimitiveExpression is equal to: Attribute op Value, where
Attribute can be a string
op can be any of these operators: == or !=
Value can be a string
- AndExpression is equal to: Expression AND Expression
- OrExpression is equal to: Expression OR Expression

In an expression, the AND operator takes precedence over an OR operator. For example, the expression

```
HTTP_PARAMS.certType==client AND HTTP_PARAMS.ou==Engineering OR
HTTP_PARAMS.certType==ca
```

is interpreted as

```
(HTTP_PARAMS.certType==client AND HTTP_PARAMS.ou==Engineering) OR
HTTP_PARAMS.certType==ca
```

Certificate Management System evaluates an expression based on the attributes in the request. The attributes are filled in by servlets from the HTTP input forms used for request submission. Some attributes, such as passwords typed in the form are not stored in the request. Other attributes regarding the end entity, such as the user ID, are set on the request after successful authentication. The servlets also interpret the form content, for example, retrieving the key material out of the KEYGEN or PKCS #10 information and setting the key in the certificate content. They can also set additional attributes related to the certificate content on the request. In general, you can configure which attributes—for example, sensitive attributes such as passwords—should or shouldn't be stored in the request.

Note that all data related to an end entity is gathered at the servlet level and set on the request before the request is passed to the policy subsystem. The policy subsystem applies configured policy rules on the request, determines whether the request needs agent approval, performs constraint- and extension-specific checks on the request attributes, and then formulates the certificate content by adding the appropriate information, such as the validity period and extensions.

The expression queries the request for the attributes, compares the value returned with the value provided in the predicate, and returns a boolean result.

Be aware that if the same name is in a HTTP form input and authentication token (authentication result) the authentication result can override the HTTP form input. For example, if `email` is in a HTTP input and an authentication module also puts `email` in the authentication result (that is, `authToken`) the `email` value from the authentication module will override the `email` value from the HTTP input in the request. A predicate using `email` in an expression will be evaluated to the value of the authentication instead of the HTTP input value.

The following are sample predicates:

```
HTTP_PARAMS.certType==client AND HTTP_PARAMS.ou==Engineering
HTTP_PARAMS.certType==server AND HTTP_PARAMS.o==Siroe OR
HTTP_PARAMS.certType==ca
```

Attributes for Predicates

Attributes for predicates can come from any of the following:

- Input form—that is, the HTML form that end entities use for submitting certificate requests.
- Authentication token—what the authentication subsystem returns after successfully authenticating an end entity.
- A service—for example, a Certificate Manager, Registration Manager, or Data Recovery Manager service can add certain attributes to the end-entity request.
- Policy processor—what the policy subsystem returns after subjecting the end-entity request to policy checking. For example, an extension-based policy can set an appropriate extension in the certificate.

Table 18-2 lists default attributes that are supported by various request object implementations.

Table 18-2 Attributes supported by request object implementations

Request type	Variable name	Description
Default attributes from an input form:		
Enrollment	<code>requestFormat</code>	Specifies the certificate request format. Default values include the following: <ul style="list-style-type: none"> • <code>keygen</code> • <code>pkcs10</code> • <code>clientAuth</code>

Table 18-2 Attributes supported by request object implementations *(Continued)*

Request type	Variable name	Description
Enrollment	<code>certType</code>	<p>Specifies the certificate type. Default values include the following:</p> <ul style="list-style-type: none"> • <code>ca</code> (Certificate Manager's CA signing certificate) • <code>caCrlSigning</code> (Certificate Manager's CRL signing certificate) • <code>CEP-Request</code> (router certificate) • <code>client</code> (client certificates) • <code>codeSignClient</code> (Object signing certificate - PKCS#10) • <code>objSignClient</code> (Object signing certificate - Browser) • <code>ocspResponder</code> (OCSP Responder certificate) • <code>other</code> • <code>ra</code> (Registration Manager's signing certificate) • <code>server</code> (SSL server certificate)
Enrollment	<code>doSslAuth</code>	<p>Specifies whether the client is required to do SSL client authentication during enrollment. Default values include the following:</p> <ul style="list-style-type: none"> • <code>on</code> • <code>off</code>
Enrollment	<code>certauthEnroll</code>	<p>Specifies whether it is a certificate-based enrollment. Default values include the following:</p> <ul style="list-style-type: none"> • <code>on</code> • <code>off</code>
Enrollment	<code>certauthEnrollType</code>	<p>Specifies the number of keys to be generated for a certificate-authenticated enrollment—whether a single signing key, a single encryption key, or dual keys (one for signing and another for encryption) is to be generated. Default values include the following:</p> <ul style="list-style-type: none"> • <code>single</code> • <code>encryption</code> • <code>dual</code>

Table 18-2 Attributes supported by request object implementations *(Continued)*

Request type	Variable name	Description
Enrollment	cepsubstore	Specifies the name of the CEP service; for example, <code>cep1</code> and <code>cep2</code> . When setting up multiple CEP services, you can use predicates to differentiate one service for another; see “Step 4. Set Up Multiple CEP Services” on page 844.
Enrollment, Renewal, and Revocation	requestStatus	Specifies when (or the phase in which) a request gets subjected to policy processing: <ul style="list-style-type: none"> • <code>begin</code> specifies that the request be subjected to a policy before it gets queued for agent approval. • <code>pending</code> specifies that the request be subjected to a policy after agent approval.
Renewal	requestFormat	Specifies the certificate request format. Default values include the following: <ul style="list-style-type: none"> • <code>clientAuth</code> • <code>pkcs10</code>

Default attributes from an authentication token:

(Upon successful authentication these attributes go into an enrollment request)

Enrollment	authMgrImplName	Specifies the name of the authentication plug-in module that authenticated the request.
Enrollment	authMgrInstName	Specifies the name of the authentication instance that authenticated the request.

You can define your own attributes for predicates, if there’s a need. For example, assume you have two organizational units Sales and Manufacturing and you want to issue client certificates with different validity periods to users in these two units. A quick and easy way to accomplish this would be to define a new attribute for the organizational unit, add the attribute to the enrollment form that the users in these organizational units use for certificate enrollment (so that the server receives it from the HTTP input), and use the attribute in the predicate expression for the validity constraints policy—a policy rule that determines the validity period of certificates the server issues. For details on this policy, check the “ValidityConstraints Plug-in Module” section in Chapter 3, “Constraints Policy Plug-in Modules” of *CMS Plug-Ins Guide*.

Note that to define a new attribute in any of the HTML forms, all you need to do is to add the following line to the corresponding HTML form:

```
<input type="HIDDEN" name="attribute_name" value="attribute_value">
```

Assuming that the new attribute you define for the organizational unit is `orgunit`, the line you would add to the enrollment form would be:

```
<input type="HIDDEN" name="orgunit" value="Sales">
```

To add this line to an enrollment form, you would:

1. Open the corresponding HTML file in a text editor.
2. Locate the section that lists the HTTP input variables.
3. Add this line: `<input type="HIDDEN" name="orgunit" value="Sales">`
4. Save your changes and close the file.

For the server to use the attribute (to distinguish enrollment requests from users in the Sales unit versus those in the Manufacturing unit) to issue certificates with the appropriate validity periods, you must formulate your predicate expression with the attribute you added. Here's how you do this:

1. Create a new instance of the `ValidityConstraints` policy plug-in implementation.
2. Enter the appropriate values for all the attributes. Assume you
 - o named the instance `ValidityRule1`
 - o set the minimum validity period to 10 days
 - o set the maximum validity period to 180 days
 - o defined the predicate expression as `HTTP_PARAMS.certType==client AND HTTP_PARAMS.orgunit==Sales`

(This expression specifies that the policy be applied to only client certificate requests from users in the organizational unit named Sales.)

A sample of the resulting configuration entries in the CMS configuration file would be as follows:

```
ca.Policy.rule.ValidityRule1.enable=true
ca.Policy.rule.ValidityRule1.implName=ValidityConstraints
ca.Policy.rule.ValidityRule1.maxValidity=180
ca.Policy.rule.ValidityRule1.minValidity=10
ca.Policy.rule.ValidityRule1.predicate=HTTP_PARAMS.certType==
  client AND HTTP_PARAMS.orgunit==Sales
```

Now, for setting the validity period in certificates of users who are not in the Sales organization—in this case, this would be Manufacturing—you would create another instance of `ValidityConstraints` policy rule as before with a different set values.

Assume you

- named the instance `ValidityRule1`
- set the maximum validity period to 60 days
- set the minimum validity period to 10 days
- defined the predicate expression as `HTTP_PARAMS.certType==client AND HTTP_PARAMS.orgunit!=Sales`

(This expression specifies that the policy be applied to only client certificate requests from users who are not in the organizational unit named Sales.)

A sample of the resulting configuration entries in the CMS configuration file would be as follows:

```
ca.Policy.rule.ValidityRule2.enable=true
ca.Policy.rule.ValidityRule2.implName=ValidityConstraints
ca.Policy.rule.ValidityRule2.maxValidity=60
ca.Policy.rule.ValidityRule2.minValidity=10
ca.Policy.rule.ValidityRule2.predicate=HTTP_PARAMS.certType==
  client AND HTTP_PARAMS.orgunit!=Sales
```

The new configuration would result in certificates with a validity period of six months for users in the Sales organizational unit and a validity period of three months for users in the Manufacturing unit.

Policy Processor

Each subsystem—the Certificate Manager, Registration Manager, or Data Recovery Manager—has its own policy processor. Each processor subjects an incoming request to the applicable policy rules for that subsystem.

When a subsystem starts up, its policy processor reads the current policy configurations from the configuration file, initializes them, and classifies them based on their type (see “Types of Policy Rules” on page 605). Then, when the subsystem receives an authenticated request, its request processor invokes the policy processor to apply policies on that request. The policy processor applies the rules on the request based on the request type. The policy processor also filters the rules based on predicates (see “Using Predicates in Policy Rules” on page 606).

Note that the policy processor applies only the *enabled* policy rules, in the order in which they are configured, before determining the final outcome. Each rule the processor executes returns a `PolicyResult` object. Three return values are possible:

- `PolicyResult.REJECTED` (indicates that the request failed the rule)
- `PolicyResult.DEFERRED` (indicates that the request requires agent approval)
- `PolicyResult.ACCEPTED` (indicates that the request passed the rule)

After all the policy rules are applied, the processor determines the status of the request (in this order):

1. If the request failed any policy rule (that is, if any of the policy rules returned a `PolicyResult.REJECTED` value), the processor rejects the request. The rule that rejected the request sets appropriate error messages on the request.
2. If at least one of the policy rules requires agent approval for the request (that is, if any of the policy rules returned a `PolicyResult.DEFERRED` value), the processor stores the request in the request queue for agent approval.
3. If the request passes all the policy rules (that is, all policy rules returned a `PolicyResult.ACCEPTED` value), the request gets serviced—for example the certificate is issued or renewed.

Configuring Policy Rules for a Subsystem

You can configure the main subsystems of Certificate Management System (CMS)—the Certificate Manager, Registration Manager, and Data Recovery Manager—to apply certain organizational policies on end entities' certificate enrollment, renewal, and revocation requests before servicing them. This section explains how to configure a subsystem to evaluate end-entity requests based on a set of policy rules.

The steps are as follows:

- Step 1. Before You Begin
- Step 2. Modify Existing Policy Rules
- Step 3. Delete Unwanted Policy Rules
- Step 4. Add New Policy Rules
- Step 5. Reorder Policy Rules

- Step 6. Restart the Server
- Step 7. Test Policy Configuration

For information on adding or changing policy-specific information in the configuration file, see “Changing the Configuration by Editing the Configuration File” on page 359.

Step 1. Before You Begin

Before configuring a Certificate Manager’s or Registration Manager’s policy, be sure to do this:

- Refer to the X.509 standard and PKIX standard RFC 2459 (see <http://www.ietf.org/rfc/rfc2459.txt>) to get familiar with certificate content, including extensions.
- Read Chapter 3, “Constraints Policy Plug-in Modules” and Chapter 4, “Certificate Extension Plug-in Modules” of *CMS Plug-Ins Guide*. Determine the rules that you want to use to govern the generation and formulation of certificates in your PKI setup. To locate an online version of this book, see “Where to Go for Related Information” on page 29.

This planning will help you configure a Certificate Manager and Registration Manager with the appropriate policy rules so that your end entities get the right kind of certificate.

Step 2. Modify Existing Policy Rules

You can modify a policy rule by editing its configuration parameter values; you cannot edit the name of a rule. To change the name of a rule, you need to create a new rule exactly like the rule you want to rename, except with a new name, and delete the old rule.

As a part of editing a rule, you can change its status from enabled to disabled or vice versa by checking or unchecking the `enable` parameter. A subsystem subjects certificate requests only to those rules that are enabled.

During installation, the Certificate Manager and Registration Manager automatically create a set of policy rules (that you would most likely want to use) using the policy modules registered by default. Figure 18-1 shows the policy rules created for a Certificate Manager. The Registration Manager also has a similar list. Table 18-3 summarizes the default rules created for both Certificate Manager and Registration Manager.

After installation, you must verify whether you want to use these rules, check how these rules are configured, and make the appropriate configuration changes. Keep in mind some of these policy rules are essential for the server to process requests. For example, the server won't be able to process certificate-issuance requests if `DefaultValidityRule` is disabled. Similarly, the server won't be able to process certificate-renewal requests if `DefaultRenewalValidityRule` is disabled.

If you don't want to use a rule, delete it from the configuration as explained in “Step 3. Delete Unwanted Policy Rules” on page 618; alternatively, you may keep it in the disabled state. If you want to create a new rule, you can do so as explained “Step 4. Add New Policy Rules” on page 618.

Figure 18-1 Default policy rules created for a Certificate Manager

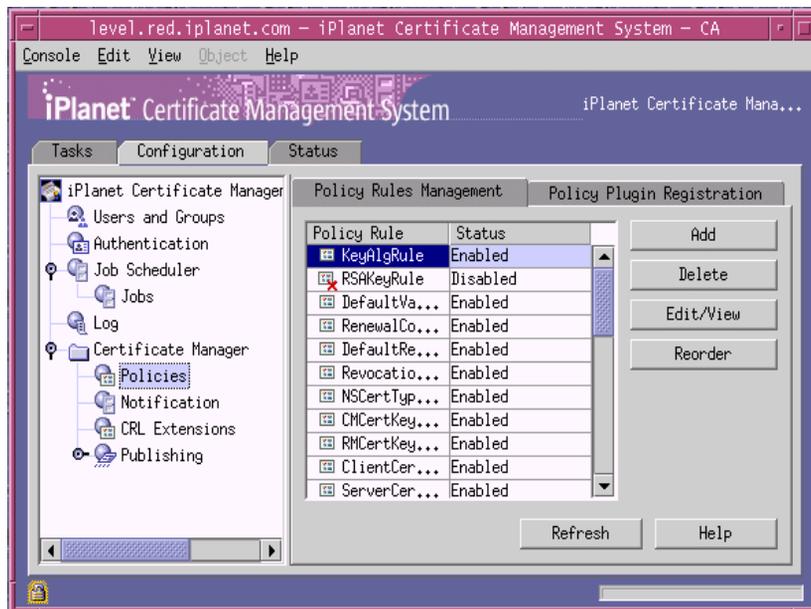


Table 18-3 Default policy rules of a Certificate Manager and Registration Manager

Policy rule name	Certificate Manager	Registration Manager
KeyAlgRule	Yes	Yes
DSAKeyRule	Yes	Yes
RSAKeyRule	Yes	Yes
DefaultValidityRule	Yes	Yes

Table 18-3 Default policy rules of a Certificate Manager and Registration Manager

Policy rule name	Certificate Manager	Registration Manager
RenewalConstranitsRule	Yes	Yes
DefaultRenewalValidityRule	Yes	Yes
RevocationConstranitsRule	Yes	Yes
NSCertTypeExt	Yes	Yes
CMCertKeyUsageExt	Yes	Yes
RCMertKeyUsageExt	Yes	Yes
ClientCertKeyUsageExt	Yes	Yes
ServerCertKeyUsageExt	Yes	Yes
ObjSignCertKeyUsageExt	Yes	Yes
CRLSignCertKeyUsageExt	Yes	Yes
SubjectKeyIdentifierExt	Yes	Yes
CertificatePoliciesExt	Yes	Yes
NSCCommentExt	Yes	Yes
OCSPNoCheckExt	No	No
OCSPSigningExt	Yes	Yes
CODESigningExt	Yes	Yes
GenericASN1Ext	Yes	Yes
CRLDistributionPointsExt	Yes	Yes
SubjectAltNameExt	Yes	Yes
SigningAlgRule	Yes	No
AuthorityKeyIdentifierExt	Yes	No
AuthInfoAccess	Yes	Yes
BasicConstraintsExt	Yes	No
UniqueSubjectName	Yes	No
NameConstraintsExt	Yes	No
PolicyConstraintsExt	Yes	No
SubCANameCheck	Yes	No
PolicyMappingsExt	Yes	No

Table 18-3 Default policy rules of a Certificate Manager and Registration Manager

Policy rule name	Certificate Manager	Registration Manager
IssuerRule	Yes	No

To modify a policy rule in the CMS configuration:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select the subsystem to which the policy rule you want to modify belongs.
4. Select Policies.

The Policy Rules Management tab appears. It lists configured policy rules.

5. In the Policy Rule list, select a rule that you want to modify.

For the purposes of this instruction, assume that you selected the rule named `DefaultValidityRule`.

6. Click Edit/View.

The Policy Rule Editor window appears, showing how this rule is configured. An example is shown below.

Policy Rule Editor

Policy Rule ID: DefaultValidityRule
Policy Plugin ID: ValidityConstraints

enable

predicate

minValidity

maxValidity

leadTime

lagTime

notBeforeSkew

Ensures that the user's requested validity period is acceptable. If not specified, as is usually the case, this policy will set the validity. See RFC 2459.

OK Cancel Help

7. Make the necessary changes and click OK.
You are returned to the Policy Rules Management tab.
8. Repeat steps 5 through 7 for the remaining rules.
9. Click Refresh.

Step 3. Delete Unwanted Policy Rules

You can delete any unwanted policy rules from the CMS configuration. If you think you might need a rule in the future, instead of deleting it from the configuration you should disable it by unchecking the `enable` parameter. In this way, you can avoid re-creating the rule in the future. Because the subsystems subject end-entity requests only to rules that are currently enabled (see “Policy Processor” on page 612), keeping unwanted rules in the disabled state in the configuration does not affect policy decisions made by a subsystem.

To delete a policy rule from the CMS configuration:

1. In the Policy Rules Management tab, select the rule you want to delete and click Delete.
2. When prompted, confirm the delete action.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. Don't restart the server yet; you can do so after you've made all the required changes.

Step 4. Add New Policy Rules

Adding a policy rule to the CMS configuration involves creating a new instance of an already registered policy plug-in module, assigning a unique name for the instance, and entering appropriate values for the parameters that define the module you want to create an instance of.

When you add a policy rule, the CMS configuration gets updated with policy-specific information. Keep the following points in mind:

- When naming a policy instance (or rule), be sure to formulate the name using any combination of letters (aA to zZ), digits (0 to 9), an underscore (`_`), and a hyphen (`-`); other characters and spaces are not allowed. For example, you can type `My_Policy_Rule` or `MyPolicyRule` as the instance name, but not `My Policy Rule`.

- The status of the rule, enabled or disabled, depends on whether you check or uncheck the `enable` parameter. A subsystem subjects certificate requests only to rules that are enabled.
- The server does not automatically reorder rules. Be sure to change the order of the rule, if required.

Figure 18-2 shows the policy modules registered with a Certificate Manager. The Registration Manager also has a similar list. Table 18-4 summarizes the default modules registered with both Certificate Manager and Registration Manager.

Figure 18-2 Default policy modules registered with a Certificate Manager

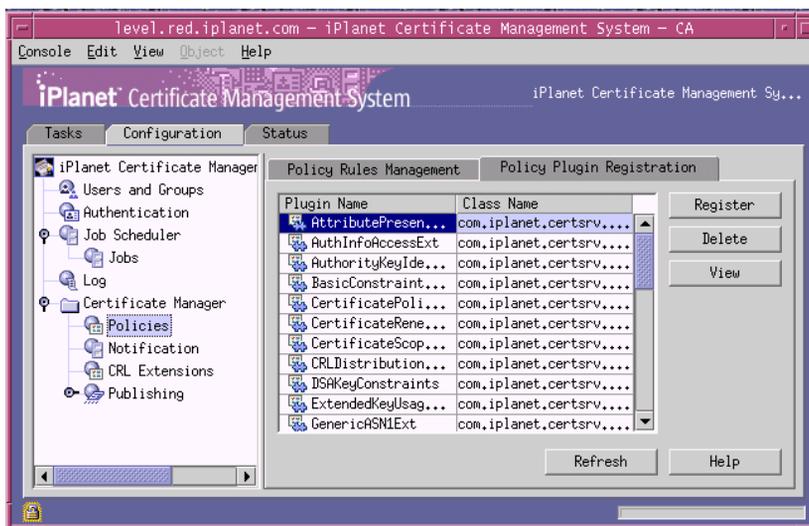


Table 18-4 Policy modules of a Certificate Manager and Registration Manager

Policy plug-in module name	Certificate Manager	Registration Manager
AttributePresentConstraints	Yes	Yes
AuthInfoAccessExt	Yes	Yes
AuthorityKeyIdentifierExt	Yes	No
BasicConstraintsExt	Yes	No
CertificatePoliciesExt	Yes	Yes
CertificateRenewalWindowExt	Yes	Yes
CertificateScopeOfUseExt	Yes	Yes

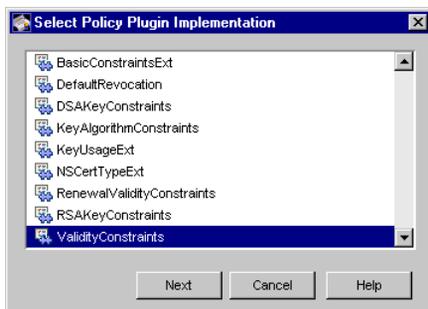
Table 18-4 Policy modules of a Certificate Manager and Registration Manager *(Continued)*

Policy plug-in module name	Certificate Manager	Registration Manager
CRLDistributionPointsExt	Yes	Yes
DSAKeyConstraints	Yes	Yes
ExtendedKeyUsageExt	Yes	Yes
GenericASN1Ext	Yes	Yes
IssuerAltNameExt	Yes	Yes
IssuerConstraints	Yes	No
KeyAlgorithmConstraints	Yes	Yes
KeyUsageExt	Yes	Yes
NameConstraintsExt	Yes	No
NSCComment	Yes	Yes
NSCertTypeExt	Yes	Yes
OCSPNoCheckExt	Yes	Yes
PolicyConstraintExt	Yes	No
PolicyMappingsExt	Yes	No
PrivateKeyUsagePeriodExt	Yes	Yes
RemoveBasicConstraintsExt	Yes	No
RenewalConstraints	Yes	Yes
RenewalValidityConstraints	Yes	Yes
RevocationConstraints	Yes	Yes
RSAKeyConstraints	Yes	Yes
SigningAlgorithmConstraints	Yes	Yes
SubCANameConstraints	Yes	No
SubjectAltNameExt	Yes	Yes
SubjectDirectoryAttributesExt	Yes	Yes
SubjectKeyIdentifierExt	Yes	Yes
UniqueSubjectNameConstraints	Yes	No
ValidityConstraints	Yes	Yes

To add a new policy rule to the CMS configuration:

1. In the Policy Rules Management tab, click Add.

The Select Policy Plugin Implementation window appears. It lists registered policy plug-in modules. If you have registered any custom policy modules (see “Registering a Policy Module” on page 626), they too will be listed here.

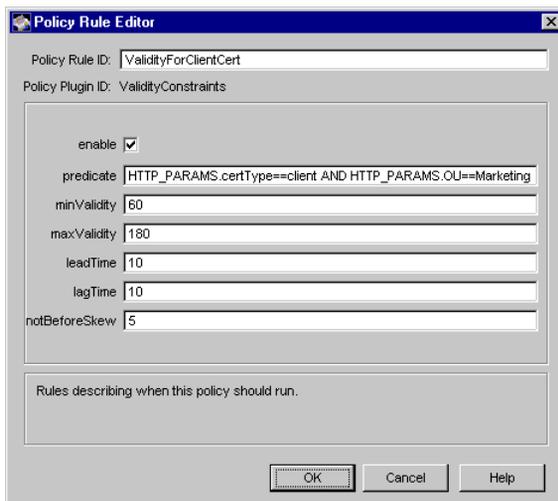


2. Select a plug-in module.

For the purposes of this instruction, assume that you selected the `ValidityConstraints` module.

3. Click Next.

The Policy Rule Editor window appears, listing the configuration information.



4. Enter the appropriate information.

Policy Rule ID. Type a unique name that will help you identify the rule; be sure to use an alphanumeric string without spaces.

enable. Check the box to enable the rule (default). If you enable the rule and set the remaining parameters correctly, the server sets the configured validity period in certificates specified by the `predicate` parameter. Uncheck the box to disable the rule. If you disable the rule, the server does not set the configured validity period in certificates; it sets the validity period to the one specified in the request.

predicate. Type the predicate expression for this rule. If you want this rule to be applied to all certificate requests, leave the field blank (default). To form a predicate expression, see “Using Predicates in Policy Rules” on page 606.

minValidity. Type the minimum validity period, in days, for certificates. The value must be an integer greater than zero and less than the value you will type for the `maxValidity` parameter next. The default value is 180 days.

maxValidity. Type the maximum validity period, in days, for certificates. The value must be an integer greater than zero and also greater than the value you typed for the `minValidity` parameter. The default value is 730 days.

leadTime. Type the lead time, in minutes, for certificates. For a certificate renewal request to pass the renewal validity constraints policy, the value of the `notBefore` attribute in the certificate request must not be more than value of the `leadTime` parameter in the future, relative to the time when the policy rule is run. The default value is 10 minutes.

The `notBefore` attribute value specifies the date on which the certificate validity begins.

lagTime. Type the lag time, in minutes, for certificates. For a certificate renewal request to pass the renewal validity constraints policy, the value of the `notBefore` attribute in the certificate request must not be more than the value of the `lagTime` in the past, relative to the time when the policy is run. The default value is 10 minutes.

The `notBefore` attribute value specifies the date on which the certificate validity ends.

notBeforeSkew. Type the number of minutes to subtract from the current time when creating the value for the certificate’s `notBefore` attribute. It can help some clients with incorrectly set clocks use the new certificate after downloading. For example, if the certificate is issued at 11:30 a.m. and the clock settings of the client into which the certificate is downloaded is 11:20 a.m., the

certificate cannot be used for 10 minutes. Setting the value of the `notBeforeSkew` parameter to 10 minutes would adjust the value of the `notBefore` parameter to 11:20 a.m.—thus making the certificate usable following the download. The default value is 5 minutes.

5. Click OK.

You are returned to the Policy Rules Management tab.

6. Repeat steps 1 through 5 and create additional rules, if required.

Step 5. Reorder Policy Rules

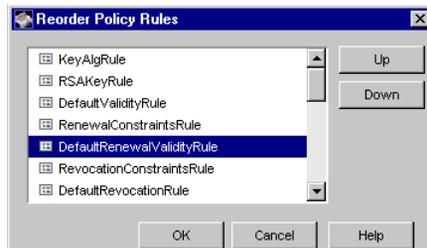
For maintaining priority levels, Certificate Management System supports a linear list of policy rules in increasing order of priority. This means that for a given policy category in the configuration file, a policy configuration with a lower priority precedes one with a higher priority. This simple linear listing avoids the need to have explicit locking on request attributes to prevent conflicting changes. By ordering the rules, you introduce a concurrency control whereby a higher-priority rule configuration overwrites any changes made by a lower-priority rule configuration that precedes it.

You may want to specify policies at different priority levels for the same operation depending on the end-entity information. For example, authentication policies, if any, need to precede others in the list.

To reorder policy rules in the CMS configuration:

1. In the Policy Rules Management tab, click Reorder.

The Reorder Policy Rules window appears. It lists configured policy rules in the order in which they are executed by the subsystem.



2. To change the order of a rule, select it in the list and click the Up or Down button, as appropriate.

Keep in mind that the server executes the rules on a first-come-first-served basis, overwriting the configuration determined by the previous rule, if any.

3. When you have the correct order, click OK.

You are returned to the Policy Rules Management tab.

4. To view the updated configuration, click Refresh.

Step 6. Restart the Server

This step is required only if you were prompted to restart the server in any of the previous steps.

To restart the server from the CMS window:

1. Click the Tasks tab
2. Click Restart the Server.

Step 7. Test Policy Configuration

To make sure that you've configured the server correctly, request a certificate and check the certificate for details such as for validity period, key type and size, and extensions.

- Step A. Enroll for a Certificate
- Step B. Approve the Request
- Step C. Check the Certificate Details

Step A. Enroll for a Certificate

The steps outlined below explain how to request a personal certificate from the Certificate Manager using the manual enrollment method. If you've configured the Certificate Manager for automated certificate issuance, for example the directory-based enrollment, you can use the appropriate form and request a certificate.

To request a client or personal certificate from the Certificate Manager:

1. Open a web browser window.
2. Go to the End Entity Services interface of the Certificate Manager you configured (or the Registration Manager that's connected to this Certificate Manager).

The URL is in this form: `https://<hostname>:<end_entity_HTTPS_port>` or `http://<hostname>:<end_entity_HTTP_port>`

3. In the left frame, under **Browser**, click **Manual**.

This opens the manual enrollment form.

4. Fill in all the values and submit the request.

The client prompts you to enter the password for your key database.

5. When you enter the correct password, the client generates the key pairs.

Do not interrupt the key-generation process.

Step B. Approve the Request

This step is required if you used the manual enrollment form for requesting the certificate. The request you submitted is waiting in the agent queue for approval by an agent.

To approve the request:

1. Go to the Certificate Manager's Agent Services interface.

The URL is in this format: `https://<hostname>:<agent_port>`

2. In the left frame, click the link that says **List Requests**.
3. In the form that appears, select the "Show pending requests" option and click **Find**.

You should see your request in the list of pending requests.

4. Locate the request you submitted and approve the request.

You should see a confirmation page indicating that the certificate has been issued. Don't close the page until you finish the next step.

Step C. Check the Certificate Details

Verify that the certificate contains the required details. Be sure to check the **Extension** section to see if it contains all the required extensions.

Using JavaScript for Policies

Certificate Management System includes a facility for complex scripting of the policy plug-in instances via JavaScript™. Using the JavaScript policy processor allows you to:

- Determine the call sequence of existing Java plug-ins
- Use complex logic to determine whether to call a plug-in (versus the simpler predicates)
- Write policies in JavaScript
- Develop extensions without needing to know Java or have the Java SDK

Certificate Management System uses the Rhino JavaScript engine from [Mozilla.org](http://www.mozilla.org). You can get more details about the Rhino project from this site:

<http://www.mozilla.org/rhino>

To learn more about how to use JavaScript in Certificate Management System, consult the sample `policy.js` file included in the distribution:

```
<server_root>/bin/cert/js/policy.js
```

Managing Policy Plug-in Modules

This section explains how to use the CMS window to perform the following operations:

- Registering a Policy Module
- Deleting a Policy Module

For information on adding or changing policy-specific information in the configuration file, see “Changing the Configuration by Editing the Configuration File” on page 359.

Registering a Policy Module

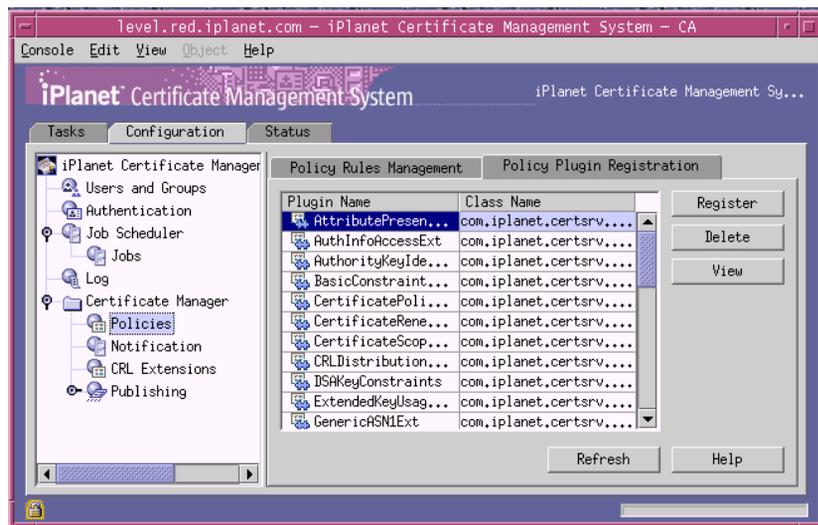
You can register new policy plug-in modules in a subsystem’s policy framework. Registering a new policy module involves specifying the name of the module and the full name of the Java class that implements the policy interface. For example, you can add a policy implementation, named as follows, to the Data Recovery Manager’s policy framework: `com.iplanet.policy.KeyArchivalPolicy`

Before registering a plug-in module, be sure to put the Java class for the module in the `classes` directory (the implementation must be on the class path).

To register a policy module in a subsystem's policy framework:

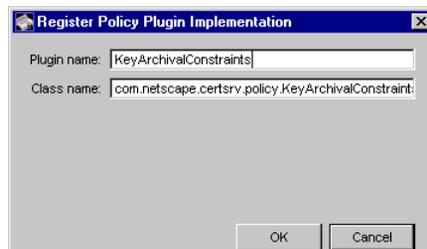
1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select the subsystem that will use the module you want to register.
4. Select Policies, and then in the right pane, select the Policy Plugin Registration tab.

The Policy Plugin Registration tab appears. It lists registered policy plug-in modules.



5. Click Register.

The Register Policy Plugin Implementation window appears.



6. Specify information as appropriate:

Plugin name. Type a name for the plug-in module.

Class name. Type the full name of the class for this module—that is, the path to the implementing Java class. If this class is part of a package, be sure to include the package name. For example, if you are registering a class named `myPolicy` and if this class is in a package named `com.myCompany`, type `com.myCompany.myPolicy`.

7. Click OK.

You are returned to the Policy Plugin Registration tab.

8. To view the updated configuration, click Refresh.

Deleting a Policy Module

You can delete unwanted policy plug-in modules using the CMS window. Before deleting a module, be sure to delete all the policy rules that are based on this module; see “Step 3. Delete Unwanted Policy Rules” on page 618.

To delete a policy module from a subsystem’s policy framework:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select the subsystem that registers the module you want to delete.
4. Select Policies, and then in the right pane, select the Policy Plugin Registration tab.

The Policy Plugin Registration tab appears. It lists registered policy modules.

5. In the Plugin Name list, select the module you want to delete and click Delete.
6. When prompted, confirm the delete action.

Setting Up LDAP Publishing

iPlanet Certificate Management Server (CMS) provides a customizable publishing framework for the Certificate Manager, enabling it to publish certificates, certificate revocation lists (CRLs), and other certificate-related objects to any of the supported repositories—an LDAP-compliant directory, a flat file, and an online validation authority—using the appropriate protocol.

The ability of a Certificate Manager to publish certificates, CRLs, and other certificate-related objects to a directory using the LDAP or LDAPS protocol is called *LDAP publishing* and the directory to which it publishes is called the *publishing directory*.

This chapter explains how to configure the Certificate Manager to publish certificates and CRLs to an LDAP directory. The chapter also tells you how to update the directory manually, if the need arises.

The chapter has the following sections:

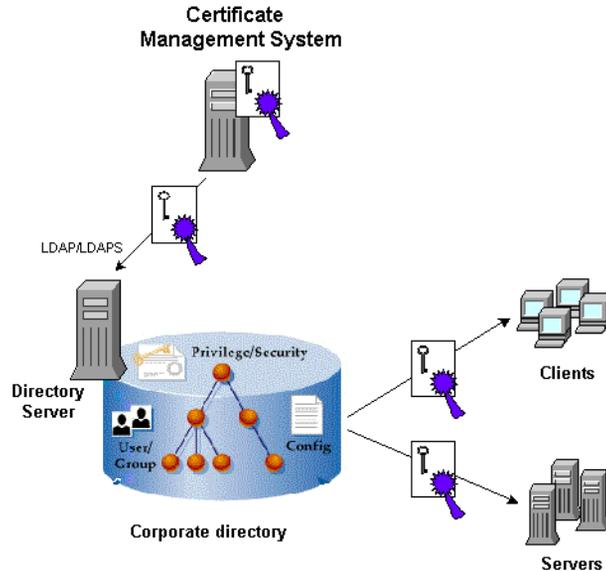
- Publishing of Certificates to a Directory (page 629)
- Publishing of CRLs (page 634)
- Configuring a Certificate Manager to Publish Certificates and CRLs (page 639)
- Manually Updating Certificates and CRLs in a Directory (page 686)

Publishing of Certificates to a Directory

Large corporations typically use Lightweight Directory Access Protocol (LDAP) directories, such as iPlanet Directory Server, to store and manage corporatewide data, including user and group information and network resource data. If you have deployed an LDAP-compliant directory, you can configure the Certificate Manager to automatically publish your CA and end-entity certificate-related information to

that directory. For example, if you have configured the Certificate Management System to employ directory-based authentication, you should consider publishing the CA and end-entity certificates to the same directory. This way, you can keep your users' security credentials with the rest of the user information (see Figure 19-1).

Figure 19-1 Publishing certificates to a directory for distribution



Note that configuring the Certificate Manager for LDAP publishing is optional—you can turn this feature off without affecting any of the certificate issuance, renewal, and revocation operations handled by the server.

You can configure the Certificate Manager to automatically publish certificates to the directory every time a certificate is issued and at a predetermined interval—for example, every day or once every week. Privileged users (administrators and agents) can also manually initiate the LDAP publishing process.

Figure 19-2 illustrates LDAP publishing by the Certificate Manager when a certificate requested via the manual-enrollment process is issued.

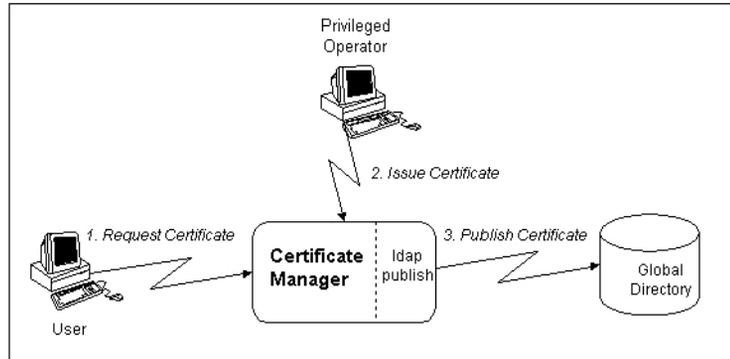
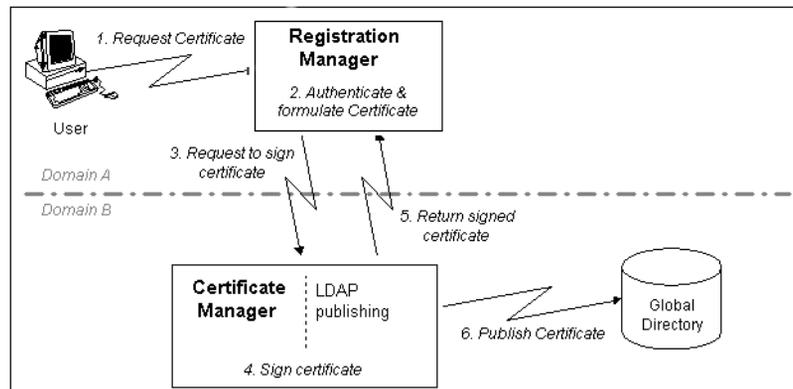
Figure 19-2 Publishing by a Certificate Manager

Figure 19-3 illustrates how certificates requested via a Registration Manager get published to the directory.

Figure 19-3 Publishing of certificates requested via a Registration Manager

Timing of Directory Updates

If the LDAP directory is properly configured to work with the Certificate Manager (and vice versa), any changes to the certificate information in the Certificate Manager are automatically made also in the publishing directory.

The publishing directory is updated at these times:

- When the Certificate Manager starts up, it publishes its *CA signing certificate* to the directory.
- When the Certificate Manager issues a new certificate (the request may originate from Registration Managers that're connected to the Certificate Manager), it stores a copy of the certificate in its internal database and then publishes the certificate to the configured directory.
- When the Certificate Manager revokes a certificate (the request may originate from Registration Managers that're connected to the Certificate Manager), it marks the copy of the certificate in its internal database as revoked and then unpublishes or removes the revoked certificate from the configured directory.
- When a certificate expires, the Certificate Manager can remove that certificate from the configured directory. Note that the server doesn't do this automatically. You need to configure the server to run the appropriate job. For details, see "Configuring a Subsystem to Run Automated Jobs" on page 589.
- When the certificate revocation list is created or updated (either through the CMS window or through the certificate-revocation feature provided in the agent or end-entity interface), the Certificate Manager publishes that list to the configured directory.

Table 19-1 summarizes the above-listed actions of the Certificate Manager. The table also indicates how the Certificate Manager populates an LDAP directory, if configured for publishing. Note that certificates (and CRLs) are published as DER-encoded binary blobs.

Table 19-1 Details of objects published by the Certificate Manager

Object	Action and Timing	LDAP entry	LDAP attribute
End-entity certificate	Publishing occurs when a certificate is issued or renewed	End-entity's entry	<code>userCertificate;binary</code>
	Unpublishing (removal) occurs when a certificate is revoked or expired	End-entity's entry	<code>userCertificate;binary</code>
CA certificate	Publishing occurs when the Certificate Manager is started	CA's entry	<code>caCertificate;binary</code>
CRL (full)	Publishing (replacement) occurs when a new CRL is generated	CA's entry	<code>certificateRevocationList;binary</code>

The Certificate Manager cannot update the directory in the following cases:

- If an end-entity entry is not present or if an entry cannot be found to publish the certificate.
- If the directory's schema doesn't include the appropriate attributes. To configure the directory for LDAP publishing, see "Step 2. Set Up the Directory for Publishing" on page 641. Note that the Certificate Manager publishes to the `userCertificate;binary` attribute, which is an LDAP v3 standard. Unless you are using a non-standards compliant directory, this situation shouldn't arise.
- When the directory is unreachable because maintenance work is being performed, or because of network or system failures.

Note that the Certificate Manager's LDAP publishing action happens as a separate transaction from any certificate operation (such as issuance); the operation of a certificate is not affected by whether it was successfully published or not.

Directory Update Process

As indicated in Table 19-1 on page 632, when a Certificate Manager is requested to issue a certificate, update certificate information, or publish a CRL, it automatically updates the corresponding entry in the configured directory with relevant information. To locate the correct directory entry, the Certificate Manager relies on object-mapping rules, which can be defined using the mapper modules. Once an entry is located in the directory, to publish the object to the correct attribute of the located entry, the Certificate Manager relies on object-publishing rules, which can be defined with the help of publisher modules. For details about mapper and publisher modules, see Chapter 5, "Mapper Plug-in Modules" and Chapter 6, "Publisher Plug-in Modules" of *CMS Plug-Ins Guide*.

Similarly, when you revoke a certificate, the Certificate Manager uses the object mapping and publishing rules to locate and delete the corresponding certificate from the directory.

For step-by-step instructions to configure a Certificate Manager to publish to an LDAP directory, see "Configuring a Certificate Manager to Publish Certificates and CRLs" on page 639.

Directory Synchronization

The Certificate Manager and the publishing directory can become out of sync if certificates are issued or revoked while Directory Server is down. Certificates that were issued or revoked need to be published or unpublished manually when Directory Server comes back up.

To help find certificates that are out of sync with the directory—that is, valid certificates that are not in the directory and revoked or expired certificates that are still in the directory—the Certificate Manager keeps a record of whether a certificate in its internal database has been published to the directory. If the Certificate Manager and the publishing directory become out of sync, you can use the Update Directory option in the Certificate Manager Agent Services interface to synchronize the publishing directory with the internal database.

The following choices are available for synchronizing the directory with the internal database:

- Search the internal database for certificates that are out of sync and publish or unpublish accordingly.
- Publish certificates that were issued from time A to time B while Directory Server was down. Similarly, unpublish certificates that were revoked or that expired while Directory Server was down.
- Publish or unpublish a range of certificates based on serial numbers (from serial number *xx* to serial number *yy*).

For instructions, see “Manually Updating Certificates in the Directory” on page 687.

Publishing of CRLs

This section covers the following topics:

- What’s a CRL?
- Reasons for Revoking a Certificate
- Revocation Checking by Netscape Clients
- Revocation Checking by iPlanet Servers
- Publishing of CRLs to an LDAP Directory
- CRL Issuing Points

What's a CRL?

Server and client applications that use public-key certificates as tokens of identification need access to information about the validity of a certificate; because one of the factors that determines the validity of a certificate is its revocation status, these applications need to know whether the certificate being validated has been revoked. In that regard, the CA has a responsibility to do the following:

- Revoke the certificate if any of the certificate assertions becomes false—for example, if the subject's key gets compromised or the status of the subject's role or right changes. (See “Reasons for Revoking a Certificate” on page 636.)
- Make the revoked certificate available to parties or applications that need to verify its validity status.

One of the standard methods for conveying the revocation status of certificates is by publishing a list of revoked certificates. This list is known as the *certificate revocation list* (CRL). The CRL is a publicly available list of certificates that have been revoked.

A CRL is issued and digitally signed by the certificate authority (CA) that issued the certificates listed in the CRL. The CA may use a single key pair to sign the certificates and CRLs or two separate key pairs, one for signing certificates and another one for signing CRLs. The CA's function includes creating the CRLs periodically and distributing them to other applications. For example, the CA may publish the CRL to a global directory which other applications may use for checking the revocation status of a certificate or from which other applications can retrieve the CRL.

In Certificate Management System, the Certificate Manager can create the CRL, sign it, and publish it to any of the configured repositories, such as an LDAP directory, a file, and an OCSP responder. Configuring a Certificate Manager to publish CRLs is optional. Note that the Registration Manager cannot create or publish the CRL.

By default, the Certificate Manager uses a single key pair for signing the certificates it issues and CRLs it generates. This key pair and the corresponding certificate is explained in “CA Signing Key Pair and Certificate” on page 451. You may choose to create another key pair for the Certificate Manager and use it exclusively for signing the CRLs it generates. For details, see “CRL Signing Key Pair and Certificate” on page 453.

Normally, whenever a certificate is revoked (by administrators, agents, or end users), the Certificate Manager automatically updates the status of the certificate in its internal database—it marks the copy of the certificate in its internal database as revoked and removes the revoked certificate from the directory, if the Certificate

Manager is configured to do so. In addition to certificates, the Certificate Manager also maintains a CRL in its internal database. You can configure the Certificate Manager to generate the CRL every time a certificate is revoked and at periodic intervals.

You can also configure the Certificate Manager to generate and publish CRLs conforming to X.509 (either version 1 or version 2) standards by enabling or disabling the CRL extension-specific modules in the server's configuration. Note that the server supports standard CRL extensions that are explained in Chapter 7, "CRL Extension Plug-in Modules" of *CMS Plug-Ins Guide*.

For instructions on how to configure a Certificate Manager to publish CRLs, see "Configuring a Certificate Manager to Publish Certificates and CRLs" on page 639.

Reasons for Revoking a Certificate

A Certificate Manager can revoke any certificate it has issued. A certificate needs to be revoked if one or more of the following situations occur:

- The owner of the certificate has changed status and no longer has the right to use the certificate.
- The private key of a certificate owner has been compromised.
- The certificate owner doesn't want to use the certificate.
- The private key of the CA that issued the certificate has been compromised.

A certificate can be revoked by administrators, agents, and end entities, such as end users and individual server administrators. Agents and administrators (with agent privileges) can revoke certificates by using the forms provided in the agent interface. Administrators, agents, and end users can revoke certificates by using the forms provided in the Revocation tab of the end-entity interface. Note that end users can revoke only their own certificates, whereas agents and administrators can revoke any certificates issued by the server. End users are also required to authenticate to the server in order to revoke their certificate; see "Authentication of End Users During Certificate Revocation" on page 540.

Whenever a certificate is revoked, the Certificate Manager updates the status of the certificate in its internal database. This way, the server keeps track of all revoked certificates in its internal database and it makes the revoked list of certificates public (by publishing it to a central repository) to notify other users that the certificates in the list are no longer valid.

Revocation Checking by Netscape Clients

At the time of this writing, Netscape Communicator versions 4.7 and later, when used in conjunction with the security module called Netscape Personal Security Manager, enable automatic revocation-status verification of certificates using the OCSP protocol. Chapter 21, “Setting Up an OCSP Responder” explains how the revocation status of a certificate is verified in an OCSP-compliant PKI setup.

Earlier versions of Netscape client products do not have the ability to automatically check to see whether a certificate has been revoked. However, these clients do give the user the ability to check the revocation status of a certificate if it includes the `NetscapeRevocationURL` extension. For details about this extension, check this site: <http://home.netscape.com/eng/security/cert-exts.html>

In addition, from the Retrieval tab of the CMS end-entity interface, Netscape client users can manually check the revocation status of a particular certificate and automatically import the latest version of the CRL into their browsers. If your users are not using Netscape clients, they can download the latest CRL in binary form to a local file, and then import this file into their browsers by an appropriate method. Users can also view the header information of the master or full CRL published by the Certificate Manager, which contains the date and time of the latest update, and then compare this information to that in their browser’s CRL to see if they have the latest version.

Revocation Checking by iPlanet Servers

Because iPlanet servers currently cannot check the revocation status of a certificate, you should use other forms of access control. For example, you can remove individual users from access groups to prevent them from accessing the server.

Because Certificate Management System can check the revocation status of the certificates that it issues, you do not need to rely on other forms of access control.

Publishing of CRLs to an LDAP Directory

The Certificate Manager can publish the CRL to an LDAP-compliant directory using the LDAP protocol or LDAP over SSL (LDAPS) protocol, and applications can retrieve the CRL over HTTP. Support for retrieving CRLs over HTTP enables some browsers, such as Netscape Communicator, to automatically import the latest CRL from the directory that receives regular updates from the Certificate Manager. The browser can then use the CRL to automatically check all certificates to ensure that they have not been revoked.

For applications that are incapable of retrieving the CRL over HTTP, the Certificate Manager also supports retrieval of the CRL in binary form. For example, if the browser you've deployed doesn't support CRL retrieval over HTTP, your users may download the CRL to a local file and then import the file into their browsers by an appropriate method.

You can configure a Certificate Manager to publish the CRL it maintains to a directory, for example, to the same directory in which end-entity certificates are published. If you configure the Certificate Manager and directory to work properly, any changes to the CRL information in the Certificate Manager are automatically updated in the publishing directory. Note that the server publishes the CRL to the `certificateRevocationList;binary` attribute of the CA's entry in the directory. To locate the correct directory entry, the Certificate Manager uses object mapping rules; to publish the CRL to the correct attribute of the located entry, the server uses publishing rules. For details about mapper and publisher rules, see Chapter 5, "Mapper Plug-in Modules" and Chapter 6, "Publisher Plug-in Modules" of *CMS Plug-Ins Guide*.

Directory updates take place depending on how you configure the Certificate Manager—that is, publish the CRL to the directory every time a certificate is revoked or at specific intervals, or both. It's important to understand that when the Certificate Manager revokes a certificate, it marks the copy of the certificate in its internal database as *revoked*, generates the CRL, and then publishes it to the configured directory. For example, if you configure the server to publish the CRL every time a certificate is revoked, CRL will be generated whenever a certificate is revoked.

For instructions on configuring a Certificate Manager for publishing CRLs to a directory, see "Configuring a Certificate Manager to Publish Certificates and CRLs" on page 639.

If the Certificate Manager and publishing directory become out of sync for some reason, privileged users (administrators and agents) can also manually initiate the publishing process. For instructions, see "Manually Updating the CRL in the Directory" on page 688.

CRL Issuing Points

Because CRLs can grow very large, several methods have been developed to minimize the overhead of retrieving and delivering large CRLs. One of these methods is based on partitioning the entire certificate space and associating a separate CRL with every partition. This partition is called a *CRL issuing or distribution point*—it is the location where a subset of all the revoked certificates are

maintained. Partitioning can be based on revocation reason, on whether the revoked certificate is a CA certificate or end-entity certificate, on end users' names, and so on. Each issuing point is identified by a set of names, which can be in various forms.

Once the issuing points have been defined, they can be included in certificates so that an application that needs to check the revocation status of a certificate can access the CRL issuing points specified in the certificate instead of the master or main CRL—the application would check the CRL maintained at the issuing point, which would be smaller in size compared to the master CRL, and thus speed up the revocation-status-checking process.

CRL distribution points can be associated with certificates by setting the `CRLDistributionPoint` extension in them.

By default, the Certificate Manager only generates and publishes a single CRL, identified as the *master CRL*. However, for interoperability purposes, the server does enable you to add the `CRLDistributionPoint` extension to the certificates it issues. For details, see section “`CRLDistributionPointsExt` Plug-in Module” in Chapter 4, “Certificate Extension Plug-in Modules” of *CMS Plug-Ins Guide*.

Configuring a Certificate Manager to Publish Certificates and CRLs

If you are using an LDAP-compliant directory, such as iPlanet Directory Server, to publish and manage your user and group data, you can configure the Certificate Manager to communicate with this directory. The Certificate Manager can then publish end-entity as well as CA certificates and the certificate revocation list (CRL) to the directory. This way, your publishing directory acts as a common distribution point for information about users and other entities on the network, including each entity's current security credentials.

Once the Certificate Manager is configured to publish to the directory, many certificate and CRL-related operations are performed automatically. For details, see “Timing of Directory Updates” on page 631.

To configure a Certificate Manager to publish certificates and CRLs to a directory, follow these steps:

- Step 1. Before You Begin
- Step 2. Set Up the Directory for Publishing
- Step 3. Configure the Certificate Manager to Publish Certificates

- Step 4. Configure the Certificate Manager to Publish CRLs
- Step 5. Identify the Publishing Directory
- Step 6. Test Certificate and CRL Publishing (optional)

Step 1. Before You Begin

Before configuring a Certificate Manager to publish its CA certificate, end-entity certificates, and CRLs to a directory, do this:

- Read “Publishing of Certificates to a Directory” on page 629 and “Publishing of CRLs to an LDAP Directory” on page 637 to understand how the Certificate Manager publishes certificates and CRLs to the directory.
- Read Chapter 5, “Mapper Plug-in Modules” and Chapter 6, “Publisher Plug-in Modules” of *CMS Plug-Ins Guide*. Be sure to take a look at the default mappers and publishers created during CMS installation and determine whether they are suitable for your setup. If they’re unsuitable, decide on the mapper and publisher modules you want to use.
- If you decided to not use the default mappers created using the `LdapCaSimpleMap` module, you will be required to manually create an entry for the CA in the publishing directory. (This document explains how to create an entry for the CA in Netscape Directory Server, version 4.x only.)
- Read “Publishing of CRLs” on page 634. Determine whether you want the Certificate Manager to publish version 1 or version 2 CRLs to the directory. If you decide to publish version 2 CRLs, read Chapter 4, “Certificate Extension Plug-in Modules” of *CMS Plug-Ins Guide* and determine the CRL extensions you want the Certificate Manager to set; you will be required to configure the server to set these extensions.
- Identify your publishing directory. If you’ve already configured the Certificate Manager to use an LDAP directory for authenticating users (for example, if you’re using the directory-based or directory- and PIN-based authentication), you should consider publishing certificates and CRLs to the same directory. This way, users’ security credentials will be kept with the rest of the user information.
- Note the following information for the directory: the host name, the port number, and the port type—whether it’s an SSL or nonSSL port.

- Determine how you want the Certificate Manager to authenticate to the directory: whether to publish with basic authentication, publish over SSL without SSL client authentication, or publish over SSL with SSL client authentication. Accordingly, you will need to configure the Directory Server.
- If you want the Certificate Manager to authenticate to the directory using SSL client authentication, determine the certificate the Certificate Manager must use for SSL client authentication; see “Certificate Manager’s Key Pairs and Certificates” on page 451. By default, the server uses its SSL server certificate; see “SSL Server Key Pair and Certificate” on page 455.

Depending on your PKI setup, you may use an external CA for requesting the certificate. For example, if your Certificate Manager is a subordinate CA to an external CA, you can get the Directory Server’s certificate signed by the same CA that signed your Certificate Manager’s certificate.

- If certificates the Directory Server and Certificate Manager will use during SSL-enabled communication already exist, check the CA that issued these certificates. The CA that issued the Directory Server’s SSL server certificate must be trusted by the Certificate Manager. Similarly, the Directory Server must trust the CA that issued the certificate the Certificate Manager will use for client authentication.
- Determine how you want the Certificate Manager to bind to the directory: whether to bind as `CN=directory manager` or as another user; if it’s another user, the entry must have read-write privileges to the directory tree that contains entries for end-entities to whom you intend to issue certificates.
- If you’re not the directory administrator, consult the directory administrator about making changes to the schema, if required.
- Keep your directory documentation handy. For an online version of iPlanet Directory Server documentation, check this file:
`<server_root>/manual/index.html`

You can find documentation for other versions of Directory Server at this site:
http://www.sun.com/software/products/directory_srvr/home_directory.html

Step 2. Set Up the Directory for Publishing

For a Certificate Manager to publish certificates and CRLs to an LDAP directory, the directory needs to be set up to receive certificate- and CRL-related information from the Certificate Manager.

- Step A. Verify the Directory Schema
- Step B. Add an Entry for the CA
- Step C. Identify an Entry That Has Write Access
- Step D. Verify Entries for End Entities
- Step E. Specify the Directory Authentication Method
- Step F. Modify the Certificate Mapping File
- Step G. Restart Directory Server

Step A. Verify the Directory Schema

For a Certificate Manager to publish certificates and CRLs to a directory, it must be configured with specific attributes and object classes. This section discusses those basic schema requirements. It is assumed that you're familiar with directory schema and related terminology. If you're not, check the Directory Server documentation.

Required Schema for Publishing End-Entity Certificates

The Certificate Manager publishes an end entity's certificate to the `userCertificate;binary` attribute within the end entity's or subject's directory object. This attribute is multivalued; each value is a DER encoded binary X.509 certificate. The LDAP object class named `inetOrgPerson` allows this attribute. This object class is supported by Netscape Directory Server versions 1.0, 3.x, and 4.x. The mix-in object class named `strongAuthenticationUser` allows this attribute and can be combined with any other object class to allow certificate publication to that object. Note that the Certificate Manager does not automatically add this object class to the schema table of the corresponding Directory Server while publishing or unpublishing end-entity certificates. If the directory object that it finds does not allow the `userCertificate;binary` attribute, the addition or removal of that specific certificate fails.

If you have created user entries as `inetOrgPerson`, the `userCertificate;binary` attribute already exists in the directory. Otherwise, you must add the `userCertificate;binary` attribute to your directory's schema table. For information on modifying directory schema, check the Directory Server documentation.

Required Schema for Publishing the CA Certificate

The Certificate Manager publishes its own CA certificate in the `caCertificate;binary` attribute of the CA's directory object when the server is started; this is the object that corresponds to the Certificate Manager's issuer name. This is a required attribute of the `certificationAuthority` object class. Note that the Certificate Manager will add this object class to the directory entry for the CA, provided that it finds the CA's directory entry.

Required Schema for Publishing CRLs

The Certificate Manager maintains its list of revoked certificates in its internal database; this list is called the certificate revocation list (CRL). You can configure the server to publish the CRL to the directory whenever it is generated, which could be when a certificate is revoked and at regular intervals. You can also manually trigger the server to generate a CRL and publish it to the directory.

The Certificate Manager publishes the updated CRL to the CA's directory object under this attribute: `certificateRevocationList;binary`.

This attribute is an attribute of the object class `certificationAuthority`. The value of the attribute is the DER encoded binary X.509 certificate revocation list. The CA's entry must already be a certificate authority.

Step B. Add an Entry for the CA

Complete this step only if you want to manually create an entry for your CA in the directory—that is, you do not want use the automated feature built into the `LdapCaSimpleCAMap` plug-in module for creating the CA's entry in a directory.

For the Certificate Manager to publish its CA certificate and CRL, the directory must include an entry for the CA. This section explains how to manually add this entry in iPlanet Directory Server using the Directory Server window (which you can launch from within iPlanet Console). To add this entry in Netscape Directory Server 3.x, use its HTML forms-based interface (also called the HTTP gateway).

When adding the CA's entry to the directory, you need to select the entry type based on the distinguished name of your CA:

- If your CA's distinguished name begins with the `CN` component, create a new `person` entry for the CA. (If you select a different type of entry, the interface may not allow you to specify a value for the `CN` component.)
- If your CA's distinguished name begins with the `OU` component, create a new `organizational unit` entry for the CA.

After you select the correct entry type, you need to specify the required information to create the entry. Note that the entry you create doesn't have to be in the `certificationAuthority` object class. The Certificate Manager will convert this entry to the `certificationAuthority` object class automatically by publishing its CA's signing certificate (as explained in "Required Schema for Publishing the CA Certificate" on page 643).

To create an entry for the Certificate Manager in Netscape Directory Server 4.x:

1. Log in to Console (see "Logging In to iPlanet Console" on page 344).
2. Locate the Directory Server instance you want the Certificate Manager to use for publishing certificates and CRLs.
3. Double-click the instance or select the instance and click Open.

This opens the Directory Server window.

4. Select the Directory tab.
5. Select the domain name, right click, select New, and then select Other.

The "New object" window appears.

6. Select "person" and click OK.

The Property Editor - New window appears.

7. Enter the required information.

Full name. Enter the common name (the value of the `CN` component) of the CA exactly as it appears in the issuer DN; this DN shows up in the CA's signing certificate. For example, if your CA's issuer DN is `CN=testCA, OU=Research Dept, O=Siroe Corp, ST=California, C=US`, you should enter `testCA` in this field.

Last name. Enter the name again; it must be the same as the one you entered in the "Full name" field.

8. Keep the default values in the remaining fields, and click OK.

The new entry appears in the Directory tab.

9. Verify that the entry has been created.
 - a. Double-click Directory Administrators, click Members, and then click Add.
 - b. Search for the user entry you added earlier.
 - c. Click OK and again OK.

Step C. Identify an Entry That Has Write Access

When you configure the Certificate Manager to work with Directory Server, you'll be required to specify a distinguished name in the directory that has read-write permissions to the directory. To publish certificates and CRLs to the directory, the Certificate Manager needs to use a user entry (in the directory) that has write access to the directory. This enables the Certificate Manager to bind to the directory as this user and modify the user entries with certificate-related information and the CA entry with CA's certificate and CRL related information.

To provide the Certificate Manager with a user entry that has read-write permission, you can do either of the following:

- Use the DN of an existing entry that has write access. For example, you can use the entry of the Directory Manager or choose an alternative.
- Give write access to the user entry you created for the Certificate Manager in the previous step. The entry can be identified by the Certificate Manager's DN. For example, it may look like this:

```
CN=testCA, OU=Research Dept, O=Siroe Corp, ST=California, C=US
```

For instructions on giving write access to the Certificate Manager's entry, see your LDAP directory documentation. In either case, note the entry DN and the corresponding password as you will be required to identify this user entry to the Certificate Manager later; see "Step 5. Identify the Publishing Directory" on page 680.

Step D. Verify Entries for End Entities

The publishing directory must contain an entry for each end entity for whom you want a certificate published. If the end entity does not have an entry in the directory, the Certificate Manager will not be able to publish the end entity's certificate.

To add an entry for each end entity, you can use the tools provided with Directory Server. Keep in mind that the end-entity entries must belong to an object class, such as `inetOrgPerson`, that allows the `userCertificate;binary` attribute.

NOTE If you configured the Certificate Manager to use directory-based authentication for end entities and are using the same directory for authentication and publishing, you may not have to deal with this issue. The server will not issue certificates to end entities that do not have entries in the directory. See "Authentication of End Entities During Certificate Enrollment" on page 537.

Step E. Specify the Directory Authentication Method

Depending on how you want the Certificate Manager to authenticate to the directory, you must set up Directory Server for one of the following methods of communication:

- Basic authentication
- SSL without client authentication
- SSL with client authentication

The instructions that follow explain how to configure Netscape Directory Server 4.x for all of the above methods of communication. If you're using any other directory, refer to the documentation that accompanied that product.

Publishing With Basic Authentication

To configure Directory Server for basic authentication:

1. Go to the Directory Server window.
2. Select the Configuration tab, and then in the right pane, select the Encryption tab.
3. Make sure that the Enable SSL box is unchecked. If it's checked, uncheck it.
4. Click Save.

You are prompted to restart the server. Don't restart the server yet; you can do this after you've made all the configuration changes.

Publishing Over SSL Without Client Authentication

To configure the Directory Server for SSL-enabled communication:

1. Go to the Directory Server window.
2. Select the Configuration tab, and then in the right pane, select the Encryption tab.
3. Check the Enable SSL box.
4. In the Cipher Family section, check the RSA box.
5. Click the Cipher Preferences button and select the appropriate SSL ciphers.

For details on individual ciphers, click the Help button.

6. In the Client Authentication section, select the “Allow client authentication” option.

Be sure not to select the “Require client authentication” option. If you do, Console will not be able to communicate with the directory.

7. Click Save.

You are be prompted to restart the server. Don’t restart the server yet; you can do this after you’ve made all the configuration changes.

Publishing Over SSL With Client Authentication

For the Certificate Manager to publish to the directory with SSL client authentication, Directory Server must:

- Contain an SSL server certificate in its certificate database
- Trust the CA that issued its SSL server certificate
- Trust the CA that issued the certificate the Certificate Manager will use for SSL client authentication
- Use a valid, secure port number for communication with the Certificate Manager
- Have SSL-enabled communication turned on in its configuration

The steps that follow explain how you can configure Directory Server for all of the above.

1. Check the Directory Server’s certificate database.

Before getting an SSL server certificate, determine whether Directory Server already has an SSL server certificate installed in its certificate database and whether you want Directory Server to use the same certificate during the SSL handshake.

To check the Directory Server’s certificate database:

- a. Go to the Directory Server window.
- b. Select the Tasks tab.
- c. From the Console menu, choose the Manage Certificates option.

The Certificate Management dialog box appears showing a list of all the certificates installed for Directory Server.

- d. Scroll through the list to see if it contains the SSL server certificate that you want to use.

If the server has an SSL server certificate, check the CA that has issued the certificate. If this CA is trusted by the Certificate Manager, you can configure Directory Server to use the same certificate. If the CA is untrusted by the Certificate Manager and you want the Certificate Manager to trust it, you need to check the Certificate Manager's certificate database for the CA certificate, add it if it isn't present, and specify that it be trusted. For instructions on manipulating the Certificate Manager's certificate database, see "Changing the Trust Settings of a CA Certificate" on page 526.

After you've made sure that the CA is trusted by the Certificate Manager, go to Step 10 on page 654.

If the server does not have an SSL server certificate, or if you don't want the Certificate Manager to trust the CA that has issued the Directory Server's certificate, you must get an SSL server certificate for the Directory Server from a CA that is trusted by the Certificate Manager. You may get this certificate from the Certificate Manager itself. The instructions that follow (Step 2 through Step 9) explain how to do this.

2. Generate an SSL server certificate request for Directory Server.

The steps below explain in general how to generate a certificate signing request (CSR) using the Certificate Setup Wizard, which is built into the Directory Server window available within Console. For detailed instructions on each step of the wizard, you should read the on-screen instructions and view the online help by clicking the Help button.

In the first step of generating the CSR, you will be asked to specify whether the certificate is for a new key pair or an existing key pair and the method for submitting the CSR to the CA.

If you want to request the certificate from an external CA, you should click the Show CA button to see whether the CA of your interest is listed there. If it is listed, you can open the SSL server enrollment interface of that CA so that you can paste the CSR the wizard will generate.

If you want to request the certificate from the Certificate Manager, there are three possible ways in which you can submit the CSR to the Certificate Manager:

- o Submit the CSR directly from the wizard; in this method, you do not need to copy the CSR the wizard generates.

- o Submit the CSR as an email to the CA's administrator; to use this method, you need to know the email address of the person who processes certificate requests for the CA and you need to copy the CSR the wizard generates.
- o Submit the CSR manually by pasting it into the Certificate Manager's SSL server enrollment form; to use this method, you need to copy the CSR the wizard generates.

To generate a certificate request:

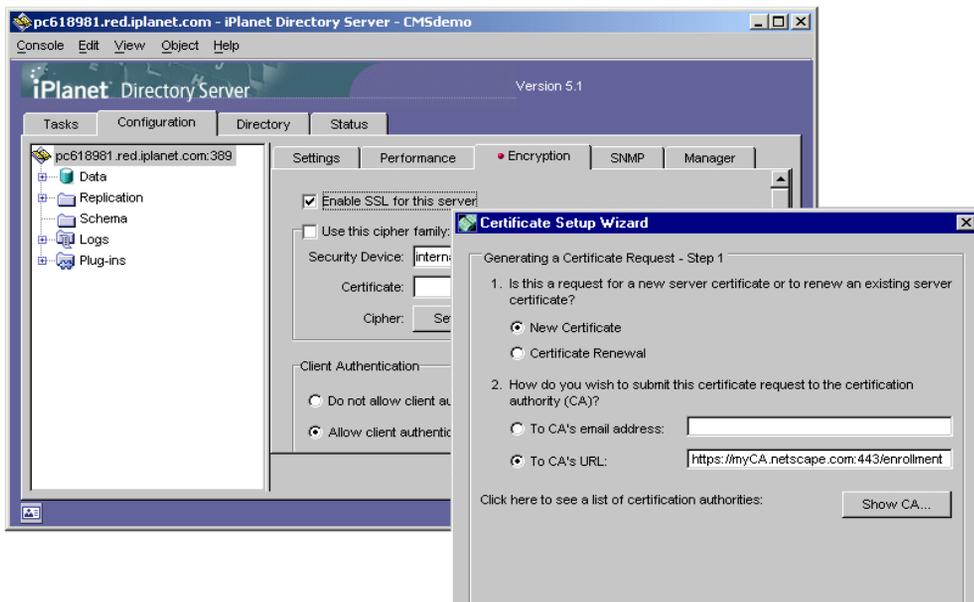
- a. In the Directory Server window, select the Tasks tab, and then click the Certificate Setup Wizard button.
- b. Select the token for generating the key pair (and for storing the certificate). Since you don't have the certificate, select No.

If you're generating the certificate for the first time, the wizard informs you that it needs to create a trust database (`cert7.db` and `key3.db` files) for Directory Server.

- c. When prompted for the password, enter the password.

Remember this password because you will be required to supply it when starting Directory Server from now on. Once the trust database is generated, the wizard steps for generating the CSR begin.

- d. You are asked to specify whether the certificate is for a new key pair or an existing key pair and the method for submitting the CSR to the CA.



The choices for submitting the CSR to the CA include the following:

To CA's email address. Select this if you want to send the CSR to the CA administrator's email address. Type the administrator's email address (for example, `jdoe@siroe.com`) in the adjoining field. The administrator will then be required to submit the request to the CA by pasting the CSR in the CA's server enrollment form.

To CA's URL. Select this if you want to submit the CSR to the Certificate Manager directly. Depending on the end-entity port that's enabled, type either of the following URL:

```
http://<CA_hostname>:<end_entity_port>/enrollment or
https://<CA_hostname>:<end_entity_ssl_port>/enrollment
```

Note that the request submitted to the CA's URL gets queued for approval by the Certificate Manager agent.

- e. When the wizard displays the CSR, if you are running the wizard on a Windows NT system, copy the CSR (displayed in its base-64 encoded format) to a text file. The information you copied should look similar to the sample shown below. Do not make any changes to it. (As indicated in the message, a copy of this information is also saved to the `/temp` file in the host machine's file system.)

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

```
MMIIBnzCCAQgCAQAwXzELMAkGA1UEBhMCVXMxEzARBgNVBAgTCkNBTElGT1JO
SUExHTAbBgNVBAopME5ldHNjYXB1IENvbW0uIENvcnAuMRwwGgYDVQQDExNzd
XByaXlhLW50Lm1jb20uY29tMIGfMA0GCSGSIb3DQEBAQUAA4GNADCBiQKBgQC
k49jBib3SZQqTt5YtIGugnVOq38Y1CcB9owCsapR+DIow8MUVWGRUT38mcX01
fpNT4hzW1aePiHersIMZFLxRge10kEtJ0C1WfNqKzrnOfpL1H3CjcljSM5hWa
Ft0M5hfzEtPk+XBsMbu3dCtbRacxs0M2I0AVkf+kp24ePvqD8QIDAQABoA
```

```
-----END NEW CERTIFICATE REQUEST-----
```

3. Submit the request to a CA and get the SSL server certificate.

If you decided to submit the certificate request to an external CA, you need to go to that CA's enrollment area and use the form provided for requesting SSL server certificates. After you submit the request, hold on to the confirmation message until you receive the certificate. When the CA sends the certificate to you, complete the remaining configuration, starting from Step 6 on page 652.

The instructions in this step explain how to request the SSL server certificate from the Certificate Manager manually. You should complete this step if you didn't use the auto-submit feature of the wizard to directly send the CSR to the Certificate Manager's URL.

To submit the request to the Certificate Manager manually:

- a. Open a web browser window.
- b. Go to the end-entity interface of the Certificate Manager (or to the Registration Manager that's connected to the Certificate Manager).
- c. In the left frame, under Server, click SSL Server.
- d. In the server enrollment form that appears, enter the required information:

PKCS#10 Request. Paste the base-64 encoded blob, including the -----BEGIN NEW CERTIFICATE REQUEST----- and -----END NEW CERTIFICATE REQUEST----- marker lines, you copied to the text file earlier.

Name. Type your full name.

Email. Type your business email address, for example, jdoe@sirioe.com.

Phone. Type your business phone number.

Additional Comments. Type any information that will help you identify this request in the future or will help the person who will process this request.

- e. Click Submit.
4. Approve the request you submitted.

Skip to the next step if you submitted the CSR to an external CA. Complete this step if you submitted the CSR to the Certificate Manager.

To access the agent queue and approve the SSL server certificate request you submitted:

- a. In the browser window, go to the Certificate Manager Agent Services interface. (If you submitted the request to a Registration Manager, go its agent interface.)
- b. In the left frame, select List Requests. In the form that appears, select the "Show pending request" option and click Find.
- c. In the request queue, locate the request you submitted and click Details.
- d. Verify the information and click Do It.

If your request contained all the required information, the server issues a certificate and you should see a message indicating so.

- e. Click Show Certificate.

The complete details about the certificate appear. Don't close the page; in the next step, you'll need to copy the certificate from this page.

5. Copy the SSL server certificate.

You must go through this step, irrespective of whether you submitted the CSR to the Certificate Manager or to an external CA.

To install the certificate in the Directory Server's database, you need to have a copy of the certificate in its base 64-encoded format:

- o If you submitted the CSR to an external CA, wait till you receive the certificate. When you receive the certificate, look for the base 64-encoded blob of the certificate.
- o If you submitted the CSR to the Certificate Manager, check the confirmation page that you received in the previous step; it contains the certificate in its base 64-encoded format.

The steps below explain how to copy the base 64-encoded blob of the certificate from the confirmation page that you received from the Certificate Manager:

- a. In the page that shows the certificate details, scroll down to the section that says "Installing this certificate in a server".
- b. Copy the base-64 encoded certificate, including the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` marker lines, to a text file or to the clipboard; be sure not to make any changes to the text blob. An example of the information you should copy is shown below:

```
-----BEGIN CERTIFICATE-----  
MMIICVDCCAf6gAwIBAgIBDDANBgkqhkiG9w0BAQQFADB6MQswCQYDVQQGEwJV  
UzELMAKGA1UECBMQ0M0ExFjAUBgNVBACjTDU1vdW50YWluIFZpZxcxETAPBgNV  
BAoTCE5ldHNjYXB1MRUwEwYDVQQLEwxTZW1cm10eVB1YnMxHDAaBgNVBAMTE0  
NlcnRpZmljYXR1IE1hbmFnZXIwHhcNOTkwNzA5MjIxNjQ5WhcMDAwNzA4MjIx  
NjQ5WjCQYDVQQGEwJVczETMBEGA1UECBMQ0FMSUZPUBk5JQTEdMBSGA1UEChM  
UTm0c2NhcGUgQ29tbs4gQ29ycC4xHDAaBgNVBAMTE3N1cHJpeW
```

6. Install the certificate in the Directory Server's certificate database.

You must go through this step, irrespective of whether you requested the certificate from the Certificate Manager or from an external CA.

To install the SSL server certificate in the Directory Server's certificate database:

- a. Go to the Directory Server window.
 - b. Start the Certificate Setup Wizard.
 - c. In the first step that the wizard displays, select the “Install Certificate for This Server” option.
 - d. In the second step, select the “The certificate is located in the following text field” option and paste the certificate blob, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines, you copied earlier.
 - e. Follow the prompts and add the certificate to the certificate database.
7. Copy the CA chain.

You must go through this step, irrespective of whether you requested the certificate from the Certificate Manager or from an external CA.

The steps in this section explain how to copy the CA chain, if you requested the SSL server certificate from a Certificate Manager. If you got the certificate from an external CA, make sure that the CA’s chain exists in certificate database of Directory Server; otherwise, go to the CA site and copy the chain.

- a. Go to the end-entity interface of the Certificate Manager (or to the Registration Manager that’s connected to the Certificate Manager).
 - b. Click the Retrieval tab.
 - c. In the left frame, click Import CA Certificate Chain.
 - d. In the form that appears, select the “Display the CA certificate chain in PKCS#7 for importing into a server” option, and click Submit.
- The CA certificate chain appears.
- e. Copy the base-64 encoded certificate blob, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines, to a text file or the clipboard.
8. Install the CA chain in the Directory Server as a trusted CA.

To install the CA chain:

- a. Go to the Directory Server window.
- b. Start the Certificate Setup Wizard.
- c. Select the option to install a certificate for a trusted certificate authority.

- d. Select the “The certificate is located in the following text field” option and paste the certificate blob, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines, you copied earlier.
 - e. Follow the prompts and add the CA certificate chain to the certificate database of Directory Server.
9. Confirm that the new certificates are installed.

To verify that the certificates are installed in the certificate database of Directory Server:

- a. In the Directory Server window, select the Tasks tab.
- b. From the Console menu, select Manage Certificates.

The Certificate Management dialog box appears showing a list of certificates installed for Directory Server.

- c. Scroll through the list. You should find the certificates you installed. If you find the certificates, your server is ready for SSL activation.

10. Verify the port number.

Before turning on SSL-enabled communication for Directory Server, you must verify that the configured port number can be used for this purpose. If not, you must change the port number to a valid one.

To modify the port (for a secure port) on which the Directory Server listens for incoming requests:

- a. In the Directory Server window, select the Configuration tab, and then in the navigation tree, select the root (the topmost) item.
- b. Select the Settings tab in the right pane.

Port. Type the port number you want the server to use for non-SSL communication. The default port number is 389.

Encrypted Port. Type the port number you want the server to use for SSL-enabled communication. The default secure port number is 636. The encrypted port number that you specify must not be the same as one you specified in the Port field.

- c. Click Save.

You are be prompted to restart the server. Don't restart the server yet; you can restart it after you've made all the changes.

Be aware that changing the Directory Server port number requires you to change the corresponding port number in all other servers that communicate with the directory. For example, if you have other servers installed that point to the directory, you need to update those server configurations to use the new port number. For details, see *Managing Servers with Console*. To locate this document, see `<server_root>/manual/index.html`.

11. Turn on SSL-enabled communication.

To turn on SSL-enabled communication in Directory Server:

- a. In the Directory Server window, select the Configuration tab, and then in the right pane, select the Encryption tab.
- b. Check the Enable SSL box.
- c. In the Cipher Family section, check the RSA box.
- d. Click the Cipher Preferences button and select the appropriate SSL ciphers. For details on individual ciphers, click the Help button.
- e. In the Token drop-down list, select the token that contains the key pair for the certificate you installed (or for the certificate you want the server to use).
- f. Select the certificate you want the server to use during SSL-enabled communication with the Certificate Manager.
- g. In the Client Authentication section, select the appropriate option:

Do not allow client authentication. Select this if you want to configure the directory for basic authentication or for SSL-based communication without client authentication.

Allow client authentication. Select this if you want to configure the directory for SSL client authenticated communication.

Require client authentication. Don't select this option. If you do, Console will not be able to communicate with Directory Server. This is because Console does not support client-authenticated communication yet. For example, if you're using the same directory for user data and configuration information of other servers and if you configure Directory Server to require client authentication, you will no longer be able to manage your Netscape servers from Netscape Console; instead, you will be required to use the command-line tools.

- h. Click Save.

You are prompted to restart the server. Don't restart the server yet; you can restart it after you've made all the required changes.

Step F. Modify the Certificate Mapping File

This step explains how to modify the `certmap.conf` file to add a certificate mapping rule for the CA's entry you created. You need to go through this step only if you configured the directory for SSL client authenticated communication. Otherwise, skip to "Step G. Restart Directory Server" on page 668.

When the Certificate Manager presents its certificate for SSL client authentication, Directory Server uses the mapping rule specified in the `certmap.conf` file to locate the corresponding entry in the directory and then determine the access privileges set for the entry. The certificate mapping file is located in the `<server_root>/shared/config` directory, where `<server_root>` is the directory in which the Directory Server binaries are installed.

The `certmap.conf` file specifies the following:

- Where in the directory tree the server should begin its search for locating the entry in the directory
- What certificate attributes the server should use as search criteria when searching for the entry in the directory
- Whether the server needs to go through any additional verification process

The file contains one or more named mappings, each applying to a different CA.. A mapping has the following syntax:

```
certmap <name> <issuerDN>
<name>:<property1> [<value1>]
<name>:<property2> [<value2>]
...
<name>:<propertyn> [<valuen>]
```

The first line specifies a name for the entry and the DN of the issuer of the client certificate—in this case, the issuer of the certificate the Certificate Manager will present during client authentication. (By default, the Certificate Manager uses its SSL server certificate generated during installation.) The name is arbitrary; you can define it to be whatever you want. However, the issuer DN must exactly match the issuer DN of the CA that has issued the certificate the Certificate Manager will use for client authentication. For example, the following two issuer DN lines differ only in the number of spaces separating the attribute value assertions (AVAs), but the Directory Server will treat these two entries as different:

```
certmap moz CN=myCA,OU=myDept,O=myCompany,C=US
certmap moz CN=myCA,OU=myDept,O=myCompany, C=US
```

The second and subsequent lines in the named mapping match properties with values. The `certmap.conf` file has six default properties, but the ones that should be of use to you are explained below. For in depth detail about the `certmap.conf` file, see *Managing Servers with iPlanet Console*.

- `DNComps`—This is a list of comma-separated DN attribute tags used to determine where in the directory the server should start searching for directory entries that match the Certificate Manager’s information (that is, the owner of the client certificate). The Directory Server gathers values for these tags from the certificate presented by the Certificate Manager during client authentication and uses the values to form an LDAP DN, which then determines where the server starts its search in the directory. For example, if you set `DNComps` to use the `<O=org>` and `<C=country>` DN attribute tags (`DNComps=O,C`) the server starts the search from the `O=<org>, C=<country>` entry in the directory, where `<org>` and `<country>` are replaced with values from the values specified in the subject DN of the certificate presented for client authentication.
 - If the `DNComps` entry is present but has no value, the server searches the entire LDAP tree for entries matching the filter.
 - If there isn’t a `DNComps` entry in the mapping, the server uses either the `CmapLdapAttr` setting (if present) or the entire subject DN in the Certificate Manager’s certificate.

The following component tags are supported for `DNComps`: `CN`, `OU`, `O`, `C`, `L`, `ST`, `E`, and `Mail`. Case is ignored. You can use `E` or `Mail`, but not both.

- `FilterComp`—This is a list of comma-separated DN attribute tags used to create a filter by gathering information from the subject DN in the certificate presented during client authentication. Directory Server uses the values for these tags to form the search criteria for matching entries in the directory. If Directory Server finds one or more entries in the directory that match the Certificate Manager’s information gathered from the certificate, the search is successful and the server optionally performs a verification. For example, if `FilterComps` is set to use the attribute tags `E` and `UID` (`FilterComps=E,UID`), the server searches the directory for an entry whose values for `E` and `UID` match the Certificate Manager’s information gathered from the client certificate. Email addresses and user IDs are good filters because they are usually unique entries in the directory.

Note that the filter needs to be specific enough to match only the Certificate Manager’s entry in the LDAP directory. The following component tags are supported for `FilterComps`: `CN`, `OU`, `O`, `C`, `L`, `ST`, `E`, and `Mail`. Case is ignored. You can use `E` or `Mail`, but not both.

- `verifycert`—This tells the server whether it should compare the certificate the Certificate Manager presents during client authentication with the certificate found in the Certificate Manager's entry in the directory. It takes one of the two values: `on` or `off`. It is recommended that you set this to `on` for a complete single sign-on solution. This ensures that Directory Server will authenticate the Certificate Manager unless the certificate presented exactly matches the certificate stored in the directory.

The following two examples illustrate two different ways you can use the `certmap.conf` file.

```
certmap default default
default:dnComps
default:filterComps E, UID
```

```
certmap MyCA CN=CA,OU=MyGroup,O=MyCompany,C=US
MyCA:dnComps OU,O,C
MyCA:filterComps E
MyCA:verifycert on
```

This file has two mappings: a default one and another for `MyCA`. When the Directory Server gets a certificate from anyone other than `MyCA`, the server uses the default mapping, which starts at the top of the LDAP tree and searches for an entry matching the client's email address and user ID. If the certificate is from `MyCA`, the server starts its search at the LDAP branch containing the organizational unit and searches for matching email addresses. Also note that if the certificate is from `MyCA`, the server verifies the certificate with the one stored for the entry in the directory; other certificates are not verified. Note that the issuer DN in the certificate must be identical to the issuer DN listed in the first line of the mapping. Even an extra space after a comma will cause a mismatch.

To modify the `certmap.conf` file:

1. In the Directory Server host machine, go to this directory:
`<server_root>/shared/config`
2. Open the `certmap.conf` file in a text editor.

3. Follow the instructions in the file and add the mapping information for the entry you added.

```
# 3. '#' can be used to comment out a line.
#
# 4. DNComps & FilterComps are used to form the base DN and filter resp. for
# performing an LDAP search while mapping the cert to a user entry.
#
# 5. DNComps can be one of the following:
# commented out - take the user's DN from the cert as is
# empty - search the entire LDAP tree (DN == suffix)
# attr names - a comma separated list of attributes to form DN
#
# 6. FilterComps can be one of the following:
# commented out - set the filter to "objectclass="
# empty - set the filter to "objectclass="
# attr names - a comma separated list of attributes to form the filter
#
certmap myCA CN=rootCA, O=siroe.com
#myCA:DNComps
myCA:FilterComps

certmap default default
default:DNComps
default:FilterComps e
#default:verifycert on
#default:CmapLdapAttr certSubjectDN
#default:library <path_to_shared_lib_or_dll>
#default:InitFn <Init function's name>
```

The figure above shows the following mapping rule being added to the file:

```
certmap myCA CN=rootCA, O=siroe.com
#myCA:DNComps
myCA:FilterComps
```

This mapping rule specifies that if the name of the CA that signed the certificate used for SSL client authentication by the Certificate Manager is `myCA` and that the issuer name or DN of the CA is `CN=rootCA, O=siroe.com`, the server should use the `FilterComps` attributes to locate the entry.

If you determine that the `certmap.conf` file needs an empty `DNComps` mapping (because your certificate subject name has no overlap with the corresponding directory DN), you may need to modify the default base DN in Directory Server by adding the following to the Directory Server configuration file:

```
dn: cn=config
changetype: modify
replace: nsslapd-certmap-basedn
nsslapd-certmap-basedn: dc=siroe, dc=com
```

4. Save your changes, and close the file.

Step G. Restart Directory Server

For all your changes to take effect, you must restart Directory Server.

- **Starting Directory Server**

If you configured the Directory Server for basic authentication or SSL-enabled communication without client authentication, you can start the server from the Directory Server window from within iPlanet Console:

- a. Click the Tasks tab.
- b. Click Restart the Server.

- **Starting SSL-Enabled Directory Server**

If you configured the Directory Server for SSL-enabled communication with client authentication, here's how you must start the server:

- On Windows NT, start the server from the server's host machine and supply the PIN or password that protects the key pair you generated for the Directory Server's certificate. For security reasons, the dialog box that prompts you for this PIN appears only on the server's host machine.
- On Unix, start the server from the command line.

Step 3. Configure the Certificate Manager to Publish Certificates

This section explains how to specify certificate mapping and publishing rules the Certificate Manager should use to publish certificates to the correct entries in the directory.

- **Step A. Modify the Default Mappers, Publishers, and Publishing Rules**
- **Step B. Add Mappers, Publishers, and Publishing Rules**

Step A. Modify the Default Mappers, Publishers, and Publishing Rules

Complete this step if you decided to use any of the default mappers, publishers, and publishing rules created during installation. If you want to create new mappers, publishers, and publishing rules, skip to the next step, "Step B. Add Mappers, Publishers, and Publishing Rules" on page 666.

During installation, the Certificate Manager automatically creates a set of mappers that you would most likely want to use. The names of the default mappers are as follows:

- `LdapUserCertMap`—for locating the correct attribute of user entries in the directory in order to publish user certificates.
- `LdapCrlMap`—for locating the correct attribute of the CA’s entry in the directory in order to publish the CRL.
- `LdapCaCertMap`—for locating the correct attribute of the CA’s entry in the directory in order to publish the CA certificate.

Similar to mappers, the Certificate Manager also creates a set of publishers for your convenience. The names of the default publishers are as follows:

- `LdapCaCertPublisher`
- `LdapCrlPublisher`
- `LdapUserCertPublisher`

The Certificate Manager also creates a set of publishing rules using the default mappers and publishers. The names of these rules are as follows:

- `LdapCrlRule`
- `LdapCaCertRule`
- `LdapUserCertRule`

It is important that you review each of the default mappers, publishers, and publishing rules and modify them as suitable. The instructions that follow explain how to modify the default mappers, publishers, and publishing rules.

Modifying the Default Mappers

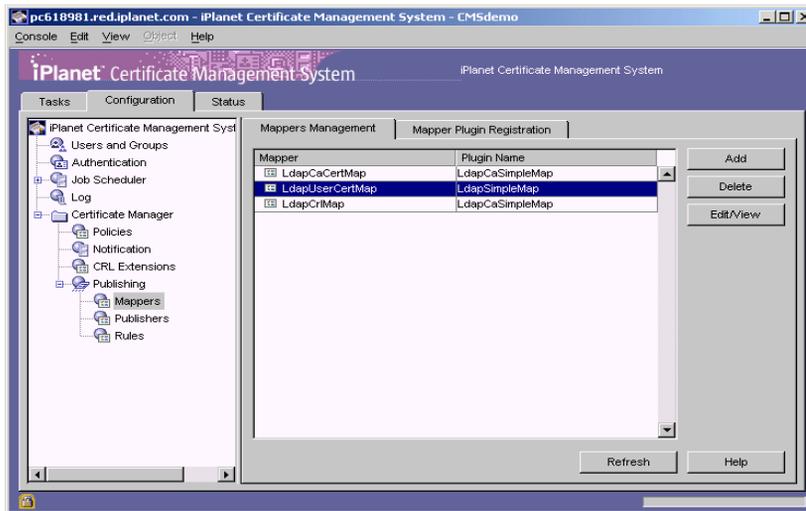
You can modify a mapper by editing its configuration parameter values; you cannot change the name of a mapper. To change the name of a mapper, you need to create a new mapper exactly like the mapper you want to rename, except with a new name, and delete the old mapper.

To modify a mapper:

1. Log in to the CMS window for the Certificate Manager (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.

3. In the navigation tree, select Publishing, and then select Mappers.

The right pane shows the Mappers Management tab, which lists configured mappers.



4. In the Mapper list, select a mapper that you want to modify.

For the purposes of completing this instruction, assume that you selected the mapper named `LdapUserCertMap`.

5. Click Edit/View.

The Mapper Editor window appears, showing how this mapper is configured. An example is shown below.



6. Make the necessary changes and click OK.

Note that if your CA certificate does not have the `cn` component in its subject name, be sure to adjust the CA certificate mapping DN pattern to reflect the DN of the entry in the directory where the CA certificate is to be published. For example, if your CA certificate subject DN is `o=Siroe Corp` and the CA's entry in the directory is `cn=Certificate Authority, o=Siroe Corp`, the pattern should look like this:

```
cn=Certificate Authority, o=$subj.o
```

This rule applies to all mappers.

7. To modify the remaining mappers, repeat steps Step 4 through Step 6.
8. Click Refresh to see the update status of all the mappers.

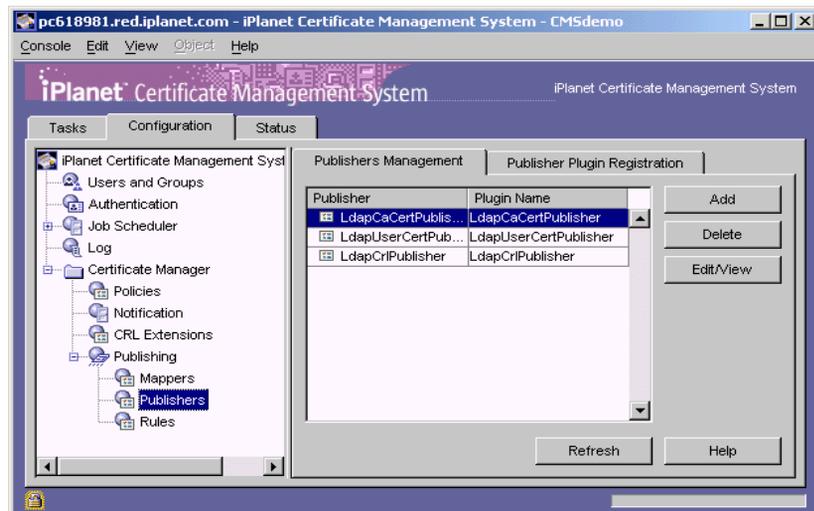
Modifying the Default Publishers

Modifying a publisher involves changing its configuration parameter values; you cannot change the name of a publisher. To change the name of a publisher, create a new publisher using the same publisher plug-in module with the same parameter values, and delete the old one.

To modify a publisher:

1. In the navigation tree, select Publishing, and then select Publishers.

The right pane shows the Publishers Management tab, which lists configured publishers.



2. In the Publisher list, select a publisher that you want to modify.

For the purposes of this instruction, assume that you selected the publisher named `LdapUserCertPublisher`.

3. Click Edit/View.

The Publisher Editor window appears, showing how this publisher is currently configured.



4. Make the necessary changes and click OK.

You are returned to the Publishers Management tab.

5. To modify the remaining publishers, repeat steps Step 2 through Step 4.
6. Click Refresh to see the update status of all the publishers.

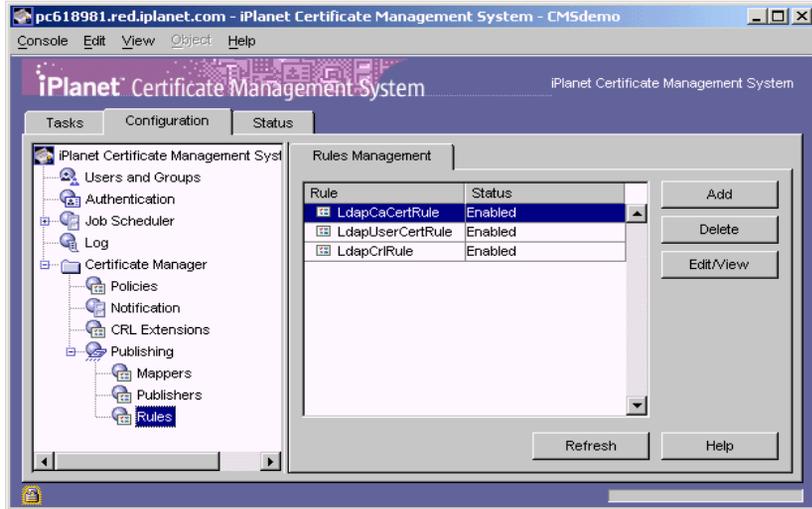
Modifying the Default Publishing Rules

Modifying a publishing rule involves changing its configuration parameter values; you cannot change the name of a publishing rule. To change the name of a publishing rule, create a new rule with the same parameter values, and delete the old one.

To modify a publishing rule:

1. In the navigation tree, select Publishing, and then select Rules.

The right pane shows the Rules Management tab, which lists configured publishing rules.

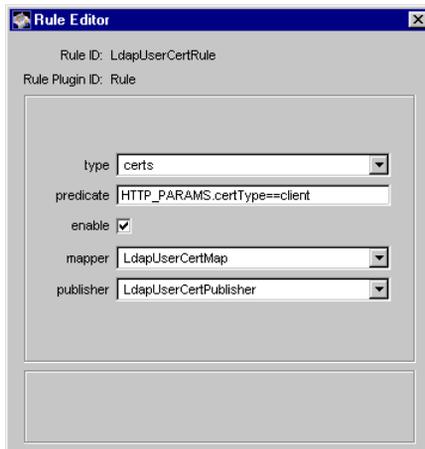


2. In the Rule list, select a publishing rule that you want to modify.

For the purposes of this instruction, assume that you selected the rule named LdapUserCertRule.

3. Click Edit/View.

The Rule Editor window appears, showing how the rule is configured.



4. Make the necessary changes and click OK.
You are returned to the Rules Management tab.
5. To modify the remaining rules, repeat steps Step 2 through Step 4.
6. Click Refresh to see the update status of all the rules.

Step B. Add Mappers, Publishers, and Publishing Rules

Complete this step only if you need to create new mappers, publishers, or publishing rules. For example, if you already configured the Certificate Manager for publishing all types of certificates in “Step A. Modify the Default Mappers, Publishers, and Publishing Rules” on page 660, you can skip to the next step, , “Step 4. Configure the Certificate Manager to Publish CRLs” on page 672.

The instructions that follow cover how to add new mappers, publishers, and publishing rules for a CA certificate and for end-entity certificates. Creating of new mappers, publishers, and publishing rules for CRLs is covered in “Step 4. Configure the Certificate Manager to Publish CRLs” on page 672.

Follow the steps that’s appropriate for you:

- Creating a Mapper for the CA Certificate
- Creating a Mapper for End-Entity Certificates
- Creating a Publisher for the CA Certificate
- Creating a Publisher for End-Entity Certificates
- Creating a Publishing Rule for the CA Certificate
- Creating Publishing Rules for End-Entity Certificates

Creating a Mapper for the CA Certificate

Creating a mapper for the CA certificate involves creating an instance of the mapper module that enables the Certificate Manager to locate the CA’s entry in the directory. Later, when creating the publishing rule for the CA certificate, you specify the mapper you create here.

To create a mapper:

1. In the navigation tree of the CMS window, under Publishing, select Mappers.
The right pane shows the Mappers Management tab, which lists configured mappers.

2. Click Add.

The Select Mapper Plugin Implementation window appears. It lists registered mapper modules.

3. Select a module.

The following choices are the ones provided by default with the Certificate Manager for mapping a CA's certificate to the CA's directory entry. (If you have registered any custom mapper modules, they too will be available here for selection.)

LdapDNCompsMap. Select this if you want the server to locate the CA's entry by formulating the entry's distinguished name from components in the certificate subject name and using it as the search DN.

LdapDNExactMap. Select this if you want the server to locate the CA's entry by searching for its certificate subject name.

LdapSimpleMap. Select this if you want the server to locate the CA's entry by formulating the entry's DN from components specified in the certificate subject name and attribute variable assertion (AVA) constants.

LdapSubjAttrMap. Select this if you want the server to locate the CA's entry by searching for an LDAP attribute whose value matches the certificate subject name.

For the purposes of this instruction, assume that you selected `LdapDNCompsMapper`.

4. Click Next.

The Mapper Editor window appears.

5. Enter the appropriate information:

Mapper ID. Type a unique name for the mapper that will help you identify it; use an alphanumeric string with no spaces.

baseDN. Type the DN from which the server should start searching for the CA's entry in the directory. If you leave the next field, `dnComps`, blank, the server uses the base DN value to start its search in the directory. For example, `O=siroe.com`.

dnComps. Type DN components (attributes) separated by commas, that you want the server to use to locate an LDAP entry that match the CA's information. The server gathers values for these attributes from the CA certificate subject name and uses the values to form an LDAP DN, which then determines where in the LDAP directory the server starts its search. For

example, if the subject name of your CA's certificate is `CN=testCA, O=siroe.com, C=US`, and you set `dnComps` to use the `O` and `C` attributes of the DN, the server starts the search from the `O=siroe.com, C=US` entry in the directory.

If you leave the `dnComps` field empty, the server checks the value in the `baseDN` field and searches the directory tree specified by that DN. The server searches the entire LDAP tree for entries matching the filter specified by `filterComps` parameter values.

filterComps. Type components the server should use to filter entries that result from the search. The server uses the `filterComps` values to form an LDAP search filter for the subtree. The server constructs the filter by gathering values for these attributes from the certificate subject name; it uses the filter to search for and match entries in the LDAP directory.

If you need additional details about any of these parameters, click the Help button.

6. Click OK.

The Mappers Management tab appears, listing the new mapper.

Creating a Mapper for End-Entity Certificates

Creating a mapper for end-entity certificates involves creating an instance of the mapper module that enables the Certificate Manager to locate the correct end-entity entry in the directory. Later, when creating the publishing rule for end-entity certificates, you specify the mapper you create here.

To create a mapper for end-entity certificates, follow the procedure in Step B.1, above. Unlike the CA certificate mapper configuration, keep this mapper's configuration generic so that the Certificate Manager is able to locate any end-entity entry in the directory.

Creating a Publisher for the CA Certificate

Creating a publisher for the CA certificate involves creating an instance of the publisher module that enables the Certificate Manager to publish the CA certificate to the correct attribute in the CA's directory entry. Later, when creating the LDAP publishing rule for the CA certificate, you specify the publisher you create here.

To create a publisher:

1. In the navigation tree of the CMS window, under Publishing, select Publishers.

The right pane shows the Publishers Management tab, which lists configured publishers.

2. Click Add.

The Select Publisher Plugin Implementation window appears. It lists registered publisher modules.

3. Select the module named `LdapCaCertPublisher`.

Only this module publishes the CA certificate to `caCertificate;binary` attribute in the CA's directory entry. (If you have registered any custom publisher modules, they too will be available here for selection.)

4. Click Next.

The Publisher Editor window appears.

5. Enter the appropriate information:

Publisher ID. Type a unique name for the publisher that will help you identify it later; be sure to use an alphanumeric string with no spaces.

caCertAttr. The field shows `caCertificate;binary`, the directory attribute to publish the CA certificate. Leave it as it is. If the field is empty, type `caCertificate;binary`.

caObjectClass. The field shows `certificationAuthority`, the object class for the CA's entry in the directory. Leave it as it is. If the field is empty, type `certificationAuthority`.

6. Click OK.

The Publishers Management tab appears, listing the new publisher.

Creating a Publisher for End-Entity Certificates

Creating a publisher for end-entity certificates involves creating an instance of a publisher module that enables the Certificate Manager to publish an end-entity certificate to the correct attribute in the end entity's directory entry. Later, when creating publishing rules for end-entity certificates, you specify the publisher you create here.

To create a publisher for end-entity certificates, complete the procedure in Step B.3 above. When selecting the publisher module, be sure to choose the module named `LdapUserCertPublisher` as this is the only module that allows publishing to the `userCertificate;binary` attribute of a mapped-directory entry.

Creating a Publishing Rule for the CA Certificate

Creating a publishing rule for the CA certificate involves creating a rule that uses the mapper and publisher that you created for the CA certificate in the previous steps.

To create a publishing rule:

1. In the navigation tree, under Publishing, select Rules.

The right pane shows the Rules Management tab, which lists configured publishing rules.

2. Click Add.

The Select Rule Plugin Implementation window appears. It lists registered modules that enable creating of publishing rules.

3. Select the module named `Rule`.

This is the default module. (If you have registered any custom modules, they too will be available for selection.)

4. Click Next.

The Rule Editor window appears.

5. Enter the appropriate information:

Rule ID. Type a unique name for the rule; use an alphanumeric string with no spaces.

enable. Select this option.

predicate. Type `HTTP_PARAMS.certType==ca`, indicating that the rule be applied to the CA certificate only. (For information on predicates, see “Using Predicates in Policy Rules” on page 606.)

type. Select `ca`.

mapper. Select the mapper you added for locating the CA’s entry in the directory.

publisher. Select the publisher you added for publishing the CA’s certificate to the directory.

6. Click OK.

The Rules Management tab appears, listing the new rule.

Creating Publishing Rules for End-Entity Certificates

Creating a publishing rule for end-entity certificates involves creating a rule for publishing each type of end-entity certificates the Certificate Manager will issue:

- SSL client certificates
- SSL server certificates

- Object signing certificates
- Registration Manager signing certificates
- OCSP responder certificates
- Router certificates

You need to create a rule for each type of certificate using the mapper and publisher that you created for end-entity certificates.

To create a publishing rule:

1. In the navigation tree, under Publishing, select Rules.

The right pane shows the Rules Management tab, which lists configured publishing rules.

2. Click Add.

The Select Rule Plugin Implementation window appears. It lists registered modules that enable creating of publishing rules.

3. Select the module named `Rule`.

This is the default module. (If you have registered any custom modules, they too will be available for selection.)

4. Click Next.

The Rule Editor window appears.

5. Enter the appropriate information:

Rule ID. Type a name for the rule; use an alphanumeric string with no spaces.

enable. Select this option.

predicate. Type `HTTP_PARAMS.certType==client`, indicating that the rule be applied to client certificates only (see Table 19-2).

type. Select `certs`.

mapper. Select the mapper you added for locating end-entity entries in the directory.

publisher. Select the publisher you added for publishing end-entity certificates (to the `userCertificate;binary` attribute of an end-entity entry in the directory).

6. Click OK.

The Rules Management tab appears, listing the new rule you just created for publishing end users' client certificates.

7. Repeat steps 1 through 6 for each type of end-entity certificate the Certificate Manager will issue. Use Table 19-2 for filling in the correct values in the `type` and `predicate` fields. (For information on predicates, see "Using Predicates in Policy Rules" on page 606.)

Table 19-2 Certificate types and predicate expressions

End-entity certificate type	"type" field value	"predicate" field value
SSL client certificate	<code>certs</code>	<code>HTTP_PARAMS.certType==client</code>
SSL server certificate	<code>certs</code>	<code>HTTP_PARAMS.certType==server</code>
Object signing certificate	<code>certs</code>	<code>HTTP_PARAMS.certType==objSignClient</code>
Certificate Manager signing certificate (subordinate CA)	<code>cacert</code>	<code>HTTP_PARAMS.certType==ca</code>
Registration Manager signing certificate	<code>certs</code>	<code>HTTP_PARAMS.certType==ra</code>
OCSP responder certificate	<code>certs</code>	<code>HTTP_PARAMS.certType==ocspResponder</code>
Router certificate	<code>certs</code>	<code>HTTP_PARAMS.certType==CEP-Router</code>

Step 4. Configure the Certificate Manager to Publish CRLs

If you don't want the Certificate Manager to publish CRLs to the directory, skip to "Step 5. Identify the Publishing Directory" on page 680.

You can configure the Certificate Manager to publish CRLs to the directory that is currently configured for publishing the CA and end-entity certificates. A configured Certificate Manager will publish the CRL to the CA's entry in the specified directory, replacing the old CRL with the new one; the old CRL is not saved. The Certificate Manager connects to the directory using the base DN and password that you will specify in "Step 5. Identify the Publishing Directory" on page 680.

To configure a Certificate Manager to publish CRLs to the directory, follow these steps:

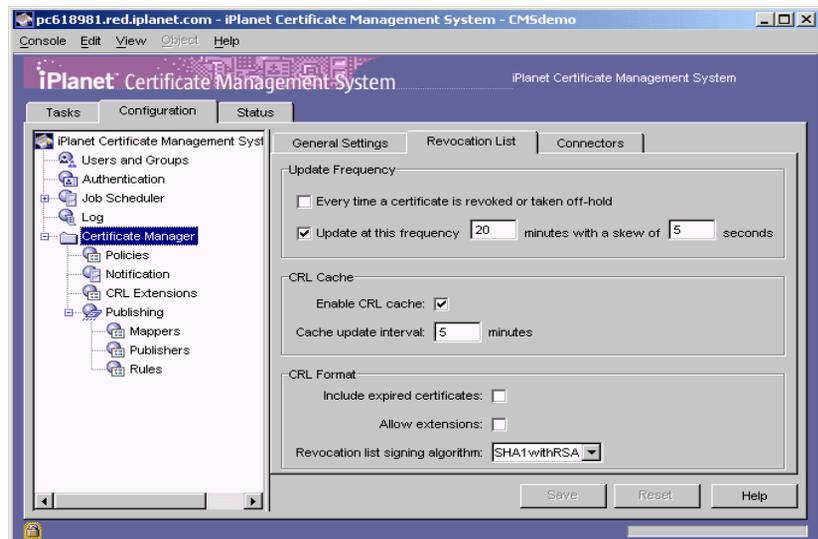
- Step A. Specify CRL Details
- Step B. Set the CRL Extensions
- Step C. Create a Mapper for the CRL
- Step D. Create a Publisher for the CRL
- Step E. Create a Publishing Rule for the CRL

Step A. Specify CRL Details

You can specify information, such as the publishing interval, the CRL version (whether to include CRL extensions), and the signing algorithm the Certificate Manager should use for signing the CRL object.

To specify CRL details:

1. In the navigation tree of the CMS window, select Certificate Manager, and then in the right pane, select the Revocation List tab.



2. In the Update Frequency section, specify the interval for publishing the CRL to the directory:

Every time a certificate is revoked, or taken off-hold. Select this option if you want the Certificate Manager to generate the CRL every time it revokes a certificate. Keep in mind that the Certificate Manager attempts to publish the CRL to the configured directory whenever the CRL is generated, in this case, every time a certificate is revoked. Publishing a CRL can be time consuming if the CRL is large. Configuring the Certificate Manager to publish CRLs every time a certificate is revoked may engage the server for a considerable amount of time; during this time, the server will not be able to service any requests it receives and will not be able to update the directory with any changes it receives.

Update at this frequency. Select this option if you want the Certificate Manager to generate CRLs at regular intervals. In this case, the server publishes the CRL to the configured directory at the interval you specify.

In the adjoining text field, type the interval, in minutes, at which the Certificate Manager should publish CRLs. For example, if you want the server to publish CRLs every day, you should type 1440 in this field.

with a skew of. If you configure the server to update the CRL automatically every time period, the server by default adds a 5 second skew to the next update time to allow time to create the CRL and publish it. For example, if you configure the server to update the CRL every 20 minutes, and if the CRL is updated at 16:00:00, the CRL will be updated again at 16:19:55. You can configure the skew by changing the default value, which is specified in seconds.

3. In the CRL Cache section, specify whether to enable CRL caching:

Enable cache. Check this box to enable CRL caching. Leave the box unchecked if you don't want the server to maintain a cache.

Update interval. If you enabled caching, type the interval for updating the cache.

4. In the CRL Format section, specify the format for publishing the CRL:

Include expired certificates. Check this box if you want the server to include revoked certificates that have expired in the CRL.

Allow extensions. Check this box if you want to allow extensions in the CRL. If you enable this option, the server generates and publishes CRLs conforming to X.509 version 2 standard. If you disable this option, the server generates and publishes CRLs conforming to X.509 version 1 standard. By default, the server publishes version 1 CRLs. If you enable this option, be sure to set the required CRL extensions as described in “Step B. Set the CRL Extensions” on page 675.

Revocation list signing algorithm. Select the algorithm the server should use to sign the CRL. If the Certificate Manager’s signing key type is RSA, select MD2 with RSA, MD5 with RSA, or SHA-1 with RSA. If the Certificate Manager’s signing key type is DSA, select SHA-1 with DSA.

5. To save your changes, click Save.

If the changes you made require you to restart the server, you are prompted accordingly. However, don’t restart the server yet; you can restart it after you’ve made all the required changes.

Step B. Set the CRL Extensions

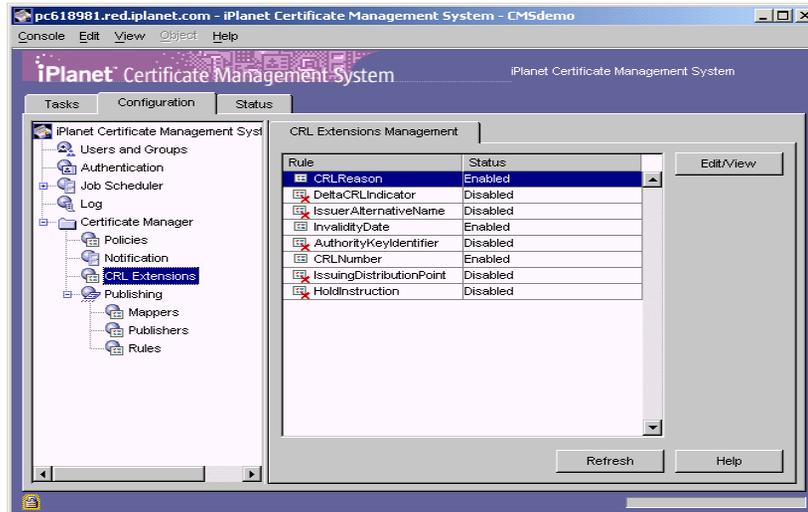
Complete this step only if you configured the Certificate Manager to publish version 2 CRLs—that is, you selected the “Allow extensions” option in “Step A. Specify CRL Details” on page 673.

During installation, the Certificate Manager creates default CRL extension rules; these are documented in *CMS Plug-Ins Guide*. Note that the server is configured to add the CRL Reason extension only; all the other rules are in the disabled state. In this step, you modify the default CRL extension rules to add the required CRL extensions.

To specify the CRL extensions the Certificate Manager should set:

1. In the navigation tree, under Certificate Manager, select CRL Extensions.

The right pane shows the CRL Extensions Management tab, which lists configured extensions.



2. To modify a rule, select it and then click Edit/View.
3. Change the information as appropriate.

Be sure to supply all the required values. Click the Help button for detailed information on individual parameters.

4. Click OK.

You are returned to the CRL Extensions Management tab.

5. To modify other rules, repeat steps 2 through 4.
6. Click Refresh to see the updated status of all the rules.

Step C. Create a Mapper for the CRL

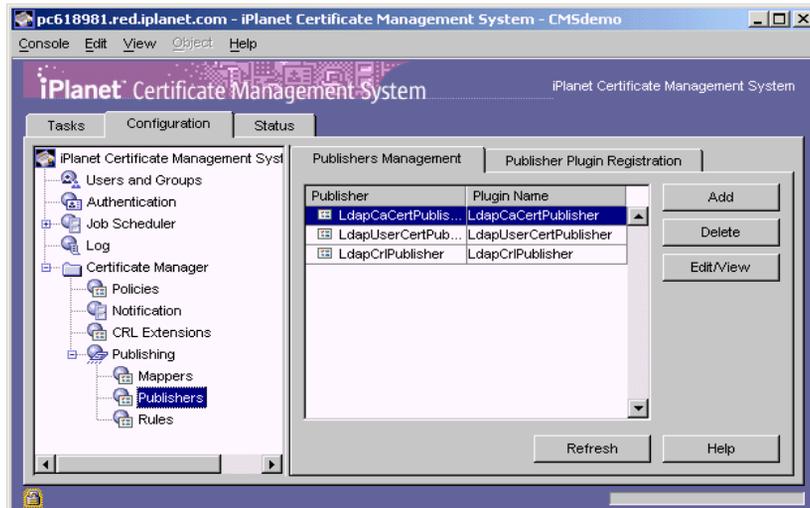
The Certificate Manager publishes the CRL to the `certificateRevocationList;binary` attribute of the CA's directory entry. (See "Required Schema for Publishing CRLs" on page 643.)

Since you already created a mapper for locating the CA's entry (either in "Step A. Modify the Default Mappers, Publishers, and Publishing Rules" on page 660 or in "Creating a Mapper for the CA Certificate" on page 666), you can configure the Certificate Manager to use that mapper to locate the CA's entry for publishing the CRL; you don't need to create another mapper for publishing CRLs.

Step D. Create a Publisher for the CRL

Creating a publisher for the CRL involves creating an instance of the publisher module that enables the Certificate Manager to publish the CRL to the correct attribute in the CA's directory entry. In the next step, described in "Step E. Create a Publishing Rule for the CRL" on page 679, you specify the publisher you create here.

For your convenience, during the installation of a Certificate Manager a publisher named `LdapCrlPublisher` is automatically created for publishing CRLs. You don't need to create a new publisher if the default one still exists. In which case, you can skip to the next step.



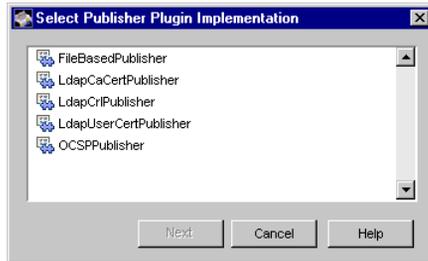
You should create a new publisher if the default `LdapCrlPublisher` instance has been deleted:

1. In the navigation tree, click Publishers.

The right pane shows the Publishers Management tab, which lists configured publisher instances.

2. Click Add.

The Select Publisher Plugin Implementation window appears. It lists registered publisher modules.

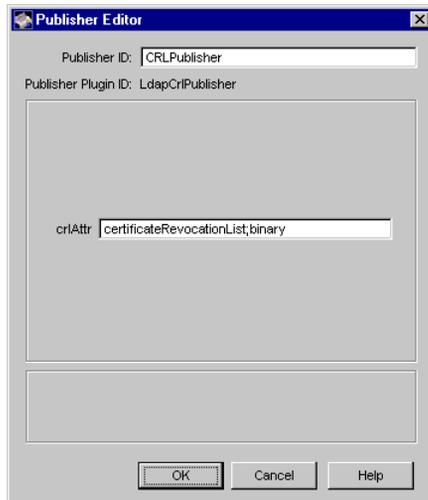


3. Select the module named `LdapCrlPublisher`.

Only this publisher module enables the Certificate Manager to publish the CRL to the `certificateRevocationList;binary` attribute of the CA's directory entry. (If you have registered any custom publisher modules, they too will be available for selection.)

4. Click Next.

The Publisher Editor window appears.



5. Enter the appropriate information:

Publisher ID. Type a name for the rule; use an alphanumeric string with no spaces. For example, `CRLPublisher`.

crAttr. Make sure this field shows the directory attribute to publish the CRL, `certificateRevocationList;binary`. If necessary, type it in.

6. Click OK.

The Publishers Management tab appears, listing the new publisher.

Step E. Create a Publishing Rule for the CRL

Creating a publishing rule for the CRL involves creating a rule that uses the mapper and publisher created for publishing CRLs.

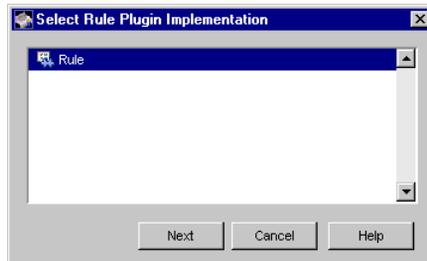
To create a new publishing rule:

1. In the navigation tree, click Rules.

The right pane shows the Rules Management tab, which lists any currently configured publishing rules.

2. Click Add.

The Select Rule Plugin Implementation window appears.



3. Select the module named `Rule`.

This is the default module. (If you have registered any custom modules, they too will be available for selection.)

4. Click Next.

The Rule Editor window appears.

5. Enter the appropriate information:

Rule ID. Type a name for the rule; be sure to use an alphanumeric string with no spaces.

enable. Select this option.

predicate. Leave this field blank.

type. Select `crl`.

mapper. Select the mapper you added for locating the CA's entry in the directory.

publisher. Select the publisher you added for publishing the CRL.

6. Click OK.

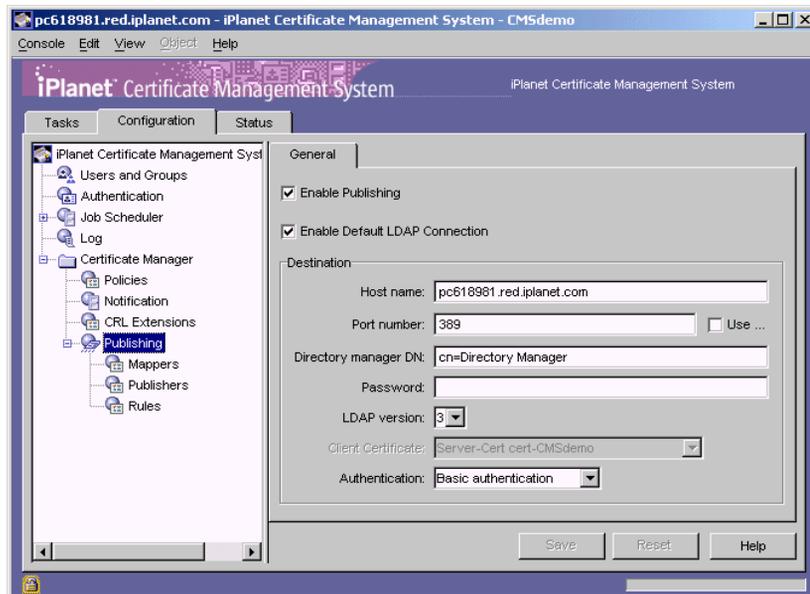
The Rules Management tab appears, listing the new rule.

Step 5. Identify the Publishing Directory

To identify the directory to which the Certificate Manager should publish the CA certificate, end-entity certificates, and CRLs:

1. In the navigation tree of the CMS window, select Certificate Manager, and then select Publishing.

The right pane shows the publishing details necessary for the server to publish to an LDAP-compliant directory.



2. To enable LDAP publishing, select both “Enable Publishing” and “Enable default LDAP connection” options.

3. In the Destination section, identify the Directory Server instance.

Host name. Type the full host name of the Directory Server instance in this format: <machine_name>.<your_domain>.<domain>

The Certificate Manager uses this name to locate the directory.

If you configured the Directory Server for SSL client authenticated communication (in “Step E. Specify the Directory Authentication Method” on page 646), the name you enter here must match the `CN` component in the subject DN of the Directory server’s SSL server certificate. For example, the host name may look like `corpDirectory.siroe.com`.

Port number. Type the TCP/IP port number at which the Directory Server is listening to certificate and CRL publishing requests from the Certificate Manager; you specified this port in “Verify the port number.” on page 654. The port you specify must be unique on the Directory Server host system; make sure no other application is attempting to use the port.

Authentication. Select the authentication type appropriate to your Directory Server configuration. The choices are `Basic authentication` and `SSL client authentication`.

If you configured the Directory Server for basic authentication or for SSL communication without client authentication, select `Basic authentication` and specify values for the Directory manager DN and password.

If you configured the Directory Server for SSL communication with client authentication, select `SSL client authentication`, select the `Use SSL communication` option, and identify the certificate that the Certificate Manager must use for SSL client authentication to the directory.

Use SSL communication. Select this option if the port number you specified is an SSL port; deselect the box if the port is non-SSL. The type of port you specify determines whether the Certificate Manager needs to do SSL client authentication prior to publishing certificates and CRLs to the directory.

Client certificate. Select the certificate you want the Certificate Manager to use for SSL client authentication to the publishing directory. By default, the Certificate Manager uses its SSL server certificate for this purpose (see “SSL Server Key Pair and Certificate” on page 459).

Directory manager DN. Type the distinguished name (DN) of the directory entry that you identified in “Step C. Identify an Entry That Has Write Access” on page 645. The Certificate Manager uses this DN to access the directory tree and to publish to the directory. The access control set up for this DN determines whether the Certificate Manager can perform publishing.

Typically, you would want to enter the directory manager's DN because it has read-write permission to the entire directory tree (the root DN). For more information on root DN, see Appendix A, "Distinguished Names" in *CMS Plug-Ins Guide*.

Password. Type the password for this DN. The Certificate Manager saves this password in the single sign-on password cache and uses it during startup; for details, see "Required Start-up Information" on page 322. (If you change the password, the server updates the single sign-on password cache with the new password.)

LDAP version. Select the version of LDAP protocol appropriate to your version of Directory Server. If the directory you want the Certificate Manager to publish to is based on Netscape Directory Server 1.x, select version 2. For Directory Server versions 3.x and later, select LDAP version 3.

4. To save your changes, click Save.

The server attempts to connect to the specified Directory Server. If the information you specified is incorrect, the server displays an error message and you will need to correct the information and save your changes again.

If the changes you made require you to restart the server, you will be prompted accordingly. In that case, restart the server.

Step 6. Test Certificate and CRL Publishing

To test whether you've configured the Certificate Manager correctly to publish certificates and CRLs to the directory, follow these steps:

- Step A. Decide a Directory Entry for Requesting a Certificate
- Step B. Request a Certificate
- Step C. Approve the Request
- Step D. Download the Certificate to the Browser
- Step E. Check if the Directory Has the Certificate
- Step F. Revoke the Certificate
- Step G. Check the Directory for the CRL

Step A. Decide a Directory Entry for Requesting a Certificate

Decide on a user entry for which you will request a certificate. This way, you can check whether the Certificate Manager published the certificate to that entry. The entry you choose could be any end-entity's directory entry, as long as it supports the `userCertificate;binary` attribute.

If you don't have a directory entry yet, you can create one for testing purposes.

Step B. Request a Certificate

The steps outlined below explain how to request a personal certificate from the Certificate Manager using the manual enrollment method. If you've configured the Certificate Manager for automated certificate issuance, for example for directory-based enrollment, you can use the appropriate form and request a certificate.

To request a client or personal certificate from the Certificate Manager:

1. Open a web browser window.
2. Go to the end-entity interface of the Certificate Manager you configured (or to the Registration Manager that's connected to this Certificate Manager). The URL is in this form:

```
https://<hostname>:<end_entity_HTTPS_port> or
http://<hostname>:<end_entity_HTTP_port>
```

3. In the left frame, under **Browser**, select **Manual**.

This opens the manual enrollment form.

4. Fill in all the values and submit the request.

The client prompts you to enter the password for your key database.

5. When you enter the correct password, the client generates the key pairs.

Do not interrupt the key-generation process.

Step C. Approve the Request

Skip this step if you used an automated enrollment method for requesting the certificate. Complete this step if you used the manual enrollment form for requesting the certificate; the request you submitted is waiting in the agent queue for approval by an agent.

To approve the request:

1. Go to the Certificate Manager's Agent Services interface.
The URL is in this format: `https://<hostname>:<agent_port>`
2. In the left frame, click List Requests.
3. In the form that appears, select the "Show pending requests" option and click Find.
4. In the list of pending requests, locate the request you submitted and approve the request.

You should see a confirmation page indicating that the certificate has been issued. Don't close the page until after you read the next step.

Step D. Download the Certificate to the Browser

To download the certificate into your browser's certificate database:

1. In the confirmation page, scroll down to the section that says "Installing this certificate in a client."
2. Follow the on-screen instructions and download the certificate to your browser's certificate database.

(An alternative way to download the certificate is from the Retrieval tab of the end-entity services interface.)
3. Open the browser's security information window and verify that the certificate has been stored in the certificate database.

Step E. Check if the Directory Has the Certificate

If you've configured the Certificate Manager and Directory Server correctly, the Certificate Manager automatically publishes the certificate to the directory whenever it issues a certificate.

Verify that the Certificate Manager has published the certificate to the correct user entry. If you're using Netscape Directory Server version 4.x you can do this verification from the Directory Server window as follows:

1. In iPlanet Console, double-click the Directory Server instance that corresponds to the publishing directory.
This opens the Directory Server window.
2. Select the Directory tab.

3. Locate the user entry for which you requested the certificate.
4. Double-click the entry and check if the entry has a `certificate` attribute.

You should find the certificate published to the attribute. You won't be able to see anything interesting about the certificate; it will be a DER-encoded binary blob.

Alternatively, you can point your browser to the user entry in the directory to verify that the certificate has been published. To do this:

1. Open a web browser window.
2. In the URL field, type
`ldap://<hostname>:<port>/<base_dn>??sub?(uid=<user_id>)`,
 substituting `<hostname>` with the fully qualified host name of the Directory Server, `<port_number>` with the port number at which the Directory Server is listening to publishing requests from the Certificate Manager `<base_dn>` with the DN to start searching for the user's entry, and `<user_id>` with the ID of the user to whom you issued the certificate.

For example, if the directory host name is `corpDirectory`, port number is `389`, base DN is `O=siroe.com`, and user's ID is `jdoe`, the URL would look like this:
`ldap://corpDirectory:389/O=siroe.com??sub?(uid=jdoe)`

In the resulting page, look for the user's certificate-related information. The information typically includes the owner of the certificate, the CA that issued the certificate, the serial number, the validity period, and the certificate fingerprint.

Step F. Revoke the Certificate

To check whether you've configured the Certificate Manager to publish the CRL to the directory correctly, revoke the certificate you issued. In "Step A. Specify CRL Details" on page 673, if you didn't configure the Certificate Manager to publish the CRL every time a certificate is revoked, go back to the Revocation List tab and select the "Every time a certificate is revoked or taken off-hold" option. After you complete testing, remember to go back to the same tab and uncheck the option.

To revoke the certificate:

1. Go to the end-entity interface for the Certificate Manager (or to the Registration Manager that's connected to this Certificate Manager. Be sure to go to the HTTPS interface (the revocation feature is not available in the HTTP interface).
2. Select the Revocation tab.

3. In the left frame, select User Certificate.
The User Certificate Revocation form appears.
4. In the Revocation Reason section, select Unspecified and click Submit.
The client displays the “Select a Certificate” dialog box and prompts you to choose the certificate you want to revoke.
5. Select the certificate you downloaded and click OK.
The certificate is revoked.

Step G. Check the Directory for the CRL

Verify that the Certificate Manager published the CRL (in this case, containing the single certificate that you revoked) to the correct location in the directory—that is, the `certificateRevocationList;binary` attribute of the CA’s entry in the directory.

1. Go to the publishing directory.
2. Locate the CA’s entry.
3. Check the `certificateRevocationList;binary` attribute.

You should find the CRL published.

Manually Updating Certificates and CRLs in a Directory

Normally you do not need to manually update the directory with certificate-related information; if configured properly, the Certificate Manager handles most of the updates automatically. However, a situation might arise in which you need to update the directory manually. For example, Directory Server might be down for a while and be unable to receive changes from the Certificate Manager. In such a situation, use the forms provided in the Certificate Manager Agent Services interface to manually update the directory.

Certificate Manager’s publishing directory can be manually updated by a Certificate Manager agent only. Agent operations are restricted to those with a valid agent certificate; see “Agent’s Certificate for SSL Client Authentication” on page 399. For complete details on agent operations, see *CMS Agent’s Guide*.

Manually Updating Certificates in the Directory

The Update Directory Server form in the Certificate Manager Agent Services interface enables you to manually update the directory with certificate-related information. This form lets you initiate a combination of the following operations:

- Update the directory with certificates.
- Remove expired certificates from the directory.

Note that you can automate removal of expired certificates from the publishing directory by scheduling an automated job. For details, see Chapter 17, “Scheduling Automated Jobs.”

- Remove revoked certificates from the directory.

To manually update the directory with changes:

1. Open a web browser window.
2. Go to the Certificate Manager Agent Services interface.

You must submit the proper certificate to get access to this page.

3. Select the Update Directory Server link.

The Update Directory Server page appears.

4. Select the appropriate options.
5. When you are done specifying the changes that you want updated, click Update Directory.

The Certificate Manager starts updating the directory with the certificate information in its internal database. In some circumstances, for example if the changes are substantial, updating the directory can take considerable time. During this period, any changes made through the Certificate Manager (for example, any certificates issued or any certificates revoked) may not be included in the update. If you have issued or revoked any certificates during that time, you need to update the directory again to reflect those changes.

When the directory update is complete, the Certificate Manager displays a status report. If for some reason the process gets interrupted, the server logs an error message. Be sure to check logs if that happens; for details, see “Monitoring CMS Logs” on page 803.

Note that if the Certificate Manager is installed as a root CA, when using the agent interface to update the directory with valid certificates, the CA signing certificate may get published using the publishing rule set up for user certificates and you may get an object class violation error (or other errors in the mapper). You can avoid this by selecting the appropriate serial-number range to not include the CA signing certificate; the CA signing certificate is the first certificate a root CA issues.

If the root CA has issued a subordinate CA certificate, the certificate may also get published using the publishing rule set up for user certificates, resulting in an object class violation error. To avoid the problem in publishing the subordinate CA certificate, you will need to do this:

- Modify the default publishing rule for user certificates by changing the value of the `predicate` parameter to `HTTP_PARAMS.certType!=ca`.
- Use the `LdapCaCertPublisher` publisher plug-in module to add another rule, with the `predicate` parameter set to `HTTP_PARAMS.certType==ca`, for publishing subordinate CA certificates.

Manually Updating the CRL in the Directory

The Update Certificate Revocation List form in the Certificate Manager Agent Services interface enables you to manually update the directory with CRL-related information.

To manually update the CRL information in the directory:

1. Go to the Certificate Manager Agent Services page.

You must submit the proper client certificate to get access to this page.

2. Select Update Revocation List.

The Update Certificate Revocation List page appears.

3. From the Signature algorithm drop-down list, select the appropriate signature algorithm.
4. Click Update.

The Certificate Manager starts updating the directory with the CRL in its internal database. In some circumstances, for example, if the CRL is large, updating the directory may take considerable time. During this period, any changes made to the CRL (for example, any new certificates revoked) may not be included in the update.

When the directory is updated, the Certificate Manager will display a status report. If the process gets interrupted for some reason, the server logs an error message. Be sure to check logs if that happens; for details, see “Monitoring CMS Logs” on page 803.

Publishing Certificates and CRLs to a File

iPlanet Certificate Management Server (CMS) provides a customizable publishing framework for the Certificate Manager, enabling it to publish certificates, certificate revocation lists (CRLs), and other certificate-related objects to any of the supported repositories—an LDAP-compliant directory, a flat file, and an online validation authority—using the appropriate protocol. This chapter explains how to configure the Certificate Manager to publish certificates and CRLs to a file.

Note that configuring the Certificate Manager for publishing is optional—you can turn this feature off without affecting any of the certificate issuance and management operations handled by the server.

The chapter has the following sections:

- Configuring Certificate Manager to Publish to Files
- Managing Mapper and Publisher Plug-in Modules

Configuring Certificate Manager to Publish to Files

The Certificate Manager can publish certificates and CRLs to flat files, which can then be imported into any repository, for example, into a relational database. If you configure the server to publish certificates and CRLs to flat files, it publishes them to files as DER-encoded binary blobs.

- For each certificate the server issues, it creates a file that contains the certificate in its DER-encoded format. Each file is named as `cert-<serial_number>.der`, where `<serial_number>` specifies the serial number of the certificate contained in the file. For example, the filename for a certificate with serial number 1234 will be `cert-1234.der`.
- Every time the server generates the CRL (which could be every time it revokes a certificate and at a regular interval), it creates a file that contains the new CRL in its DER-encoded format. Each file is named as `crl-<this_update>.der`, where `<this_update>` specifies the value derived from the time-dependent variable named `This Update` of the CRL contained in the file. For example, the filename for a CRL with `This Update: Friday January 28 15:36:00 PST 2000`, will be `crl-949102696899.der`.

To configure the Certificate Manager to publish certificates and CRLs to files, follow these steps:

- Step 1. Before You Begin
- Step 2. Configure the Certificate Manager
- Step 3. Test Publishing

Step 1. Before You Begin

Before configuring a Certificate Manager to publish the CA certificate, end-entity certificates, and CRLs to flat files:

- Read section “FileBasedPublisher Plug-in Module” in Chapter 6, “Publisher Plug-in Modules” of *CMS Plug-Ins Guide*.
- Identify the machine that will contain the DER-encoded files, and create a directory for the files.
- Make sure that the machine has sufficient disk space to accommodate the DER-encoded files that the Certificate Manager will generate; the server generates a file for every certificate it issues and for every CRL it generates. If disk space is a constraint, you can configure the server to create files on two different hosts, one for certificates and another one for CRLs.
- Read “Publishing of CRLs” on page 634. Determine whether you want the Certificate Manager to publish version 1 or version 2 CRLs to the directory. If you decide to publish version 2 CRLs, read Chapter 4, “Certificate Extension Plug-in Modules” of *CMS Plug-Ins Guide* and determine the CRL extensions you want the Certificate Manager to set; you will be required to configure the server to set these extensions.

- Decide the interval for publishing CRLs—configuring the server to publish every time a certificate is revoked will result in that many CRL files.
- Determine the backup media and schedule for these files.

Step 2. Configure the Certificate Manager

To configure a Certificate Manager to publish certificates and CRLs to files, follow these steps:

- Step A. Create a Publisher for the File
- Step B. Create Publishing Rules for Certificates
- Step C. Create a Publishing Rule for CRLs
- Step D. Specify CRL Details
- Step E. Set the CRL Extensions
- Step F. Make Sure Publishing is Enabled

Step A. Create a Publisher for the File

Creating a publisher for the file involves creating an instance of the publisher module that enables the Certificate Manager to publish certificates and CRLs to files. In the next step, “Step B. Create Publishing Rules for Certificates” on page 695, you specify the publisher you create here.

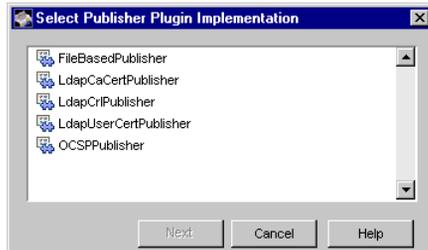
To create a publisher:

1. Log in to the CMS window for the Certificate Manager (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select Certificate Manager, select Publishing, and then select Publishers.

The right pane displays the Publishers Management tab, which lists configured publisher instances.

4. Click Add.

The Select Publisher Plugin Implementation window appears. It lists registered publisher modules.



5. Select the module named `FileBasedPublisher`.

Only this publisher module enables the Certificate Manager to publish certificates and CRLs to flat files.

6. Click Next.

The Publisher Editor window appears.



7. Enter the appropriate information:

Publisher ID. Type a name for the rule. Be sure to use an alphanumeric string with no spaces. For example, `PublishCertsToFile`.

directory. Type the complete path to the directory in which the Certificate Manager should create the DER-encoded files; the path can be an absolute path or can be relative to the CMS instance directory. For example, `C:\certificates`.

8. Click OK.

You are returned to the Publishers Management tab. It should now list the publisher you just created.

9. If you want to publish certificates and CRLs to two separate directories, repeat steps 4 through 8 to create another publisher (for example, `PublishCrlsToFile`) with the value of the `directory` parameter set to the file path to the other directory (for example, `C:\crls`).

Step B. Create Publishing Rules for Certificates

Creating a publishing rule for CA certificate and end-entity certificates involves creating a rule that uses the publisher that you created in the previous step.

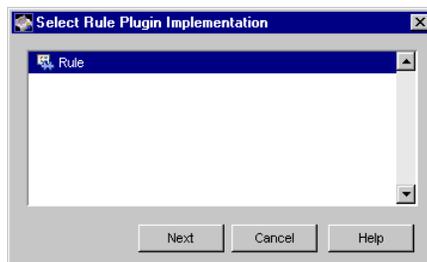
To create a publishing rule:

1. In the navigation tree, under Publishing, select Rules.

The right pane displays the Rules Management tab, which lists any configured publishing rules.

2. Click Add.

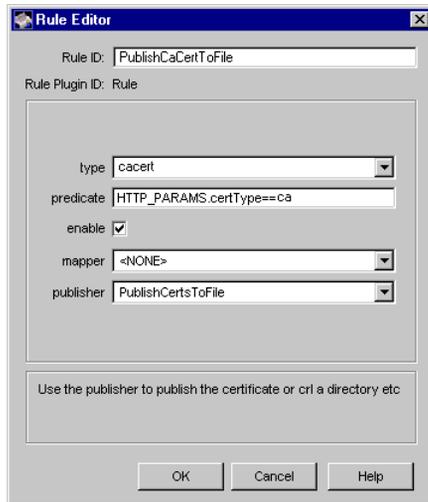
The Select Rule Plugin Implementation window appears.



3. Select the module named `Rule`.

This is the default module. (If you have registered any custom modules, they too will be available for selection.)

4. Click Next.



5. Enter the appropriate information:

Rule ID. Type a name for the rule that will help you identify it later; use an alphanumeric string with no spaces. For example, `PublishCaCertToFile`.

type. Select `cacert`.

predicate. Type `HTTP_PARAMS.certType==ca`.

enable. Select this option.

mapper. Select `<NONE>`.

publisher. Select the publisher you created in the previous step, Step A. For example, `PublishCertsToFile`.

6. Click OK.

The Rules Management tab appears, listing the new rule you just created for publishing the CA certificate to the file.

7. Repeat steps 2 through 6 to create publishing rules for each type of end-entity certificate the Certificate Manager will issue. Use Table 20-1 for filling in the correct values in the `type` and `predicate` fields. (For information on predicates, see “Using Predicates in Policy Rules” on page 606.)

Table 20-1 Certificate types and predicate expressions

End-entity certificate type	“type” field value	“predicate” field value
SSL client certificate	certs	HTTP_PARAMS.certType==client
SSL server certificate	certs	HTTP_PARAMS.certType==server
Object signing certificate	certs	HTTP_PARAMS.certType==objSignClient
Certificate Manager signing certificate (subordinate CA)	cacert	HTTP_PARAMS.certType==ca
Registration Manager signing certificate	certs	HTTP_PARAMS.certType==ra
OCSP responder certificate	certs	HTTP_PARAMS.certType==ocspResponder
Router certificate	certs	HTTP_PARAMS.certType==CEP-Router

Step C. Create a Publishing Rule for CRLs

Creating a publishing rule for the CRL involves creating a rule that uses the publisher that you created in the previous step.

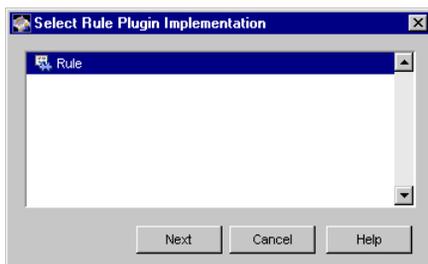
To create a publishing rule:

1. In the navigation tree, under Publishing, select Rules.

The right pane displays the Rules Management tab, which lists configured publishing rules.

2. Click Add.

The Select Rule Plugin Implementation window appears.

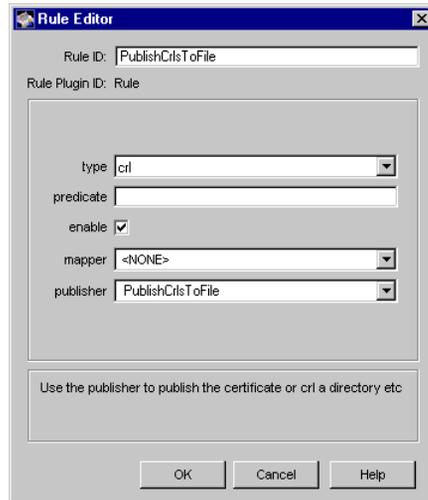


3. Select the module named Rule.

This is the default module. (If you have registered any custom modules, they too will be available for selection.)

4. Click Next.

The Rule Editor window appears.



5. Enter the appropriate information:

Rule ID. Type a name for the rule that will help you identify it later; use an alphanumeric string with no spaces. For example, PublishCertsToFile.

type. Select `crl`.

predicate. Leave this field blank.

enable. Select this option.

mapper. Select `<NONE>`.

publisher. Select the publisher you created in the previous step, Step A.

6. Click OK.

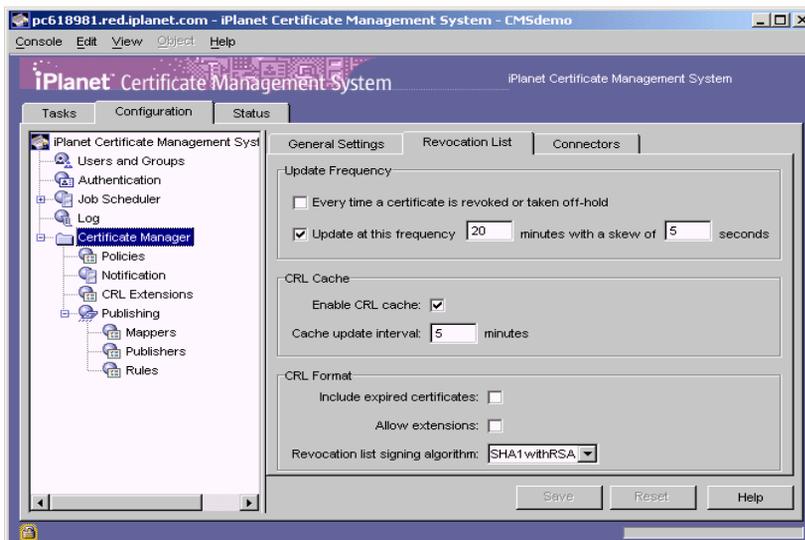
The Rules Management tab appears, listing the new rule you just created for publishing CRLs to files.

Step D. Specify CRL Details

You can specify information, such as the publishing interval, the CRL version (whether to include CRL extensions), and the signing algorithm the Certificate Manager should use for signing the CRL object.

To specify the details for the CRL:

1. In the navigation tree, select Certificate Manager, and then in the right pane, select the Revocation List tab.



2. In the Update Frequency section, specify the interval for publishing the CRL to the directory:

Every time a certificate is revoked, or taken off-hold. Select this option if you want the Certificate Manager to generate the CRL every time it revokes a certificate. Keep in mind that the Certificate Manager attempts to publish the CRL to the configured directory whenever it is generated, in this case, every time a certificate is revoked. Publishing a CRL can be time consuming if the CRL is large. Configuring the Certificate Manager to publish CRLs every time a certificate is revoked may engage the server for a considerable amount of time; during this time, the server will not be able to service any requests it receives and will not be able to update the directory with any changes it receives.

(This setting is not recommended for a standard installation. You can select this option if you want to see the results of revocation immediately, for example, when testing whether the server publishes the CRL to a flat file.)

Update at this frequency. Select this option if you want the Certificate Manager to generate CRLs at regular intervals. In this case, the server publishes the CRL to the configured directory at the interval you specify.

In the adjoining text field, type the interval, in minutes, at which the Certificate Manager should publish CRLs. For example, if you want the server to publish CRLs every day, you should type 1440 in this field.

with a skew of. If you configure the Certificate Manager to update the CRL automatically every time period, the server by default adds a 5 second skew to the next update time to allow time to create the CRL and publish it. For example, if you configure the server to update the CRL every 20 minutes, and if the CRL is updated at 16:00:00, the CRL will be updated again at 16:19:55. You can change the skew by editing the default value, which is specified in seconds.

3. In the CRL Cache section, specify whether to enable CRL caching:

Enable cache. Check this box to enable CRL caching. Leave the box unchecked if you don't want the server to maintain a cache.

Update interval. If you enabled caching, type the interval for updating the cache.

4. In the CRL Format section, specify the format for publishing the CRL:

Include expired certificates. Check this box if you want the server to include revoked certificates that have expired in the CRL.

Allow extensions. Check this box if you want to allow extensions in the CRL. If you enable this option, the server generates and publishes CRLs conforming to X.509 version 2 standard. If you disable this option, the server generates and publishes CRLs conforming to X.509 version 1 standard. By default, the server publishes version 1 CRLs. If you enable this option, be sure to set the required CRL extensions as described in "Step E. Set the CRL Extensions" on page 700.

Revocation list signing algorithm. Select the algorithm the server should use to sign the CRL. If the Certificate Manager's signing key type is RSA, select MD2 with RSA, MD5 with RSA, or SHA-1 with RSA. If the Certificate Manager's signing key type is DSA, select SHA-1 with DSA.

5. To save your changes, click Save.

The configuration is modified. If the changes you made require you to restart the server, you are prompted accordingly. Don't restart the server yet; you can restart it after you've made all the required changes.

Step E. Set the CRL Extensions

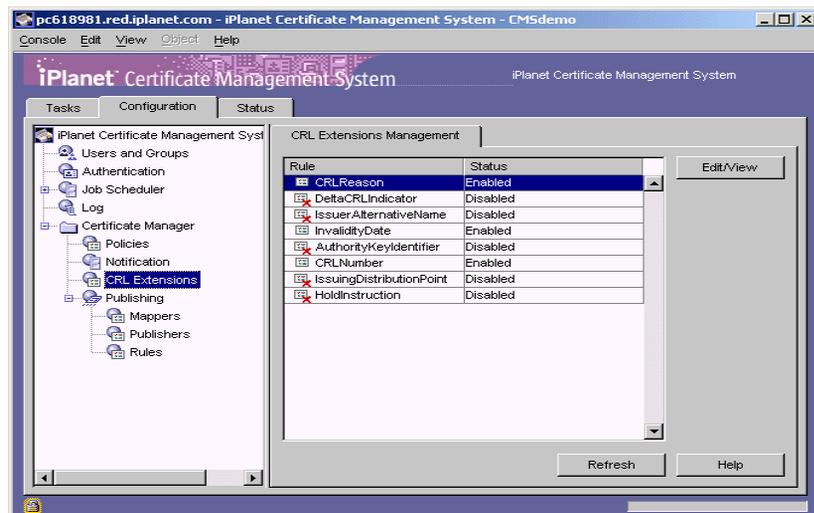
Complete this step only if you configured the Certificate Manager to publish version 2 CRLs in the previous step—that is, if you selected the "Allow extensions" option in "Step D. Specify CRL Details" on page 698.

During installation, the Certificate Manager creates default CRL extension rules. Note that the server is configured to add the CRL Reason extension only; all the other rules are in the disabled state. In this step, you modify the default rules to suit your organization's requirements.

To specify the CRL extensions the Certificate Manager should set:

1. In the navigation tree, select Certificate Manager, and then select CRL Extensions.

The right pane shows the CRL Extensions Management tab, which lists configured extensions.



2. To modify a rule, select it and then click Edit/View.
3. Change the information as appropriate.
Be sure to supply all the required values. Click the Help button for detailed information on individual parameters.
4. Click OK.
You are returned to the CRL Extensions Management tab.
5. To modify other rules, repeat steps 2 through 4.
6. Click Refresh to see the updated status of all the rules.

Step F. Make Sure Publishing is Enabled

To make sure that the Certificate Manager is configured for publishing:

1. In the navigation tree, select Certificate Manager, then select Publishing.

The right pane shows the publishing details necessary for the server to publish to an LDAP-compliant directory, to flat files, or to an online validation authority.

2. Make sure that the Enable Publishing option is selected. If it is already selected, leave it as it is. If it isn't, select it.

(Leave the "Enable default LDAP connection" option as it is; it specifies that the Certificate Manager is configured to publish certificates and CRLs to an LDAP directory.)

3. If you changed anything, click Save to save the changes.

If the changes you made require you to restart the server, you are prompted accordingly. In that case, restart the server.

Step 3. Test Publishing

To verify that the Certificate Manager is publishing certificates and CRLs correctly to files, follow these steps:

- Step A. Request a Certificate
- Step B. Approve the Request
- Step C. Download the Certificate to the Browser
- Step D. Check the File for the Certificate
- Step E. Revoke the Certificate
- Step F. Check the File for the CRL

Step A. Request a Certificate

The steps outlined below explain how to request a personal certificate from the Certificate Manager using the *manual* enrollment method. If you've configured the Certificate Manager for automated certificate issuance, for example for directory-based enrollment, you can use the appropriate form and request a certificate.

To request a client or personal certificate from the Certificate Manager:

1. Open a web browser window.
2. Go to the end-entity interface of the Certificate Manager you configured (or to the Registration Manager that's connected to this Certificate Manager).

The URL is in this form: `https://<hostname>:<end_entity_HTTPS_port>` or `http://<hostname>:<end_entity_HTTP_port>`

3. In the left frame, under **Browser**, click **Manual**.

This opens the manual enrollment form.

4. Fill in all the values and submit the request.

The client prompts you to enter the password for your key database.

5. When you enter the correct password, the client generates the key pair.

Do not interrupt the key-generation process.

Step B. Approve the Request

Skip this step if you requested the certificate using any of the automated enrollment methods in “Step A. Request a Certificate” on page 702. Complete this step if you used the manual enrollment form to request the certificate; the request you submitted is waiting in the agent queue for approval by an agent.

To approve the request:

1. Go to the Certificate Manager's Agent Services interface.

The URL is in this format: `https://<hostname>:<agent_port>`

2. In the left frame, click **List Requests**.
3. In the form that appears, select the “Show pending requests” option and click **Find**.
4. In the list of pending requests, identify the request you submitted and approve the request.

You should see a confirmation page indicating that the certificate has been issued. Don't close the page until after you complete the next step.

Step C. Download the Certificate to the Browser

To download the certificate into your browser's certificate database:

1. In the confirmation page, scroll down to the section that says "Installing this certificate in a client."
2. Follow the on-screen instructions and download the certificate to your browser's certificate database.

(An alternative way to download the certificate is from the Retrieval tab of the end-entity services interface.)

3. Open the browser's security information window and verify that the certificate has been stored in the certificate database.

Step D. Check the File for the Certificate

Whenever the Certificate Manager issues a certificate, it automatically attempts to publish the certificate to the configured repository—in this case, the file. To check whether the Certificate Manager published the correct certificate, you need to do the following:

1. Check whether the server generated the DER-encoded file containing the certificate.

To check whether the server published the certificate as a binary blob to the specified directory, go to the directory or folder you specified for the server to publish certificates. You should see a file with name similar to `cert-<serial_number>.der`, where *<serial_number>* specifies the serial number of the certificate contained in the file. If you don't see a file, check your configuration.

2. Convert the DER-encoded certificate to its base 64-encoded format using the Binary to ASCII tool (see Chapter 8, "Binary to ASCII Tool" of *CMS Command-Line Tools Guide*).

To convert the DER-encoded certificate to its base 64-encoded form:

- a. Open a command window.
- b. Go to this directory: `<server_root>/bin/cert/tools`

- c. At the prompt, enter this: `BtoA [.bat] <input_file> <output_file>`
substituting `<input_file>` with the path to the file that contains the DER encoded certificate and `<output_file>` with the path to the file to write the base-64 encoded certificate. (The optional `.bat` specifies the file extension; this is required only when running the utility on a Windows NT system.)

For example, if the file is in `C:\certificates\cert-1234.der` and you want the base-64 encoded certificate to be in

`C:\certificates\cert-1234.txt`, the command would look like this:

```
BtoA C:\certificates\cert-1234.der
```

```
C:\certificates\cert-1234.txt
```

- d. When the conversion is complete, open the `cert.txt` file in a text editor. You should see a base-64 encoded certificate similar to this:

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

```
MMIIBtgYJYIZIAYb4QgIFoIIBpzCCAZ8wggGbmIIBRaADAgEAAgEBM
A0GCSqGSIb3DQEBAUAMFcxCAJBgNVBAYTA1VTMSwwKgYDVQQQ
KEYyNOZXRzY2FwZSBDb21tdW5pY2F0aWhfyyuougjgjjgmkgjkgmjgjfjgjjj
gfyjfyj9ucyBDb3Jwb3JhdGlvbjpMEaMBGGA1UECXMRSXNzdWluZyZyhgdf
hbfdpffjphotoogdhkBBdXRob3JpdHkwHhcNOTYxMTA4MDkwNzMM0W
hcNOTgxMTA4MDkwNzMM0WjBXMQswCQYDVQQGEwJVUzEsMCo
GA1UEChMjTmV0c2NhcGUgQ29tbXVuaWNhdGlvbnMgQ29yY2F0
aW9ucyBDb3Jwb3JhdGlvbjpMEaMBGGA1UECXMRSXNzdWluZyZyBBdXR
ob3JpdHkwHh
```

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

3. Convert the base 64-encoded certificate to a human-readable form using the Pretty Print Certificate tool (see Chapter 9, “Pretty Print Certificate Tool” of *CMS Command-Line Tools Guide*).

To convert the base 64-encoded certificate to a human-readable form:

- a. Check the command window to make sure that you are in this directory:
`<server_root>/bin/cert/tools`
- b. At the prompt, enter this:

```
PrettyPrintCert [.bat] <input_file> [<output_file>]
```

substituting `<input_file>` with the path to the ASCII file that contains the base-64 encoded certificate and `<output_file>` with the path to the file to write the certificate in a human-readable form. If you don't specify an output file, the certificate information is written to the standard output. (The optional `.bat` specifies the file extension; this is required only when running the utility on a Windows NT system.)

For example, if the base-64 encoded certificate is in `C:\certificates\cert-1234.txt` and you want the human-readable form of the certificate to be displayed on your screen, the command would look like this:

```
PrettyPrintCert.bat C:\certificates\cert-1234.txt
```

When the conversion is complete, you should see the certificate you issued in human-readable form.

- c. Compare the output with the certificate you issued; be sure to check the serial number in the certificate with the one used in the filename.

If everything matches, the Certificate Manager is configured correctly to publish certificates to files.

Step E. Revoke the Certificate

To check whether the Certificate Manager is configured correctly to publish CRLs to flat files, you need to revoke the certificate you issued. Before revoking the certificate, make sure that you've configured the Certificate Manager to publish the CRL every time a certificate is revoked. (In "Step D. Specify CRL Details" on page 698, if you didn't configure the Certificate Manager to publish the CRL every time a certificate is revoked, go back to the Revocation List tab and check the "Every time a certificate is revoked or taken off-hold" option. After the testing, remember to go back to the same tab and uncheck the option.)

To revoke the certificate:

1. Go back to the end-entity interface for the Certificate Manager (or to a Registration Manager that's connected to this Certificate Manager. Be sure to go to the HTTPS interface; the revocation feature is not available in the HTTP interface.

2. Click the Revocation tab.

3. In the left frame, click User Certificate.

The User Certificate Revocation form appears.

4. In the Revocation Reason section, select Unspecified and click Submit.

The browser displays the "Select a Certificate" dialog box and prompts you to choose the certificate you want to revoke.

5. Select the certificate you downloaded and click OK.

The certificate is revoked.

Step F. Check the File for the CRL

Whenever the Certificate Manager generates a CRL, it automatically attempts to publish the CRL to the configured repository—in this case, the flat file. The CRL it publishes is a binary blob, in the DER-encoded format. To check whether the Certificate Manager published the correct CRL (in this case, the CRL contains only one certificate), you need to do the following:

1. Check whether the server generated the DER-encoded file containing the CRL.

To check whether the server published the CRL as a binary blob to the specified directory, go to the directory you specified for the server to publish CRLs. You should find a file with its name in the `crl-<this_update>.der` format, where `<this_update>` specifies the value derived from the time-dependent variable named `This Update` of the CRL contained in the file. If you don't see the file, check your configuration.

2. Convert the DER-encoded CRL to its base 64-encoded format using the Binary to ASCII tool (see Chapter 8, “Binary to ASCII Tool” of *CMS Command-Line Tools Guide*).

To convert the DER-encoded CRL to its base 64-encoded form:

- a. Open a command window.
- b. Go to this directory: `<server_root>/bin/cert/tools`
- c. At the prompt, enter this: `BtoA[.bat] <input_file> <output_file>`

substituting `<input_file>` with the path to the file that contains the DER-encoded CRL and `<output_file>` with the path to the file to write the base-64 encoded CRL. (The optional `.bat` specifies the file extension; this is required only when running the utility on a Windows NT system.)

For example, if the DER-encoded file is in

`C:\crls\crl-949102696899.der` and you want the base-64 encoded CRL to be in `C:\crls\crl-949102696899.txt`, the command would look like this:

```
BtoA C:\crls\crl-949102696899.der
C:\crls\crl-949102696899.txt
```

- d. When the conversion is complete, open the `cr1.txt` file in a text editor. You should see a base-64 encoded CRL similar to this:

```
-----BEGIN CRL-----

MIIBkjCBAIBATANBgkqhkiG9w0BAQQFADAsMREwDwyDVQQKEwhOZXRzY2FwZT
EXMBUGA1UEAxOQ2VydDQwIFRlc3QgQ0EXDTk4MTIxNzIyMzcyNFowgaowIAIB
ExcNOTgxmJjE1MTMxODMyWjAMMAoGA1UdFQDCgEBMCACARIXDTk4MTIxNTEzMj
A0MlOWDDAKBgNVHRUEAwOBAjAgAgERFw05ODEyMTYxMjUxNTRaMAwwCgYDVR0
VBAMKAQEwIAIBEBcNOTgxmJjE3MTAzNzI0WjAMMAoGA1UdFQDCgEDMCACAQoX
DTk4MTEyNTEzMTExOFowDDAKBgNVHRUEAwOBTATANBgkqhkiG9w0BAQQFAAOBg
QBCN85O0GPTnHfImYPROvoorx7HyFz2ZsuKsVblTcemsX0NL7DtOa+MyY0pPr
kXgm157JrkxEJ7GBOeogbAS6iFbmeSqPHj8+JBH5stJNfTCuhaM6Wx63Wc9L
wZXOXPsvpGxq0YYI0+DPfBZlI3z4lCsNczxJV+9NkeMrheEg==

-----END CRL-----
```

3. Convert the base 64-encoded CRL to a human-readable form using the Pretty Print CRL tool (see Chapter 10, “Pretty Print CRL Tool” of *CMS Command-Line Tools Guide*).

To convert the base 64-encoded CRL to a human-readable form:

- a. Check the command window to make sure that you are at this directory:
`<server_root>/bin/cert/tools`
- b. At the prompt, enter this: `PrettyPrintCrl[.bat] <input_file>`
`[<output_file>]`

substituting `<input_file>` with the path to the ASCII file that contains the CRL in its base 64-encoded format and `<output_file>` with the path to the file to write the CRL information in a human-readable form. If you don't specify an output file, the CRL information is written to the standard output. (The optional `.bat` specifies the file extension; this is required only when running the utility on a Windows NT system.)

For example, if the base-64 encoded CRL is in

`C:\crls\crl-949102696899.txt` and you want the human-readable form of the CRL to be displayed on your screen, the command would look like this:

```
PrettyPrintCrl.bat C:\crls\crl-949102696899.txt
```

When the conversion is complete, you should see the CRL (in this case, the CRL will only contain the certificate you revoked) in the human-readable form.

- c. Compare the output with the certificate you revoked.

If they match, the Certificate Manager is configured correctly to publish CRLs to files.

Managing Mapper and Publisher Plug-in Modules

This section explains how to use the CMS window to perform the following operations:

- Registering a Mapper or Publisher Module
- Deleting a Mapper or Publisher Module

Registering a Mapper or Publisher Module

You can register new mapper or publisher plug-in modules in a Certificate Manager's publishing framework. Registering a new mapper or publisher module involves specifying the name of the module and the full name of the Java class that implements the mapper or publisher interface. For example, you can add a mapper implementation, named as follows, to the Certificate Manager's policy framework:

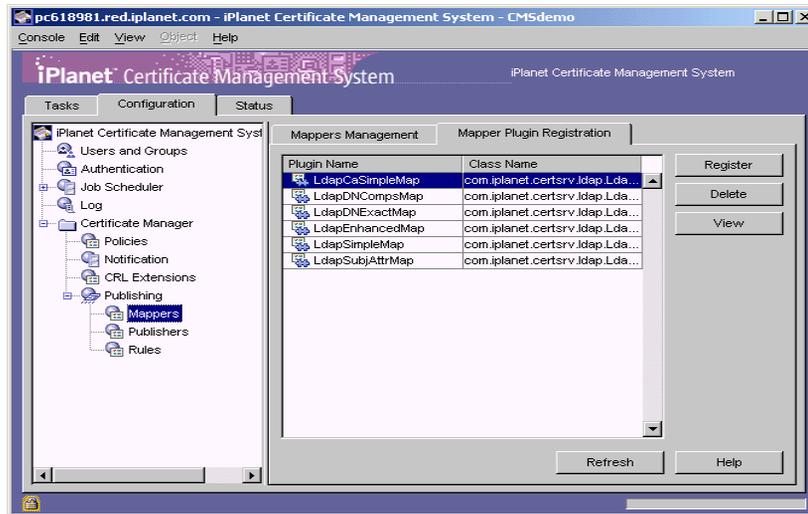
```
com.netscape.publishing.customMapper
```

Before registering a plug-in module, be sure to put the Java class for the module in the `classes` directory (the implementation must be on the class path).

To register a policy module in a Certificate Manager's publishing framework:

1. Log in to the CMS window (see "Logging In to the CMS Window" on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select Certificate Manager, and then select Publishing.
4. Select the appropriate object under Publishing:
 - To register a mapper module, select Mappers, and then in the right pane, select the Mapper Plugin Registration tab.
 - To register a publisher module, select Publishers, and then in the right pane, select the Publisher Plugin Registration tab.

This tab lists registered plug-in modules.



5. Click Register.

If you selected Mapper, the Register Mapper Plugin Implementation window appears. If you selected Publisher, the Register Publisher Plugin Implementation window appears.

6. Specify information as appropriate:

Plugin name. Type a name for the plug-in module.

Class name. Type the full name of the class for this module—that is, the path to the implementing Java class. If this class is part of a package, be sure to include the package name. For example, if you are registering a class named `myMapper` and if this class is in a package named `com.myCompany`, type `com.myCompany.myMapper`.

7. Click OK.

You are returned to the Mapper Plugin Registration tab or Publisher Plugin Registration tab.

8. To view the updated configuration, click Refresh.

Deleting a Mapper or Publisher Module

You can delete unwanted mapper or publisher plug-in modules using the CMS window. Before deleting a module, be sure to delete all the rules that are based on this module.

To delete a mapper or publisher module from a Certificate Manager's publishing framework:

1. Log in to the CMS window (see "Logging In to the CMS Window" on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select Certificate Manager, and then select Publishing.
 - To delete a mapper module, select Mappers, and then in the right pane, select the Mapper Plugin Registration tab.
 - To delete a publisher module, select Publishers, and then in the right pane, select the Publisher Plugin Registration tab.

This tab lists registered plug-in modules.

4. In the Plugin Name list, select the module you want to delete and click Delete.
5. When prompted, confirm the delete action.

Setting Up an OCSP Responder

iPlanet Certificate Management Server (CMS) provides a customizable publishing framework for the Certificate Manager, enabling it to publish certificates and certificate revocation lists (CRLs) to any of the supported repositories—an LDAP-compliant directory, a flat file, and an online validation authority—using the appropriate protocol. This chapter provides an overview of an Online Certificate Status Protocol (OCSP)-compliant PKI setup, and explains how you can use the OCSP service built into the Certificate Manager for real-time verification of certificates issued by the Certificate Manager. The chapter also explains how to configure one or more Certificate Managers to publish CRLs to the OCSP responder, called Online Certificate Status Manager, provided with Certificate Management System.

Note that configuring the Certificate Manager to publish CRLs is optional—you can turn this feature off without affecting any of the certificate issuance and management operations handled by the server.

The chapter has the following sections:

- What's an OCSP-Compliant PKI Setup? (page 714)
- Setting Up a Certificate Manager with OCSP Service (page 719)
- Setting Up a Remote OCSP Responder (page 732)

What's an OCSP-Compliant PKI Setup?

Certificate Management System supports the Online Certificate Status Protocol (OCSP) as defined in the PKIX standard RFC 2560 (see <http://www.ietf.org/rfc/rfc2560.txt>). The OCSP protocol enables OCSP-compliant applications to determine the state of a certificate, including the revocation status, without having to directly check a CRL published by a CA to the validation authority. The validation authority, which is also called an *OCSP responder*, does the checking for the application.

An OCSP-compliant PKI setup generally includes the following, which work together to verify the revocation status of a certificate:

- A CA, which issues and revokes certificates, and periodically publishes the CRL to the OCSP responder.
- An OCSP responder, which maintains the CRL it receives periodically from the CA and, when queried by an OCSP-compliant client about the status of a certificate, sends a digitally signed response.
- OCSP-compliant applications, which, when trying to validate a certificate, query the appropriate OCSP responder (using the OCSP protocol) for the status of the certificate. The applications determine the location of the OCSP responder by using the Authority Information Access Extension in the certificate being validated. (Certificate Management System enables you to add this extension to certificates. For details, see “Configuring Policy Rules for a Subsystem” on page 613.)

The revocation-status-verification process has two parts:

1. When a certificate's status needs to be verified, the OCSP client (an OCSP-compliant application) sends a request to the OCSP responder for verification and waits for a response from the responder.

The OCSP request that the client submits generally contains all the information required by the responder to identify the certificate whose status it needs to determine.

(Consider this process is similar to a cashier scanning your credit card and waiting for a response from the credit-card processing unit. The scanning unit sends identifying information, such as the credit card number, its type, validity period, and so on.)

2. Upon receipt of the request, the OCSP responder determines if the request contains all the information required by the responder to process it.

- If the request lacks any information required by the responder to process it or if the responder is not configured to provide the requested service to the client, the responder sends a rejection notification to the client. The responder also writes an appropriate error message to its log file.
- If the request meets all the criteria, the responder returns a response to the client that requested it: it checks its list of revoked certificates for the one whose status is being requested, verifies its status, composes a report, signs the report, and sends the report to the client.

Note that every response that the client receives, including a rejection notification, is digitally signed by the responder; the client is expected to verify the signature to ensure that the response came from the responder to which it submitted the request. The key the responder uses to sign the message depends on how the OCSP responder is deployed in a PKI setup. RFC 2560 recommends that the key used to sign the response belong to one of the following:

- The CA that issued the certificate and whose status is being verified by the responder.
- A responder whose public key, which corresponds to the private key it uses to sign responses, is trusted by the client. Such a responder is called a *trusted responder*.
- A responder that holds a specially marked certificate issued to it directly by the CA that revokes the certificates and publishes the CRL. Possession of this certificate by a responder indicates that the CA has authorized the responder to issue OCSP responses for certificates revoked by the CA. Such a responder is called a *CA-designated responder* or a *CA-authorized responder*.

Certificate Management System has a built-in OCSP responder and allows you to request OCSP responder certificates. The end-entity interface of both Registration Manager and Certificate Manager includes a form that allows you to manually request a certificate for the OCSP responder. The default enrollment form includes all the attributes (for example, `HTTP_PARAMS.certType==ocspResponder`) that identify the certificate as an OCSP responder certificate. The required extensions, such as `OCSPNoCheck` and `OCSPSigning`, can be added to the certificate when the certificate request is subjected to policy checking; see “Configuring Policy Rules for a Subsystem” on page 613.

The OCSP response that the client receives indicates the current status of the certificate as determined by the OCSP responder. The response could be any of the following:

- **Good or Verified**—specifying a positive response to the status inquiry. At a minimum, this positive response indicates that the certificate has not been revoked, but it does not necessarily mean that the certificate was ever issued or that the time at which the response was produced is within the certificate's validity interval. Response extensions may be used to convey additional information on assertions made by the responder regarding the status of the certificate such as positive statement about issuance, validity, etc.
- **Revoked**—specifying that the certificate has been revoked, either permanently or temporarily.
- **Unknown**—specifying that the OCSP responder doesn't know about the certificate whose status is being requested by the client.

Based on the status, the client decides whether to validate the certificate.

How to Get an OCSP Responder?

To aid you in the process of setting up a OCSP-compliant PKI setup, Certificate Management System provides two options:

- Use the OCSP-service feature built into the Certificate Manager
- Use the CMS OCSP responder, named Online Certificate Status Manager

Read the sections that follow and decide which method is suitable for your PKI setup.

How Certificate Manager's OCSP-Service Feature Works

The Certificate Manager has a built-in OCSP-service feature, which when configured, can be used by OCSP-compliant clients to directly query the Certificate Manager about the revocation status of the certificate being validated.

When queried for the revocation status of a certificate, the Certificate Manager looks up its internal database for the certificate, checks its status, and accordingly responds to the client. Since the Certificate Manager has real-time status of all certificates it has issued, this method of revocation checking is most accurate. However, because the Certificate Manager can only check its own internal database, revocation checking is limited to certificates issued by that Certificate

Manager. That is, clients can verify only those certificates that are issued by the Certificate Manager. In addition, you also need to keep the Certificate Manager's nonSSL end-entity port enabled because the server can service OCSP requests only via its HTTP port.

If your PKI setup is large, containing a hierarchy of multiple Certificate Managers (root/subordinate CAs), using the Certificate Manager for validating certificates may not be suitable for you. However, if your PKI setup large, but is made up of cloned CAs, you should be able to use the Certificate Manager's built-in OCSP service feature. For information about cloning Certificate Managers, see "Cloning a Certificate Manager" on page 288.

For step-by-step instructions to set up an OCSP-compliant PKI setup using the Certificate Manager, see "Setting Up a Certificate Manager with OCSP Service" on page 719.

How Online Certificate Status Manager Works

In addition to the built-in OCSP service feature, the Certificate Manager can also publish CRLs to an OCSP-compliant online validation authority (or server). If you install the CMS OCSP responder, Online Certificate Status Manager, you can configure one or more Certificate Managers to publish their CRLs to the Online Certificate Status Manager. The Online Certificate Status Manager stores each Certificate Manager's CRL in its internal database and uses the appropriate CRL to verify the revocation status of a certificate when queried by an OCSP-compliant client. This enables you to issue all client certificates in your PKI with the Authority Information Access extension pointing to one location, the location at which the Online Certificate Status Manager is waiting to service OCSP requests; to validate a certificate, irrespective of which Certificate Manager has issued the certificate, an OCSP-complaint client need to just query one server.

You can configure the Certificate Manager to generate and publish CRLs whenever a certificate is revoked and at specified intervals, say every 20 minutes. Because the purpose of setting up an OCSP responder is to facilitate real-time verification of certificates, you should configure the Certificate Manager to generate and publish the CRL to the Online Certificate Status Manager every time a certificate is revoked—configuring the Certificate Manager to publish CRLs at specific intervals would negate the very purpose for which it's being done because the CRL the Online Certificate Status Manager would look up during verification would always be outdated. It's important to note that if the CRL is large, the Certificate Manager could take a considerable amount of time to publish the CRL.

As explained earlier, the Online Certificate Status Manager stores each Certificate Manager's CRL in its internal database and uses it as the default CRL store for verifying certificates. You can also configure the Online Certificate Status Manager to use the CRL published to an LDAP directory. If you do so, the Online Certificate Status Manager uses the CRL published to the LDAP directory, instead of the CRL in its internal database.

For step-by-step instructions to set up an OCSP-compliant PKI setup using the Online Certificate Status Manager, see "Setting Up a Remote OCSP Responder" on page 732.

How to Get OCSP-Compliant Clients?

As mentioned in the preceding section, in addition to a CA and an OCSP responder, you also need OCSP-compliant clients if you want to set up an OCSP-compliant PKI setup. For this purpose, you can use clients such as Netscape 6 or Netscape Communicator with Netscape Personal Security Manager.

Personal Security Manager is an OCSP-compliant security plug-in module for Communicator 4.7x versions. The module, in addition to many other features, enables Communicator to check certificate validity in real time using the OCSP protocol: it enables the client to read the Authority Information Access extension in a certificate, locate the OCSP responder specified by the extension, request the revocation status of the certificate from the OCSP responder, and use the response to validate the certificate. For a brief introduction to Personal Security Manager, see page 39.

NOTE If you're using Netscape 6 as your client, you don't need to install Personal Security Manager; it's integrated into Netscape 6 as its default security component. In other words, OCSP is a built-in feature in Netscape 6. For more information about Netscape 6, check this site: <http://home.netscape.com/browsers/>

Setting Up a Certificate Manager with OCSP Service

The Certificate Manager has a built-in OCSP service feature that can be used by OCSP-compliant clients to do real-time verification of certificates issued by the Certificate Manager. This section explains how to setup an OCSP-compliant PKI setup using the Certificate Manager's OCSP-service feature.

- Step 1. Before You Begin
- Step 2. Install OCSP-Compliant Client
- Step 3. Enable Certificate Manager's HTTP Port
- Step 4. Enable Certificate Manager's OCSP Service
- Step 5. Configure Certificate Manager for Extensions
- Step 6. Restart the Certificate Manager
- Step 7. Test Your CA's OCSP Service Setup

Step 1. Before You Begin

Before you start setting up a Certificate Manager to service OCSP requests, do this:

- If you are unfamiliar with Online Certificate Status Protocol (OCSP), read the PKIX draft RFC 2560 available at this web site:
<http://www.ietf.org/rfc/rfc2560.txt>
- Read section "What's an OCSP-Compliant PKI Setup?" on page 714. Decide whether you want to use the OCSP-compliant security plug-in module for Netscape Communicator, Personal Security Manager; this plug-in enables Netscape Communicator 4.7x to query the Online Certificate Status Manager using the OCSP protocol. Netscape 6 has Personal Security Manager built into it.
- Check whether you've installed the Certificate Manager, a CMS CA. If you haven't, first identify a host machine for installing it and then follow the installation instructions in Chapter 6, "Installing Certificate Management System" to install it. During installation, note the port numbers you assign to the Certificate Manager.
- Make sure the Certificate Manager is started. Also keep the iPlanet Console login information for the Certificate Manager handy; you'll need this to verify or make changes to their configuration.

- Read “Publishing of CRLs” on page 634. Determine whether you want the Certificate Manager to publish version 1 or version 2 CRLs to the directory. If you decide to publish version 2 CRLs, read Chapter 4, “Certificate Extension Plug-in Modules” of *CMS Plug-Ins Guide* and determine the CRL extensions you want the Certificate Manager to set; you will be required to configure the server to set these extensions.

Step 2. Install OCSP-Compliant Client

If you don't want to install Personal Security Manager, skip to the next step, “Step 5. Configure Certificate Manager for Required Extension Policies” on page 745.

If you decided to install Personal Security Manager:

1. Download the latest version of Personal Security Manager from the web site (<http://www.netscape.com>) to the machine on which you have Netscape Communicator, version 4.7x, installed.
2. Locate the Release Notes ([release_notes.html](#)). It explains how to install the product and lists known issues and restrictions. You must read this first for installation instructions.

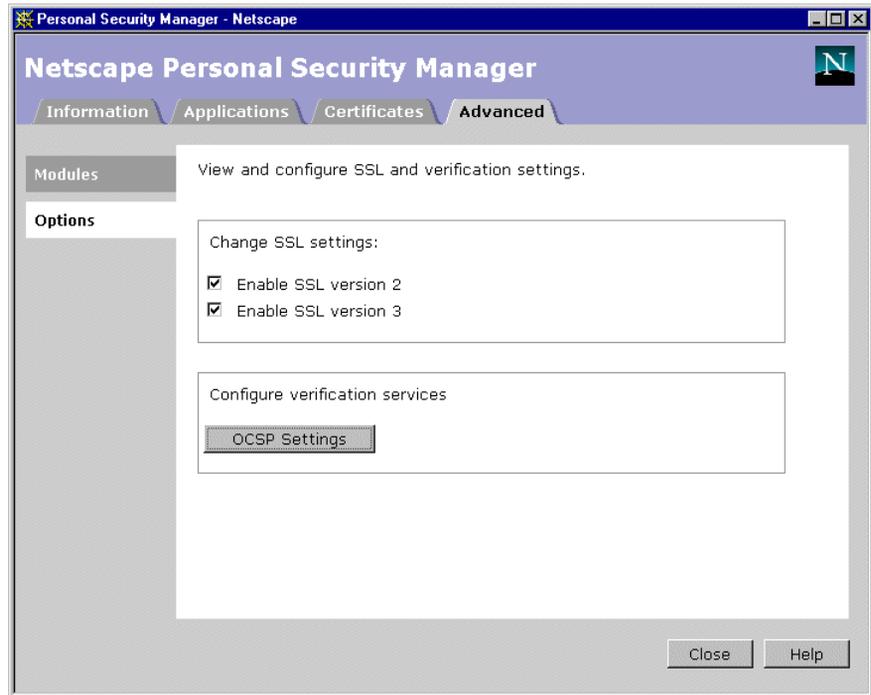
Make sure you also have the `cmjavascriptapi.html` file handy. It describes a JavaScript API for performing user certificate management operations within a client. The JavaScript runs in the context of an enrollment page served by a Certificate Manager or Registration Manager, enabling it to instruct the client to perform PKI operations, such as key generation, certificate-request generation, key archival, import of user certificates, key recovery, and revocation requests. You'll need to refer to this when setting up a Data Recovery Manager for key archival and recovery, which is covered in Chapter 22, “Setting Up Key Archival and Recovery.”

3. Follow the instructions in the release notes and install the product.

For example, in a Windows NT system, you can install Personal Security Manager by entering the path to the `psm14_win32.jar` file in the browser's URL area. On a Solaris system, you can unzip the file by running `gunzip psm_14_solaris2.6.tar.gz`, untar the file by running `tar xvf psm14_solaris2.6.tar`, and then install Personal Security Manager by running `psm-install`.

4. Verify that Personal Security Manager is installed.

In the menu bar, click Communicator, and from the Tools menu, select Security Info. You should see the Personal Security Manager interface.



5. Configure Personal Security Manager to verify certificates by using the OCSP service URL identified by the Authority Information Access extension in certificates.
- a. Select the Advanced tab.
 - b. On the left side, select Options, and then click the OCSP Settings button.
 - c. In the OCSP Settings window, select the “Use OCSP to verify only certificates that specify an OCSP service URL.” option and click OK.

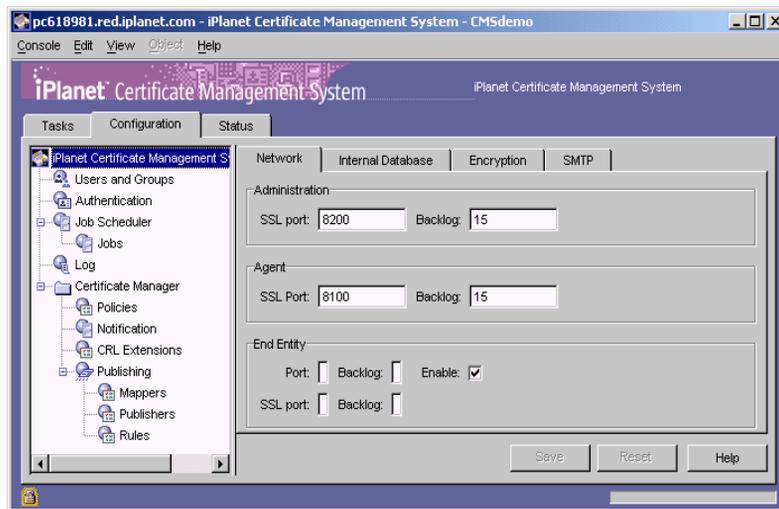
Step 3. Enable Certificate Manager’s HTTP Port

The Certificate Manager services OCSP requests via its nonSSL (HTTP) end-entity port; see “End-Entity Ports” on page 383. If you’ve disabled the port, you must enable it so that OCSP-compliant clients can successfully query the Certificate Manager for the revocation status of a certificate.

To enable the end-entity port used by the Certificate Manager for non-SSL communications:

1. Log in to the CMS window for the Certificate Manager (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.

The Network tab appears.



3. In the End Entity section, select the “Enable” option, and in the adjoining field, type a TCP/IP port number that is unique on the host system. Leave all the other options as they are.
4. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. Don't restart the server yet; you can do that after you've made all the changes.

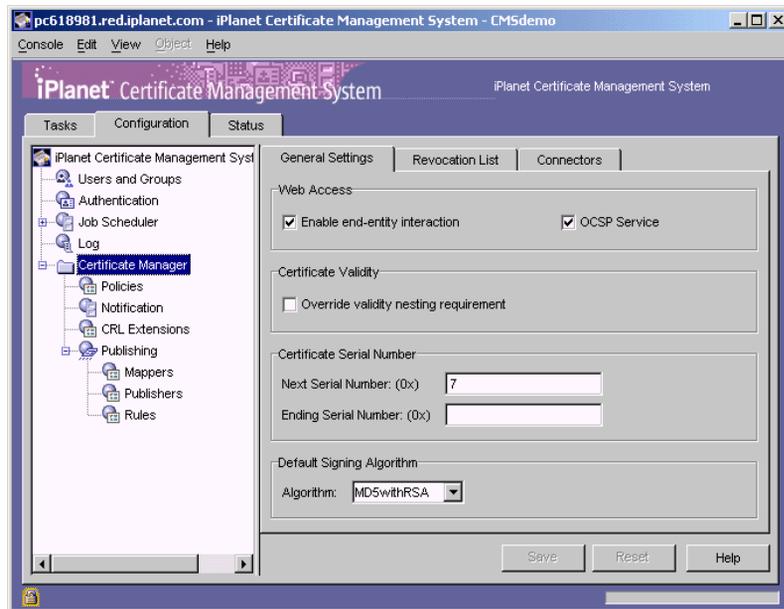
Step 4. Enable Certificate Manager's OCSP Service

During the installation of a Certificate Manager, you are given an opportunity to specify whether you want to enable Certificate Manager's OCSP service. If you chose to enable it, you just need to verify that the OCSP service is still on. If you chose to keep the service disabled, you need to follow the instructions below and enable the service.

To enable a Certificate Manager's OCSP service:

1. In the navigation tree, select Certificate Manager.

The General Setting tab appears.



2. In the Web Access section, select the "OCSP Service" option. Leave all the other selections as they are or make changes that you deem necessary; see "Step 6. Enable End-Entity Interaction" on page 563.
3. To save your changes, click Save.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. Don't restart the server yet; you can do that after you've made all the changes.

Step 5. Configure Certificate Manager for Extensions

In order for OCSP-compliant clients to query the Certificate Manager about the revocation status of a certificate, the certificate being validated must contain the Authority Information Access extension pointing to the location at which the Certificate Manager listens for OCSP service requests. For details about the Authority Information Access extension, see section “AuthInfoAccessExt Plug-in Module” of *CMS Plug-Ins Guide*.

The Certificate Manager can add an extension to a certificate it issues only if the corresponding policy is enabled and configured properly. Hence, before issuing the OCSP-compliant client certificate, you must verify that the Certificate Manager is configured with the appropriate policy rules to add the required extensions to these certificates.

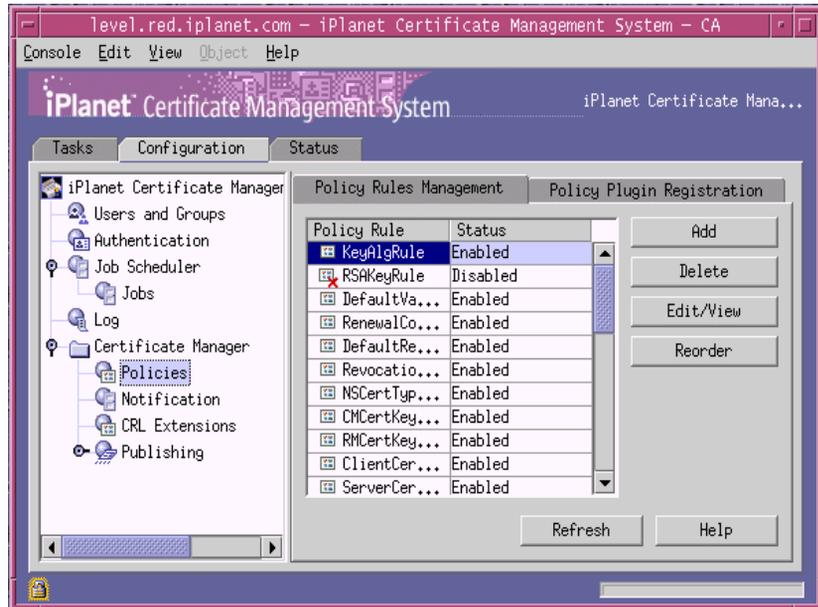
- During the installation of a Certificate Manager, if you chose to enable its OCSP service, a default policy rule (named `AuthInfoAccessExt`) is created with correct attributes for adding the Authority Information Access extension to certificates the Certificate Manager will issue following installation. If you didn't make any changes to the policy configuration of the Certificate Manager, you probably don't need to do anything.
- If you installed the Certificate Manager's with its OCSP service feature disabled, a default policy rule (named `AuthInfoAccessExt`) is created, but it may not have the correct attributes for adding the Authority Information Access extension to certificates.

In either case, it's advisable that you check the status of the said policy rule, and update it if required. Also, for testing whether your OCSP-compliant clients can verify revocation status of certificates by querying the OCSP responder, you will be issuing a client certificate containing the Authority Information Access extension to Personal Security Manager you installed.

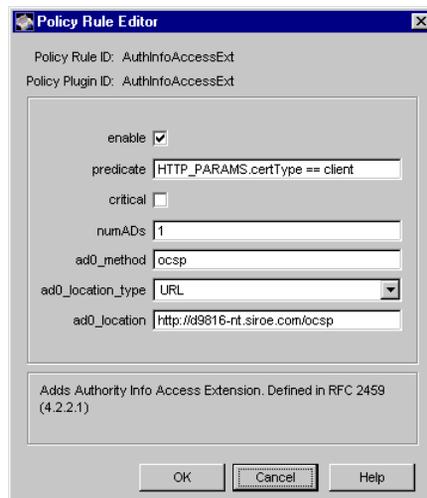
To verify the status of policy rules that enable the Certificate Manager to add the extensions required in an OCSP-compliant client certificate:

1. In the navigation tree, select Certificate Manager, and then select Policies.

The Policy Rules Management tab appears. It lists configured policy rules.



2. In the Policy Rule list, select the rule named AuthInfoAccessExt and click Edit; this rule was created by default during installation.



3. Make sure the following values are assigned:

Enable. Checked or selected.

predicate. Shows HTTP_PARAMS.certType==client.

critical. Unchecked.

numADs. Shows 1.

ad0_method. Shows ocsp or 1.3.6.1.5.5.7.48.1.

ad0_location_type. Shows URL.

ad0_location. Shows the complete path to the location where the Certificate Manager listens to calls from OCSP-compliant clients. The path should be in this format: `http://<hostname>:<nonSSL_end_entity_port>/ocsp`

If the end-entity port number is 80, you need not specify it in the URL. For example, if the hostname of your Certificate Manager is `demoCA.siroe.com` and the end-entity port number is 8000, the URL to type in the field would be:

```
http://demoCA.siroe.com:8000/ocsp
```

If you need details about any of the configuration parameters, click the Help button.

4. Click OK.

You are returned to the Policy Rules Management tab.

5. Make any other policy changes, if necessary.

6. Click Refresh.

The Certificate Manager is ready to request client certificates with Authority Information Access extension.

Step 6. Restart the Certificate Manager

For all your changes to take effect, you must restart the Certificate Manager. You can use the CMS window to restart the Certificate Manager.

1. Select the Tasks tab.
2. Click Restart the Server.

When you restart Certificate Management System, you are prompted to supply the single sign-on password for the server.

3. Type the single sign-on password you specified during installation and click OK.

Certificate Management System won't restart until you provide this password. For more information, see "Required Start-up Information" on page 322.

Step 7. Test Your CA's OCSP Service Setup

To test whether the Certificate Manager can service OCSP requests properly, follow these steps:

- Step A. Turn On Revocation Checking in the Browser
- Step B. Request a Certificate
- Step C. Approve the Request
- Step D. Download the Certificate to the Browser
- Step E. Make Sure the CA is Trusted by the Browser
- Step F. Verify the Certificate in the Browser
- Step G. Check the Status of Certificate Manager's OCSP Service
- Step H. Revoke the Certificate
- Step I. Verify the Certificate in the Browser
- Step J. Check the Certificate Manager's OCSP Service Status Again

Step A. Turn On Revocation Checking in the Browser

To ensure that Personal Security Manager (the OCSP-compliant client) is configured to verify the revocation status of certificates using the OCSP protocol:

1. Open a web browser window.
2. Open the Personal Security Manager interface.

In Communicator version 4.7, you can open this window by clicking the Security button in the navigation bar. Alternatively, you can also open this window by selecting Communicator from the main menu, selecting Tools, and then selecting Security Info.

3. Select the Advanced tab, and then in left pane, select Options.

4. Click the OCSP Settings button.
The OCSP Setting window appears.
5. Select the “Use OCSP to verify only certificates that specify an OCSP service URL” option, and click OK.
6. Click the Close button.

Step B. Request a Certificate

The steps outlined below explain how to request a client certificate from the Certificate Manager using the *manual* enrollment method. If you’ve configured the Certificate Manager for automated certificate issuance, for example for directory-based enrollment, you may use the appropriate form and request a certificate.

To request a client or personal certificate from the Certificate Manager:

1. Go to the end-entity interface of the Certificate Manager you configured (or to the Registration Manager that’s connected to this Certificate Manager).

The URL is in this form: `https://<hostname>:<end_entity_HTTPS_port>` or `http://<hostname>:<end_entity_HTTP_port>`

2. In the left frame, under Browser, click Manual.

This opens the manual enrollment form.

3. Fill in all the values and submit the request.

The client prompts you to enter the password for your key database.

4. When you enter the correct password, the client generates the key pairs.

Do not interrupt the key-generation process.

Step C. Approve the Request

Skip this step if you requested the certificate using any of the automated enrollment methods. Complete this step if you used the manual enrollment form for requesting the certificate; the request you submitted is waiting in the agent queue for approval by an agent.

To approve the request:

1. Go to the Certificate Manager’s Agent Services interface.

The URL is in this format: `https://<hostname>:<agent_port>`

2. In the left frame, click List Requests.

3. In the form that appears, select the “Show pending requests” option and click Find.
4. In the list of pending requests, identify the request you submitted and click Details.
5. Check the request to make sure that it has all the required attributes of a client certificate, including the Authority Information Access extension.
6. Scroll to the bottom of the request form, and approve the request.

You should see a confirmation page indicating that the certificate has been issued. Don't close the page until after you complete the next step.

Step D. Download the Certificate to the Browser

To download the certificate into the certificate database of Personal Security Manager:

1. In the confirmation page, scroll down to the section that says “Installing this certificate in a client.”
2. Check the certificate details for the required extensions.
3. Follow the on-screen instructions and download the certificate to your browser's certificate database.

(An alternative way to download the certificate is to go to the Retrieval tab of the end-entity services interface, search for the certificate, and download the certificate.)

Step E. Make Sure the CA is Trusted by the Browser

When you downloaded the client certificate to the browser, the Certificate Manager's certificate chain also was downloaded to the browser's certificate database. Make sure that the Certificate Manager's CA signing certificate is trusted in the browser's certificate database; this is required for proper chaining during certificate validation.

1. In the browser, open the Personal Security Manager interface.

In Communicator version 4.7, you can open this window by selecting Communicator from the main menu, selecting Tools, and then selecting Security Info.

2. Select the Certificates tab and, in the left pane, click Authorities.

The list of CA certificates currently stored in the browser's certificate database appears.

3. Locate the Certificate Manager's CA signing certificate, select it, and click Edit.
The Edit Security Certificate Settings window appears.
4. Make sure all the three options are selected and click OK.

Step F. Verify the Certificate in the Browser

To verify that the certificate has been downloaded into the certificate database of Personal Security Manager:

1. Click the Certificates tab and, in the left pane, click Mine.
You should see the names of all the client certificates, including the one you just downloaded, stored in the browser's certificate database.
2. Select the name of the certificate you just downloaded and click View.
In the View Security Certificate dialog box that appears, look for a message that says that the certificate is verified; generally, it's at the top.

Step G. Check the Status of Certificate Manager's OCSP Service

The Certificate Manager's Agent interface contains a form that enables you to check the Certificate Manager's OCSP-service status, such as how many request its received and so on. To go to the Certificate Manager's status page and verify the number of requests it has processed so far:

1. Go to the web browser window and enter the URL for the Certificate Manager's Agent interface.
The URL is in this format: `https://<hostname>:<port>`. The Certificate Manager Agent Services interface appears.

2. In the left frame, click OCSP Service.

The resulting form should show information about the Certificate Manager's OCSP service-related activity since it was last started.

Note the value assigned to the "OCSP Requests Since Startup" field. It should show a value of one (1), a proof that the OCSP-compliant client, Personal Security Manager, queried the Certificate Manager for revocation status of a certificate.

Step H. Revoke the Certificate

To revoke the certificate you issued:

1. Go to the end-entity interface for the Certificate Manager you configured (or to the Registration Manager that's connected to this Certificate Manager).

Be sure to go to the HTTPS interface. The URL is in this form:

```
https://<hostname>:<end_entity_HTTPS_port>
```

2. Select the Revocation tab.
3. In the left frame, click User Certificate.
The User Certificate Revocation form appears.
4. In the Revocation Reason section, select Unspecified and click Submit.
The client shows the "Select a Certificate" dialog box and prompts you to choose the certificate you want to revoke.
5. Select the certificate you downloaded and click OK.
The Certificate Manager revokes the certificate and updates the certificate status in its internal database.

Step I. Verify the Certificate in the Browser

To verify that the certificate has been revoked:

1. Open the Personal Security Manager interface.
2. Select the Certificates tab and then click Mine.
3. Select the certificate you revoked and click View.

In the View Security Certificate dialog box that appears, look for a message that says that the certificate could not be verified.

Step J. Check the Certificate Manager's OCSP Service Status Again

Check the Certificate Manager's OCSP-service status again to verify that these things happened:

- The browser sent an OCSP query to the Certificate Manager (this response was initiated when you clicked the View button).
- The Certificate Manager sent an OCSP response to the browser.
- The browser used that response to validate the certificate and informed you of its status (that the certificate could not be verified).

To check the Certificate Manager's OCSP-service status for verification:

1. Go to the Certificate Manager's status page.
2. Reload the page (hold down the Shift key and click on the browser's Reload icon.)
3. Compare the information to the one you noted in Step G above.

The updated statistics should indicate that Personal Security Manager queried the Certificate Manager about the status of the certificate and in response, the Certificate Manager informed Personal Security Manager that the certificate is revoked.

Setting Up a Remote OCSP Responder

You can configure a Certificate Manager to publish CRLs to an online certificate validation authority, such as the one included with Certificate Management System, and then issue end-entity certificates with Authority Information Access extension pointing to the location at which the OCSP responder waits for queries about revocation status of certificates.

This section explains how to set up a Certificate Manager functioning as a root CA to publish CRLs to a remote Online Certificate Status Manager and configure OCSP-compliant clients to query the Online Certificate Status Manager for revocation status of certificates being validated.

The procedure for setting up a Certificate Manager functioning as a subordinate CA to publish CRLs to a remote Online Certificate Status Manager would be the same, except that you would have to perform extra steps to make sure the that CA chain verification takes place smoothly. For example:

- If the Online Certificate Status Manager's SSL server certificate is signed by the same root CA that signed the subordinate Certificate Manager's certificates, then you need to mark the root CA as a trusted CA in the subordinate Certificate Manager's certificate database.
- If the Online Certificate Status Manager's SSL server certificate is signed by a different root CA, then you need to import the root CA certificate into the subordinate Certificate Manager's certificate database and mark it as a trusted CA.

To import a CA certificate into the certificate database of a subordinate Certificate Manager, you can use the Certificate Setup Wizard. For instructions, see “Using the Wizard to Install a Certificate or Certificate Chain” on page 493. After you install the certificate, you can follow the instructions in see “Changing the Trust Settings of a CA Certificate” on page 526 to trust the CA certificate you imported.

- Step 1. Before You Begin
- Step 2. Install an OCSP-Compliant Client
- Step 3. Identify the CA to the OCSP Responder
- Step 4. Configure the Certificate Manager to Publish CRLs
- Step 5. Configure Certificate Manager for Required Extension Policies
- Step 6. Configure the Online Certificate Status Manager
- Step 7. Restart the Certificate Manager
- Step 8. Restart the Online Certificate Status Manager
- Step 9. Verify Certificate Manager and Online Certificate Status Manager Connection
- Step 10. Test Your OCSP Responder Setup

Note that the Online Certificate Status Manager can be configured to receive CRLs from more than one Certificate Manager. If your deployment has many CAs and you want all of them to publish CRLs to the same Online Certificate Status Manager, you should repeat the above steps for each Certificate Manager.

Step 1. Before You Begin

Before you configure a Certificate Manager (CA) to publish CRLs to an OCSP responder, do the following:

- If you are unfamiliar with Online Certificate Status Protocol (OCSP), read the PKIX draft RFC 2560 available at this site:
<http://www.ietf.org/rfc/rfc2560.txt>
- Read section “What’s an OCSP-Compliant PKI Setup?” on page 714.

- Check whether you've installed the Online Certificate Status Manager, the OCSP responder provided with Certificate Management System. If you haven't, first identify a host machine for installing it and then follow the installation instructions in Chapter 6, "Installing Certificate Management System" to install it. During installation, note the port numbers you assign to the Online Certificate Status Manager.
- Check whether you have deployed any OCSP-compliant clients. If you haven't, determine whether you want to use the OCSP-compliant security plug-in module for Netscape Communicator, Netscape Personal Security Manager. For details, see "How to Get OCSP-Compliant Clients?"
- Keep the iPlanet Console login information for the Certificate Manager and Online Certificate Status Manager handy; you'll need this to verify or make changes to their configuration.
- Read section "OCSPublisher Plug-in Module" in Chapter 6, "Publisher Plug-in Modules" of *CMS Plug-Ins Guide*.
- Read "Publishing of CRLs" on page 634. Determine whether you want the Certificate Manager to publish version 1 or version 2 CRLs to the directory. If you decide to publish version 2 CRLs, read Chapter 4, "Certificate Extension Plug-in Modules" of *CMS Plug-Ins Guide* and determine the CRL extensions you want the Certificate Manager to set; you will be required to configure the server to set these extensions.
- Decide whether you want to configure your Online Certificate Status Manager to use its default database for CRLs or to use an LDAP directory. If you want the Online Certificate Status Manager to use the CRL published to the directory, make sure that the Certificate Manager is configured to publish CRLs to an LDAP directory. For details, see Chapter 19, "Setting Up LDAP Publishing."

Note the following information for the directory: the host name, port number, and port type—whether it's an SSL or nonSSL port. The Online Certificate Status Manager can communicate with the directory via SSL or nonSSL port.

Step 2. Install an OCSP-Compliant Client

Follow the instructions as appropriate.

- If you don't want to install Personal Security Manager, skip to the next step, "Step 3. Identify the CA to the OCSP Responder" on page 735.
- If you decided to install Personal Security Manager, follow the instructions in section "Step 2. Install OCSP-Compliant Client" on page 720 to install it.

Step 3. Identify the CA to the OCSP Responder

Before you configure a Certificate Manager to publish CRLs to the Online Certificate Status Manager, you must identify the Certificate Manager to the Online Certificate Status Manager. You do this by storing the Certificate Manager's *CA signing certificate* in the internal database of the Online Certificate Status Manager.

To locate the Certificate Manager's *CA signing certificate*, it might be useful to know whether it's self-signed or signed by another CA.

- If the certificate is self-signed, you can locate the certificate by searching for it in the Retrieval tab of Certificate Manager's end-entity interface.
- If the Certificate Manager is a subordinate CA, you can locate its signing certificate by listing the certificates in the CA certificate chain; you can download the CA chain from the Retrieval tab of a Certificate Manager's end-entity interface.

The steps below explain how to store the Certificate Manager's *CA signing certificate* in the internal database of the Online Certificate Status Manager:

1. Locate the Certificate Manager's CA signing certificate.

If the certificate is self-signed:

- a. Open a web browser window.
- b. Go to the Certificate Manager's end-entity interface. The URL is in `https://<hostname>:<SSL_port>` or `http://<hostname>:<port>` format.
- c. Select the Retrieval tab, and in the left frame, click List Certificates.
- d. In the resulting form, click List.
A list of certificates appear.
- e. Locate the Certificate Manager's CA signing certificate by looking at the subject name of the certificate.
Typically, the CA signing certificate is the first certificate the Certificate Manager issues.
- f. Click Details.
- g. In the resulting page, scroll to the section that says "Base 64 encoded certificate" and shows the CA signing certificate in its base-64 encoded format.

- h. Copy the base-64 encoded certificate, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines, to the clipboard or a text file.

The copied information should look similar to the following example:

```
-----BEGIN CERTIFICATE-----
MIICJzCCAZCgAwIBAgIBAzANBgkqhkiG9w0BAQQFADBCMSAwHgYDVQQKEXdOZ
XRzY2FwZSBDb21tdW5pYF0aW9uczngjhnMVQ2VydG1maWNhdGUgQXV0aG9yaX
R5MB4XDtk4MDgyNzE5MDAwMFOXDtk5MDIyMzE5MDAwMnBjdGngYoxIDAeBgNV
BAoTF05ldHNjYXB1IENvbW11bmljYXRpb25zMQ8wDQYDVQQLEWZQZW9wbGUx
zAVBgoJkiaJkIisZAEBEwdzdXByaXlhMRcwFQYDVQQDEw5TdXByaXlhIFNoZXR
0eTEjMCEGCSqGS1b3DbndgJA
-----END CERTIFICATE-----
```

If the certificate is signed by another CA:

- a. Open a web browser window.
- b. Go the Certificate Manager's end-entity interface. The URL is in `https://<hostname>:<SSL_port>` or `http://<hostname>:<port>` format.
- c. Select the Retrieval tab, and in the left frame, click Import CA Certificate Chain.
- d. In the resulting form, select the "Display certificates in the CA certificate chain for importing individually into a server" option.

A list of certificates appear.

- e. Locate the Certificate Manager's CA signing certificate in its base-64 encoded format.
- f. Copy the base-64 encoded certificate, including the -----BEGIN CERTIFICATE----- and -----END CERTIFICATE----- marker lines, to the clipboard or a text file.

The copied information should look similar to the following example:

```
-----BEGIN CERTIFICATE-----
MIICJzCCAZCgAwIBAgIBAzANBgkqhkiG9w0BAQQFADBCMSAwHgYDVQQKEXdOZ
XRzY2FwZSBDb21tdW5pYF0aW9uczngjhnMVQ2VydG1maWNhdGUgQXV0aG9yaX
R5MB4XDtk4MDgyNzE5MDAwMFOXDtk5MDIyMzE5MDAwMnBjdGngYoxIDAeBgNV
BAoTF05ldHNjYXB1IENvbW11bmljYXRpb25zMQ8wDQYDVQQLEWZQZW9wbGUx
zAVBgoJkiaJkIisZAEBEwdzdXByaXlhMRcwFQYDVQQDEw5TdXByaXlhIFNoZXR
0eTEjMCEGCSqGS1b3DbndgJARYU
-----END CERTIFICATE-----
```

2. Go to the Online Certificate Status Manager's Agent interface. The URL is in this format: `https://<hostname>:<port>`.

The Online Certificate Status Manager Agent Services interface appears.

3. In the left frame, click Add Certificate Authority.
4. In the resulting form, paste the encoded CA signing certificate inside the text area labeled "Base 64 encoded certificate (including header and footer)."
5. Click Add.

The certificate is added to the internal database of the Online Certificate Status Manager.

6. To verify that the certificate is added successfully, in the left frame, click List Certificate Authorities.

The resulting form should show information about the Certificate Manager (CA) you just added. Note the values assigned to the "This Update," "Next Update," and "Requests Served Since Startup" fields. All three fields should show a value of zero (0).

Keep the web browser window open. You'll need to use the same form later for verifying that the Certificate Manager can communicate with the Online Certificate Status Manager.

Step 4. Configure the Certificate Manager to Publish CRLs

In this step, you configure the Certificate Manager to publish CRLs to the Online Certificate Status Manager. Note that a configured Certificate Manager will publish the CRL to the Online Certificate Status Manager, replacing the old CRL with the new one; the old CRL is not saved.

To configure a Certificate Manager to publish CRLs to the directory, follow these steps:

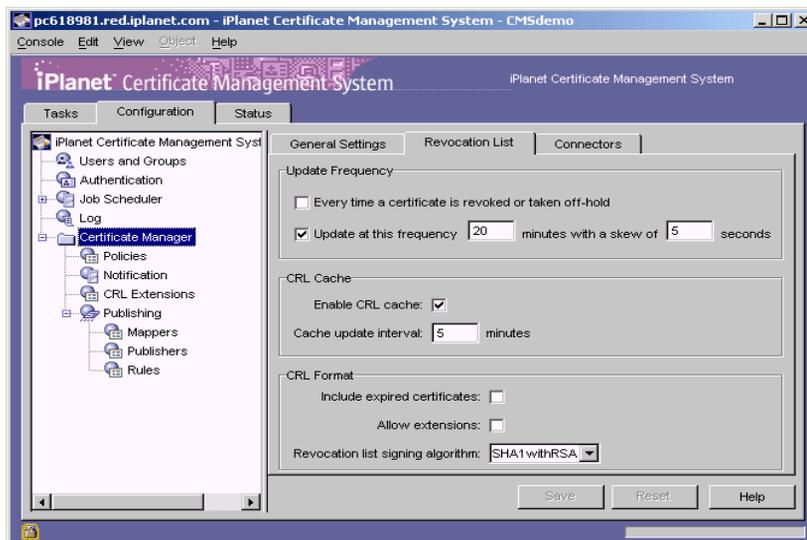
- Step A. Specify CRL Format and Publishing Interval
- Step B. Set the CRL Extensions
- Step C. Create a Publisher for the CRL
- Step D. Create a Publishing Rule for the CRL
- Step E. Make Sure Publishing is Enabled

Step A. Specify CRL Format and Publishing Interval

You can specify information, such as the publishing interval, the CRL version (whether to include CRL extensions), and the signing algorithm the Certificate Manager should use for signing the CRL object.

To specify CRL details:

1. Log in to the CMS window for the Certificate Manager (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select Certificate Manager, and then in the right pane, select the Revocation List tab.



4. In the Update Frequency section, select the “Every time a certificate is revoked or taken off-hold” option. This option enables the Certificate Manager to generate the CRL every time it revokes a certificate.

Keep in mind that the Certificate Manager attempts to publish the CRL to the configured Online Certificate Status Manager whenever the CRL is generated, in this case, every time a certificate is revoked. Publishing a CRL can be time consuming if the CRL is large. Configuring the Certificate Manager to publish CRLs every time a certificate is revoked may engage the server for a considerable amount of time.

5. In the CRL Cache section, specify whether to enable CRL caching:
 - Enable cache.** Check this box to enable CRL caching. Leave the box unchecked if you don't want the server to maintain a cache.
 - Update interval.** If you enabled caching, type the interval for updating the cache.
6. In the CRL Format section, specify the format for publishing the CRL:
 - Include expired certificates.** Check this box if you want the server to include revoked certificates that have expired in the CRL.
 - Allow extensions.** Check this box if you want to allow extensions in the CRL. If you enable this option, the server generates and publishes CRLs conforming to X.509 version 2 standard. If you disable this option, the server generates and publishes CRLs conforming to X.509 version 1 standard. By default, the server publishes version 1 CRLs. If you enable this option, be sure to set the required CRL extensions as described in “Step B. Set the CRL Extensions” on page 739.
 - Revocation list signing algorithm.** Select the algorithm the server should use to sign the CRL. If the Certificate Manager's signing key type is RSA, select MD2 with RSA, MD5 with RSA, or SHA-1 with RSA. If the Certificate Manager's signing key type is DSA, select SHA-1 with DSA.
7. To save your changes, click Save.
 - If the changes you made require you to restart the server, you are prompted accordingly. However, don't restart the server yet; you can restart it after you've made all the required changes.

Step B. Set the CRL Extensions

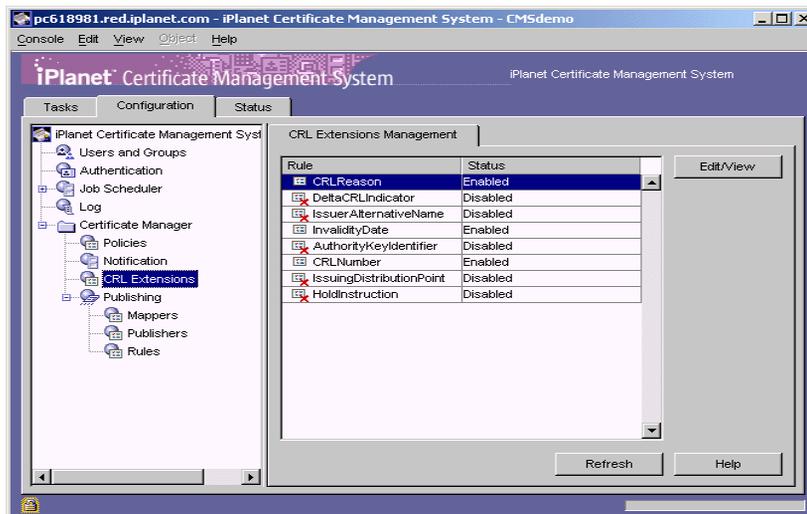
Complete this step only if you configured the Certificate Manager to publish version 2 CRLs—that is, you selected the “Allow extensions” option in “Step A. Specify CRL Format and Publishing Interval” on page 738.

During installation, the Certificate Manager creates default CRL extension rules; these are documented in *CMS Plug-Ins Guide*. Note that the server is configured to add the CRL Reason extension only; all the other rules are in the disabled state. In this step, you modify the default CRL extension rules to add the required CRL extensions.

To specify the CRL extensions the Certificate Manager should set:

1. In the navigation tree, under Certificate Manager, select CRL Extensions.

The right pane shows the CRL Extensions Management tab, which lists configured extensions.



2. To modify a rule, select it and then click Edit/View.
3. Change the information as appropriate.

Be sure to supply all the required values. Click the Help button for detailed information on individual parameters.

4. Click OK.
You are returned to the CRL Extensions Management tab.
5. To modify other rules, repeat steps 2 through 4.
6. Click Refresh to see the updated status of all the rules.

Step C. Create a Publisher for the CRL

Creating a publisher for the CRL involves creating an instance of the publisher module that enables the Certificate Manager to publish the CRL to the Online Certificate Status Manager. In the next step, described in “Step D. Create a Publishing Rule for the CRL” on page 742, you specify the publisher you create here.

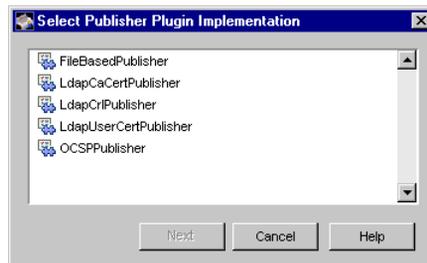
To create a publisher for the CRL:

1. In the navigation tree, click Publishers.

The right pane shows the Publishers Management tab, which lists configured publisher instances.

2. Click Add.

The Select Publisher Plugin Implementation window appears. It lists registered publisher modules.

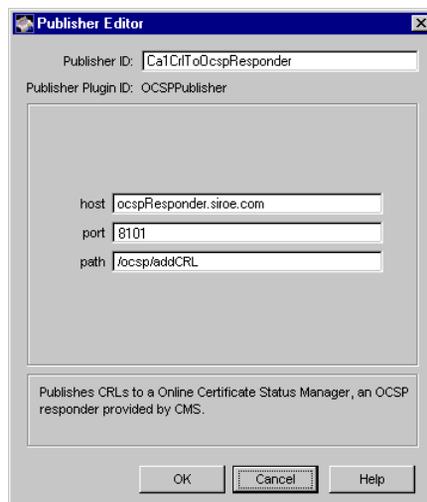


3. Select the module named OCSPublisher.

Only this publisher module enables the Certificate Manager to publish the CRL to the Online Certificate Status Manager. (If you have registered any custom publisher modules, they too will be available for selection.)

4. Click Next.

The Publisher Editor window appears.



5. Enter the appropriate information:

Publisher ID. Type a name for the rule; use an alphanumeric string with no spaces. For example, `Ca1CrlToOcsPResponder`.

host. Type the fully-qualified host name of the Online Certificate Status Manager. The name must be in the form `<machine_name>.<your_domain>.<domain>`. For example, `ocspResponder.siroe.com`.

port. Type the Online Certificate Status Manager's agent port number. For example, `8101`.

path. Make sure this field shows the default path, `/ocsp/addCRL`. If necessary, type it in.

6. Click OK.

The Publishers Management tab appears, listing the new publisher.

Step D. Create a Publishing Rule for the CRL

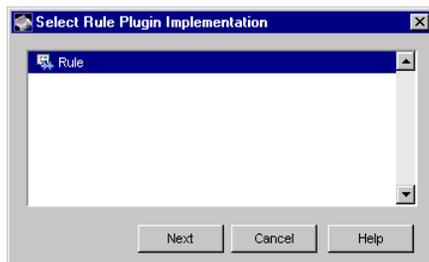
Creating a publishing rule for the CRL involves creating a rule that uses the publisher instance that you created in the previous step. To create a publishing rule:

1. In the navigation tree, click Rules.

The right pane shows the Rules Management tab, which lists any currently configured publishing rules.

2. Click Add.

The Select Rule Plugin Implementation window appears. It lists registered modules that enable creating of publishing rules.



3. Select the module named `Rule`.

This is the default module. (If you have registered any custom modules, they too will be available for selection.)

4. Click Next.

The Rule Editor window appears.

5. Enter the appropriate information:

The screenshot shows the 'Rule Editor' dialog box with the following fields and values:

- Rule ID: PublishCa1Cr1ToOcspResponder
- Rule Plugin ID: Rule
- type: crl
- predicate: (blank)
- enable:
- mapper: <NONE>
- publisher: Ca1Cr1ToOcspResponder

At the bottom of the dialog, there is a note: "Use the mapper to find the ldap dn to publish the certificate or crl". Below the note are three buttons: OK, Cancel, and Help.

Rule ID. Type a name for the rule; be sure to use an alphanumeric string with no spaces. For example, PublishCa1Cr1ToOcspResponder.

type. Select crl.

predicate. Leave this field blank.

enable. Select this option.

mapper. Select <NONE>.

publisher. Select the publisher you added for publishing the CRL to the Online Certificate Status Manager. For example, Ca1Cr1ToOcspResponder.

6. Click OK.

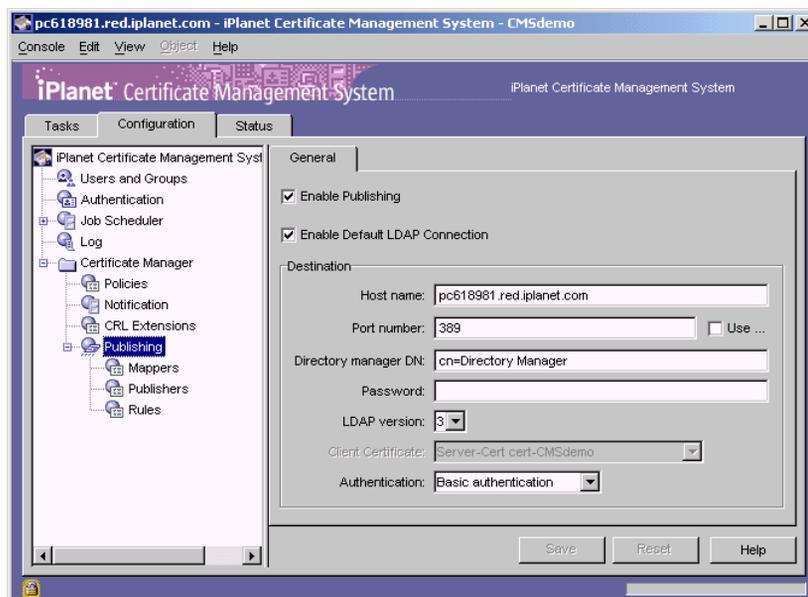
The Rules Management tab appears, listing the new rule.

Step E. Make Sure Publishing is Enabled

To make sure that the Certificate Manager is configured for publishing:

1. In the navigation tree, select Certificate Manager, then select Publishing.

The right pane shows the publishing details necessary for the server to publish to an LDAP-compliant directory, to files, or to an online validation authority.



2. Make sure that the Enable Publishing option is selected. If it is already selected, leave it as it is. If it isn't, select it.

(Leave the “Enable default LDAP connection” option as it is; it specifies that the Certificate Manager is configured to publish certificates and CRLs to an LDAP directory.)

3. If you changed anything, click Save to save the changes.

If the changes you made require you to restart the server, you are prompted accordingly. Don't restart the server yet. You can do that after you've made all the required changes.

Step 5. Configure Certificate Manager for Required Extension Policies

In order for OCSP-compliant clients to query the Online Certificate Status Manager about the revocation status of a certificate, the certificate being validated must contain the Authority Information Access extension pointing to the location at which the Online Certificate Status Manager listens for OCSP service requests. For details about the Authority Information Access extension, see section “AuthInfoAccessExt Plug-in Module” of *CMS Plug-Ins Guide*.

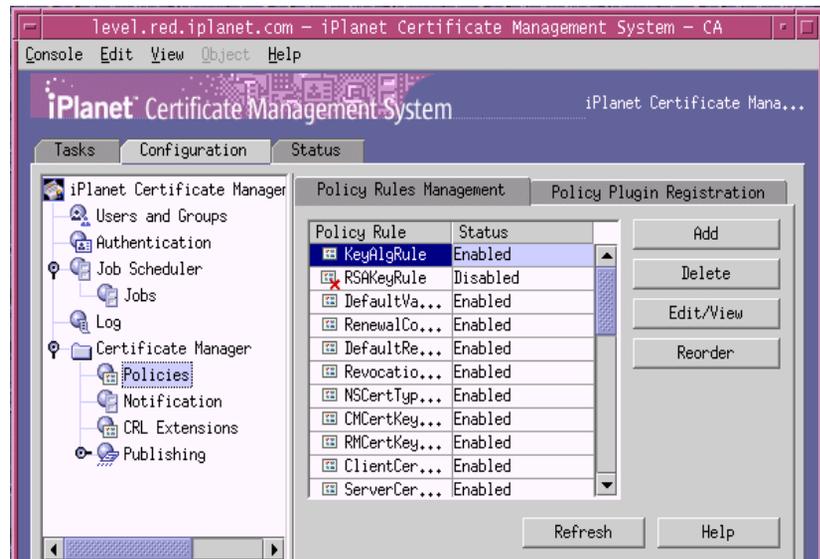
The Certificate Manager can add an extension to a certificate it issues only if the corresponding policy is enabled and configured properly. Hence, before issuing the OCSP-compliant client certificate, you must verify that the Certificate Manager is configured with the appropriate policy rules to add the required extensions to these certificates.

Also, for testing whether your OCSP-compliant clients can verify revocation status of certificates by querying the OCSP responder, you will be issuing a client certificate containing the *Authority Information Access* extension to Personal Security Manager you installed.

To verify the status of policy rules that enable the Certificate Manager to add the extensions required in an OCSP-compliant client certificate:

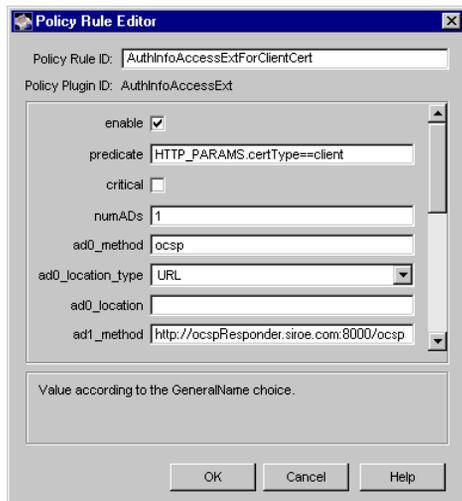
1. In the navigation tree, select Certificate Manager, and then select Policies.

The Policy Rules Management tab appears. It lists configured policy rules.



- In the Policy Rule list, select the rule named `AuthInfoAccessExt` and click Edit; this rule was created by default during installation.

The Policy Rule Editor window appears, showing how this rule is currently configured.



- Assign the following values:

Enable. Check this box.

predicate. Type `HTTP_PARAMS.certType==client`.

critical. Leave this option unchecked.

numADs. Type 1.

ad0_method. Type `ocsp` or `1.3.6.1.5.5.7.48.1`.

ad0_location_type. Select `URL`.

ad0_location. Type the complete path to the location where the Online Certificate Status Manager listens to calls from OCSP-compliant clients. The path should be in this format:

```
http://<hostname>:<end_entity_HTTP_port>/ocsp
```

For example, if the host name of your Online Certificate Status Manager is `ocspResponder.siroe.com` and the port number assigned to the non SSL end-entity port is 8000, the URL to type in the field would be:

```
http://ocspResponder.siroe.com:8000/ocsp
```

If you need details about any of the configuration parameters, click the Help button.

4. Click OK.

You are returned to the Policy Rules Management tab.

5. Make any other changes, if necessary.
6. Click Refresh.

The Certificate Manager is ready to request client certificates with Authority Information Access extension.

Step 6. Configure the Online Certificate Status Manager

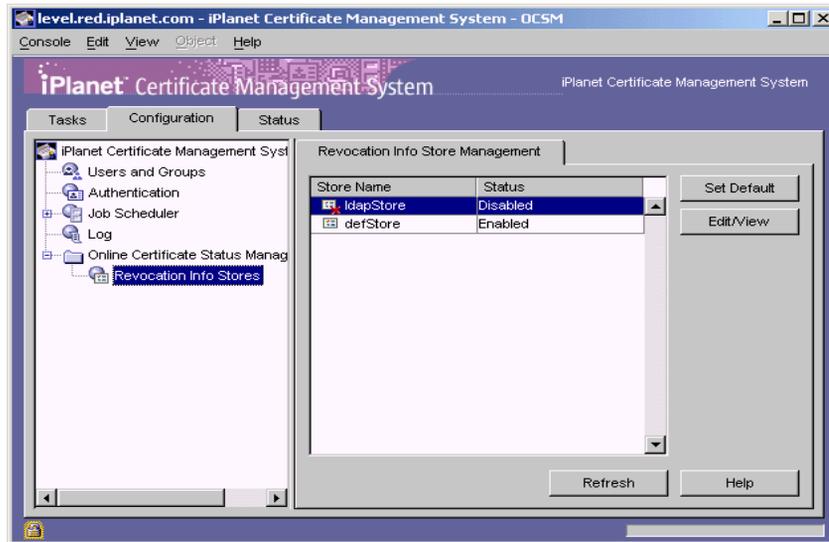
The Online Certificate Status Manager stores each Certificate Manager's CRL in its internal database and uses it as the default CRL store for verifying the revocation status of certificates. You can also configure the Online Certificate Status Manager to use the CRL published to an LDAP directory, instead of the CRL in its internal database. For example, if you've configured Certificate Managers to publish CRLs to LDAP directories (see Chapter 19, "Setting Up LDAP Publishing"), you can configure the Online Certificate Status Manager to use the CRLs published to these directories.

To configure the Online Certificate Status Manager to use the CRLs in its internal database or an LDAP directory for verifying revocation status of certificate:

1. Log in to the CMS window for the Online Certificate Status Manager (see "Logging In to the CMS Window" on page 351).
2. Select the Configuration tab.

3. In the navigation tree, select Online Certificate Status Manager, and then select Revocation Info Stores.

The right pane shows the two repositories the Online Certificate Status Manager can use; by default, it uses the CRL in its internal database.

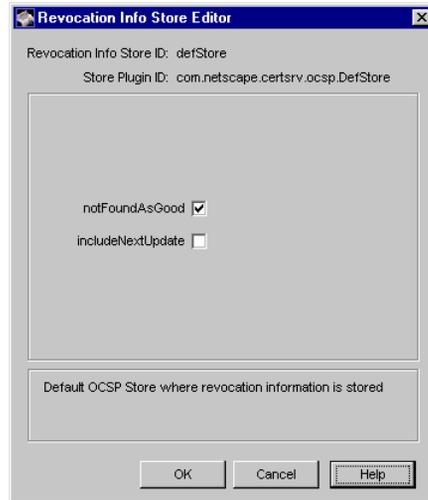


4. Select the appropriate option:
 - o If you want to configure the Online Certificate Status Manager to use the CRLs in its internal database, select `defStore` and click `Edit/View`.
 - o If you want to configure the Online Certificate Status Manager to use the CRLs in one or more directories, first click `Set Default` to enable the `ldapStore` option, select `ldapStore`, and click `Edit/View`. (Clicking the `Set Default` button toggles the selection between the two repositories.)

The Revocation Info Store Editor for the selected store appears.

5. Fill in the appropriate values.

- o If you selected `defStore`, fill in values as below:



notFoundAsGood. A certificate's status can typically be indicated by three possible OCSP responses, namely GOOD, REVOKED, and UNKNOWN. Select this option if you want the Online Certificate Status Manager to return an OCSP response of GOOD if the certificate in question cannot be found in the certificate repository. If you deselect the option, the response will be UNKNOWN, which when encountered by Netscape Personal Security Manager (an OCSP-compliant client) results in an error message.

includeNextUpdate. The Online Certificate Status Manager can include the time stamp of next CRL update—a future update time for the CRL or the revocation information—in the OCSP response that it sends to OCSP-compliant clients. (According to the OCSP protocol, it is optional to include the time stamp of next CRL update in an OCSP response.) Select this option if you want the OCSP response to contain information about the next CRL update. Leave the option deselected if you don't want the OCSP response to contain this information.

- o If you selected `ldapStore`, fill in values as below:

The screenshot shows a dialog box titled "Revocation Info Store Editor". At the top, it displays "Revocation Info Store ID: ldapStore" and "Store Plugin ID: com.netscape.certsrv.ocsp.LDAPStore". Below this are several input fields: "numConns" with the value "1", "host0" with "corpDir.siroe.com", "port0" with "389", "baseDN0" with "o=siroe.com", "refreshInSec0" with "86400", "caCertAttr" with "cACertificate;binary", and "crlAttr" with "certificateRevocationList;binary". There are two checkboxes: "notFoundAsGood" which is checked, and "includeNextUpdate" which is unchecked. At the bottom of the dialog are three buttons: "OK", "Cancel", and "Help".

numConns. Type the total number of LDAP directories the Online Certificate Status Manager should check. By default, this is set to 0. If you change the value to a positive integer, for example 1, 2, or 3, you will see that many sets of `host`, `port`, `baseDN`, and `refreshInSec` fields. (Change the value, click OK, and reopen the window to see the updated fields.)

host<n>. Type the fully-qualified hostname of the LDAP directory. The name must be in the `<machine_name>.<your_domain>.<domain>` form. For example, `corpDir1.siroe.com`.

port<n>. Type the nonSSL port of the LDAP directory. For example, 389.

baseDN<n>. Type the DN to start searching for the CRL. For example, `O=siroe.com`.

refreshInSec<n>. Type how often the connection be refreshed. The default is 86400 seconds (that is, refresh every day).

caCertAttr. Leave the default value, `cACertificate;binary`, as it is. (It's the attribute to which the Certificate Manager publishes its CA signing certificate.)

crlAttr. Leave the default value, `certificateRevocationList;binary`, as it is. (It's the attribute to which the Certificate Manager publishes CRLs.)

notFoundAsGood. A certificate's status can typically be indicated by three possible OCSP responses, namely GOOD, REVOKED, and UNKNOWN. Select this option if you want the Online Certificate Status Manager to return an OCSP response of GOOD if the certificate in question cannot be found in the certificate repository. If you deselect the option, the response will be UNKNOWN, which when encountered by Netscape Personal Security Manager (an OCSP-compliant client) results in an error message.

includeNextUpdate. The Online Certificate Status Manager can include the time stamp of next CRL update—a future update time for the CRL or the revocation information—in the OCSP response that it sends to OCSP-compliant clients. (According to the OCSP protocol, it is optional to include the time stamp of next CRL update in an OCSP response.) Select this option if you want the OCSP response to contain information about the next CRL update. Leave the option deselected if you don't want the OCSP response to contain this information.

6. Click OK.

You're returned to the Revocation Store Info Management tab

7. Click Refresh.

Step 7. Restart the Certificate Manager

For all your changes to take effect, you must restart the Certificate Manager. You can use the CMS window to restart the Certificate Manager. To restart Certificate Manager from the CMS window:

1. Select the Tasks tab.

2. Click Restart the Server.

When you restart the Certificate Manager, you are prompted to supply the single sign-on password for the server.

3. Type the single sign-on password you specified during installation and click OK. The Certificate Manager won't restart until you provide this password. For more information, see "Required Start-up Information" on page 322.

Step 8. Restart the Online Certificate Status Manager

For all your changes to take effect, you must restart the Online Certificate Status Manager. You can use the CMS window to restart the Online Certificate Status Manager:

1. Select the Tasks tab.
2. Click Restart the Server.

When you restart the Online Certificate Status Manager, you are prompted to supply the single sign-on password for the server.

3. Type the single sign-on password you specified during installation and click OK. The Online Certificate Status Manager won't restart until you provide this password. For more information, see "Required Start-up Information" on page 322.

Step 9. Verify Certificate Manager and Online Certificate Status Manager Connection

When you restart the Certificate Manager, it tries to connect to the Online Certificate Status Manager's agent port (you specified this in "Step C. Create a Publisher for the CRL" on page 740). To verify that the Certificate Manager did indeed communicate with the Online Certificate Status Manager:

1. Go to the web browser window and enter the URL for the Online Certificate Status Manager's Agent interface. The URL is in this format:
`https://<hostname>:<port>`.

The Online Certificate Status Manager Agent Services interface appears.

2. In the left frame, click List Certificate Authorities.

The resulting form should show information about the Certificate Manager (CA) you configured to publish CRLs to the Online Certificate Status Manager. Note the timestamp:

- The "This Update" and "Next Update" fields should now be updated with the appropriate timestamps, indicating that the Certificate Manager did communicate with the Online Certificate Status Manager.

- The “Requests Served Since Startup” field should show a value of zero (0), indicating that no OCSP-compliant client has queried the Online Certificate Status Manager yet for revocation status of a certificate.

Step 10. Test Your OCSP Responder Setup

To test whether the Certificate Manager is publishing to the Online Certificate Status Manager properly and to test that the online validation of certificates is taking place, follow these steps:

- Step A. Turn On Revocation Checking
- Step B. Request a Certificate
- Step C. Approve the Request
- Step D. Download the Certificate to the Browser
- Step E. Make Sure the CA is Trusted by the Browser
- Step F. Verify the Certificate in the Browser
- Step G. Check the Status of Online Certificate Status Manager
- Step H. Revoke the Certificate
- Step I. Verify the Certificate in the Browser
- Step J. Check the Online Certificate Status Manager Status Again

Step A. Turn On Revocation Checking

To ensure that Personal Security Manager (the OCSP-compliant client) is configured to verify the revocation status of certificates using the OCSP protocol:

1. Open a web browser window.
2. Open the Personal Security Manager interface.

In Communicator version 4.7, you can open this window by clicking the Security button in the navigation bar. Alternatively, you can also open this window by selecting Communicator from the main menu, selecting Tools, and then selecting Security Info.

3. Select the Advanced tab, and then in left pane, select Options.
4. Click the OCSP Settings button.

The OCSP Setting window appears.

5. Select the “Use OCSP to verify only certificates that specify an OCSP service URL” option, and click OK.
6. Click on the Close button.

Step B. Request a Certificate

The steps outlined below explain how to request a client certificate from the Certificate Manager using the *manual* enrollment method. If you’ve configured the Certificate Manager for automated certificate issuance, for example for directory-based enrollment, you may use the appropriate form and request a certificate.

To request a client or personal certificate from the Certificate Manager:

1. Go to the end-entity interface of the Certificate Manager you configured (or to the Registration Manager that’s connected to this Certificate Manager). The URL is in this form:

```
https://<hostname>:<end_entity_HTTPS_port> or  
http://<hostname>:<end_entity_HTTP_port>
```

2. In the left frame, under Browser, click Manual.

This opens the manual enrollment form.

3. Fill in all the values and submit the request.

The client prompts you to enter the password for your key database.

4. When you enter the correct password, the client generates the key pairs.

Do not interrupt the key-generation process.

Step C. Approve the Request

Skip this step if you requested the certificate using any of the automated enrollment methods. Complete this step if you used the manual enrollment form for requesting the certificate; the request you submitted is waiting in the agent queue for approval by an agent.

To approve the request:

1. Go to the Certificate Manager’s Agent Services interface.

The URL is in this format: `https://<hostname>:<agent_port>`

2. In the left frame, click List Requests.

3. In the form that appears, select the “Show pending requests” option and click Find.
4. In the list of pending requests, identify the request you submitted and click Details.
5. Check the request to make sure that it has all the required attributes of a client certificate, including the Authority Information Access extension.
6. Scroll to the bottom of the request form, and approve the request.

You should see a confirmation page indicating that the certificate has been issued. Don't close the page until after you complete the next step.

Step D. Download the Certificate to the Browser

To download the certificate into the certificate database of Personal Security Manager:

1. In the confirmation page, scroll down to the section that says “Installing this certificate in a client.”
2. Check the certificate details for the required extensions.
3. Follow the on-screen instructions and download the certificate to your browser's certificate database.

(An alternative way to download the certificate is to go to the Retrieval tab of the end-entity services interface, search for the certificate, and download the certificate.)

Step E. Make Sure the CA is Trusted by the Browser

When you downloaded the client certificate to the browser, the Certificate Manager's certificate chain also was downloaded to the browser's certificate database. Make sure that the Certificate Manager's CA signing certificate is trusted in the browser's certificate database; this is required for proper chaining during certificate validation.

1. In the browser, open the Personal Security Manager interface.

In Communicator version 4.7, you can open this window by selecting Communicator from the main menu, selecting Tools, and then selecting Security Info.

2. Select the Certificates tab and, in the left pane, click Authorities.

The list of CA certificates currently stored in the browser's certificate database appears.

3. Locate the Certificate Manager's CA signing certificate, select it, and click Edit.
The Edit Security Certificate Settings window appears.
4. Make sure all the three options are selected and click OK.

Step F. Verify the Certificate in the Browser

To verify that the certificate has been downloaded into the certificate database of Personal Security Manager:

1. Click the Certificates tab and, in the left pane, click Mine.
You should see the names of all the client certificates, including the one you just downloaded, stored in the browser's certificate database.
2. Select the name of the certificate you just downloaded and click View.
In the View Security Certificate dialog box that appears, look for a message that says that the certificate is verified; generally, it's at the top.

Step G. Check the Status of Online Certificate Status Manager

To go to the Online Certificate Status Manager's status page and verify the number of requests it has processed so far:

1. Go to the web browser window and enter the URL for the Online Certificate Status Manager's Agent interface. The URL is in this format:
`https://<hostname>:<port>`.
The Online Certificate Status Manager Agent Services interface appears.
2. In the left frame, click List Certificate Authorities.
The resulting form should show information about the Certificate Manager (CA) you configured to publish CRLs to the Online Certificate Status Manager. The page also summarizes the Online Certificate Status Manager's activity since it was last started.
Note the value assigned to the "Requests Served Since Startup" field. It should show a value of one (1), a proof that the OCSP-compliant client, Personal Security Manager, queried the Online Certificate Status Manager for revocation status of a certificate.

Step H. Revoke the Certificate

To revoke the certificate you issued so that the Certificate Manager publishes the CRL to the Online Certificate Status Manager:

1. Go to the end-entity interface for the Certificate Manager you configured (or to the Registration Manager that's connected to this Certificate Manager). Be sure to go to the HTTPS interface. The URL is in this form:

```
https://<hostname>:<end_entity_HTTPS_port>
```

2. Select the Revocation tab.
3. In the left frame, click User Certificate.
The User Certificate Revocation form appears.
4. In the Revocation Reason section, select Unspecified and click Submit.
The client shows the "Select a Certificate" dialog box and prompts you to choose the certificate you want to revoke.
5. Select the certificate you downloaded and click OK.
The Certificate Manager revokes the certificate, constructs the CRL, and publishes the CRL to the Online Certificate Status Manager.

Step I. Verify the Certificate in the Browser

To verify that the certificate has been revoked:

1. Open the Personal Security Manager interface.
2. Select the Certificates tab and then click Mine.
3. Select the certificate you revoked and click View.

In the View Security Certificate dialog box that appears, look for a message that says that the certificate could not be verified.

Step J. Check the Online Certificate Status Manager Status Again

You check the Online Certificate Status Manager status again to verify that these things happened:

- The Certificate Manager published the CRL (the revoked certificate) to the Online Certificate Status Manager.
- The browser sent an OCSP response to the Online Certificate Status Manager (this response was initiated when you clicked the View button).

- The Online Certificate Status Manager sent an OCSP response to the browser.
- The browser used that response to validate the certificate and informed you of its status (that the certificate could not be verified).

To check the Online Certificate Status Manager status for verification:

1. Go to the Online Certificate Status Manager's status page.
2. Reload the page (hold down the Shift key and click on the browser's Reload icon.)
3. Compare the information to the one you noted in Step G.

Note the updated statistics. It should indicate that Personal Security Manager queried the Online Certificate Status Manager about the status of the certificate and in response, the Online Certificate Status Manager sent a status.

Setting Up Key Archival and Recovery

When data is stored in encrypted form, you must have the private key that corresponds to the public key that was used to encrypt the data in order to decrypt and read it. If the private key is lost, the data cannot be retrieved. A private key can be lost because of a hardware failure, for example, or because the key's owner forgets the password or loses the hardware token in which the key is stored. Similarly, encrypted data cannot be retrieved if the owner of the key is unavailable to supply it—for example, has left the organization that owns the data.

This chapter explains how to use the Data Recovery Manager to archive users' encryption private keys and how to use the archived keys later, in place of missing encryption keys, to recover encrypted data.

The chapter has the following sections:

- PKI Setup for Key Archival and Recovery (page 759)
- Key Archival Process (page 761)
- Key Recovery Process (page 765)
- Configuring Key Archival and Recovery Process (page 775)

PKI Setup for Key Archival and Recovery

To be able to archive users' encryption private keys and recover them later, you need a PKI setup that includes the following elements:

- Clients that can generate dual keys and that support the key archival option (using the CRMF/CMMF protocol)
- An installed and configured Data Recovery Manager

- HTML forms with which your users can request dual certificates (based on dual keys) and key recovery agents can request key recovery

The sections that follow explain these elements in detail. For step-by-step instructions on setting up your PKI environment for key archival and recovery, see “Configuring Key Archival and Recovery Process” on page 775.

Clients That Can Generate Dual Key Pairs

Only keys that are used exclusively for encrypting data should be archived; signing keys in particular should never be archived. Having two copies of a signing key would defeat the certainty with which the key identifies its owner; a second copy could be used to impersonate the digital identity of the original key owner.

Clients that generate single key pairs use the same private key for both signing and encrypting data, so you cannot archive and recover a private key deriving from a single key pair. By contrast, clients that can generate dual key pairs use one private key for encrypting data and the other for signing data. Because the encryption private key is separate, you can archive it.

In addition to generating dual key pairs, your users' clients must also support the encryption key archival option in certificate requests. This option triggers the key archival process at the time encryption private keys are generated as a part of certificate issuance.

Netscape 6 and Netscape Communicator versions 4.7x (when used in conjunction with Netscape Personal Security Manager) support generation of dual key-pairs. For a brief introduction to Personal Security Manager, see page 39.

Data Recovery Manager

With the Data Recovery Manager, you can archive data encryption keys when they are created during dual key-pair generation. You can then recover the keys if they are lost or the key owner is unavailable.

The Data Recovery Manager can archive and recover keys only from clients that support dual key-pair generation and the key archival option in certificate requests.

Certificate Management System does not provide any policy plug-in modules for the Data Recovery Manager. However, you can write custom policy plug-in modules (that is, write Java classes that implement these rules), register them in the Data Recovery Manager's policy framework, and create policy rules using these plug-in implementations. For details about writing custom plug-ins, see "CMS SDK" on page 65.

Forms for Users and Key Recovery Agents

End users' encryption private keys are archived by the Data Recovery Manager when they are generated. So, for key archival to occur, the enrollment form that users fill out to request dual certificates must have the JavaScript code for activating the key archival process embedded in it, along with a valid copy of the Data Recovery Manager's transport certificate. Then, when a Certificate Manager or Registration Manager that is processing the user's certificate issuance request detects the key archival option, it automatically requests the service of the Data Recovery Manager. For information on customizing this form, see "Step C. Customize the Certificate Enrollment Form" on page 777.

Initiating the key recovery process also requires its own HTML form. By default, the Data Recovery Manager Agent Services interface provides a form for initiating the process and retrieving keys. For information on customizing this form, see "Step D. Customize the Key Recovery Form" on page 784.

Key Archival Process

If your certificate infrastructure has been set up for key archival, the Data Recovery Manager automatically archives users' encryption private keys. For general information on the type of PKI setup needed for archiving keys, see "PKI Setup for Key Archival and Recovery" on page 759. For specific instructions on setting up a key archival and recovery infrastructure, see "Configuring Key Archival and Recovery Process" on page 775.

Why You Should Archive Keys

If a user loses a private data-encryption key or is unavailable to use his or her private key, the key must be recovered before any data that was encrypted with the corresponding public key can be read. You can recover the private key if an archival copy of it was created when the key was generated.

Here are a few situations in which you might need to recover a user's encryption private key:

- An employee loses the encryption private key (for example, after a disk crash or by forgetting the password to the key file) and cannot read encrypted mail messages.
- An employee is on an extended leave, and you need access to an encrypted document in his or her files.
- An employee leaves the company, and company officials need to perform an audit that requires gaining access to the employee's encrypted mail.

Where the Keys are Stored

If configured properly, the Data Recovery Manager, stores your users' encryption private keys automatically whenever the associated or connected Registration Manager or Certificate Manager issues certificates to your users. The Data Recovery Manager stores encryption private keys in a secure key repository in its internal database; each key is stored as a key record.

The archived copy of the key remains encrypted (or wrapped) with the Data Recovery Manager's storage key; see "Storage Key Pair" on page 461. It can be decrypted (or unwrapped) only by using the corresponding private key, to which no individual has direct access. A combination of one or more key recovery agents' passwords enables the Data Recovery Manager to retrieve its private storage key and use it to decrypt and recover an archived key. For details on how this process works, see "Key Recovery Agents and Their Passwords" on page 765.

The Data Recovery Manager indexes stored keys by key number (or ID), owner name, and a hash of the public key, allowing for highly efficient searching by name or by public key. The key recovery agents have the privilege to insert, delete, and search for key records. The search feature works like this:

- When the key recovery agents search by the key ID, only the key that corresponds to that ID is returned.
- When the agents search by user name, all stored keys belonging to that owner are returned.
- When the agents search by the public key in a certificate, only the corresponding private key is returned.

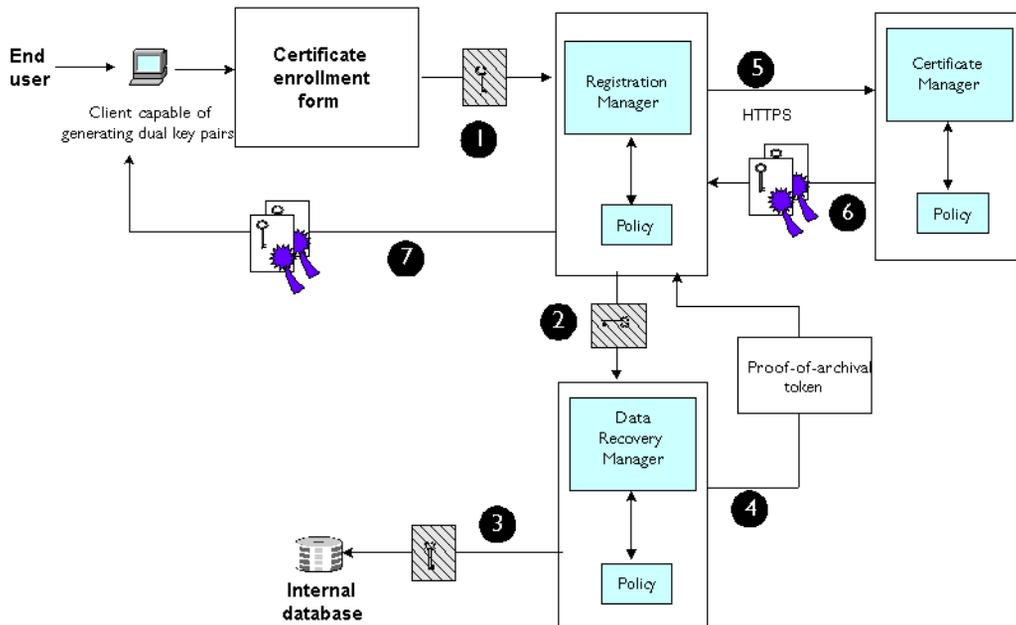
How Key Archival Works

When a Certificate Manager or Registration Manager receives a certificate request that contains the key archival option, it automatically requests the service of the Data Recovery Manager to archive the user's encryption private key. The Data Recovery Manager receives an encrypted copy of the user's private key and stores the key in its key repository. To archive the key, the Data Recovery Manager uses two special key pairs:

- A transport key pair and corresponding certificate
- A storage key pair

Figure 22-1 illustrates how the key archival process occurs when a user requests a certificate. The deployment scenario shown in this figure has a Registration Manager acting as the trusted enrollment authority to a Certificate Manager and Data Recovery Manager.

Figure 22-1 How the key archival process works



These are the steps shown in Figure 22-1:

1. A user uses a client capable of generating dual key pairs to access the certificate enrollment form served by the Registration Manager, fills in all the information, and submits the request.

The Registration Manager detects the key archival option in the user's request and asks the client for the user's encryption private key.

The client encrypts the user's encryption private key with the public key from the Data Recovery Manager's transport certificate; a copy of the transport certificate is embedded in the enrollment form.

2. Upon receiving the encrypted key from the client, the Registration Manager sends it to the Data Recovery Manager for storage, along with some other information (including the user's public key). Then, the Registration Manager waits for verification from the Data Recovery Manager that the private key has been received and stored and that it corresponds to the user's public encryption key.
3. Upon receiving the encrypted key from the Registration Manager, the Data Recovery Manager decrypts it with the private key that corresponds to the public key in its transport certificate. After confirming that the private encryption key corresponds to the user's public encryption key, the Data Recovery Manager encrypts it again with its storage key before storing it in its internal database. (The storage key either resides in a software or a hardware token and is never exposed to any other entity.)
4. Once the user's private encryption key has been successfully stored, the Data Recovery Manager uses the private key of its transport key pair to sign a token confirming that the key has been successfully stored; the Data Recovery Manager then sends the token to the Registration Manager.
5. After the Registration Manager receives and verifies the signed token, it sends the certificate request to the Certificate Manager for issuance.
6. The Certificate Manager formulates two certificates, one each for signing and encryption key pairs, and returns them to the Registration Manager.
7. The Registration Manager forwards the certificates to the client (the user).

Note that all three subsystems subject the request to configured policy rules at appropriate stages. If the request fails to meet any of the policy rules, the subsystem rejects the request.

Key Recovery Process

The Data Recovery Manager supports agent-initiated key recovery. In this method of key recovery, designated recovery agents use the Key Recovery form provided in the Data Recovery Manager Agent Services interface to process key recovery requests, list archived keys, and approve recovery. With the approval of a specified number of agents, an organization can recover keys when the key's owner is unavailable or when keys have been lost.

Key Recovery Agents and Their Passwords

Key recovery agents have the authority to retrieve end users' encryption private keys. The recovery agent's role can be performed by any person in your organization. As system administrator, you can designate one or more individuals to be key recovery agents. These individuals need to do the following:

- They must specify a secure password, which in combination with other recovery agents' passwords will be used for protecting the database in which the Data Recovery Manager stores users' keys. You facilitate this by allowing each recovery agent to enter a password in the Data Recovery Manager configuration.
- They must be available to retrieve your users' encryption private keys if the need arises. It isn't necessary for all key recovery agents to be available for the key recovery operation. You specify how many agents are required to authorize the recovery of a key; see "Key Recovery Agent Scheme" on page 771. However, the specified number of key recovery agents must all provide their passwords at the same time to authorize the recovery of a specific key.

The first time you create key recovery agents and specify their passwords is during the installation of the Data Recovery Manager. However, you can change the number of recovery agents and their passwords later by modifying it in the Data Recovery Manager configuration; see "Changing Key Recovery Agents' Passwords" on page 773.

Secret Sharing of Storage Key Password

The Data Recovery Manager uses the private key of its *storage key pair* to encrypt the repository where it store users' encryption private keys. This requires that the storage key be well protected. For the protection of the storage key pair, the Data Recovery Manager supports a password-splitting mechanism called *m of n secret*

splitting or sharing, whereby it splits the PIN that protects the token in which the storage key pair resides among n number of key recovery agents and reconstructs the PIN only if m number of recovery agents provide their individual passwords; n must be an integer greater than 1 and m must be an integer less than or equal to n .

Here's how the m of n secret splitting mechanism gets built and works:

During the installation of a Data Recovery Manager, you generate the storage key pair and specify the hardware token in which the key pair is to be stored. At this time, you also specify a PIN (or password) to protect the token, the total number of key recovery agents (n), and how many of these agents (m) are required to perform a key recovery operation. You can change the m of n secret splitting later; for details, see "Key Recovery Agent Scheme" on page 771.

The Data Recovery Manager splits the PIN for the token into n parts or pieces. It then encrypts these parts with the passwords that are provided by the authorized key recovery agents.

During the key recovery procedure, the required number of key recovery agents (m) provide their identifiers and passwords. After verifying the passwords, the Data Recovery Manager reconstructs the PIN for the token based on the given information.

Interface for the Key Recovery Process

With the Key Recovery form provided in the Data Recovery Manager Agent Services interface, key recovery agents can collectively unlock the key repository of the Data Recovery Manager and retrieve end users' encryption private keys and associated certificates in a PKCS #12 package, which can then be imported into the client. For an overview of this process, see "How Agent-Initiated Key Recovery Works" on page 768.

Because key recovery agents use the Data Recovery Manager Agent Services interface, agent-initiated key recovery invariably involves the Data Recovery Manager agent and key recovery agents. The Data Recovery Manager agent's certificate is required to access the Key Recovery form, and key recovery agents' passwords are required to unlock the key repository. For information on Data Recovery Manager agents, see "Agents" on page 397.

Your organization's PKI policy may require that the key recovery process be restricted to authorized recovery agents only, preventing any Data Recovery Manager agent from being involved. If so, you should ask all key recovery agents to get client certificates and set them up as Data Recovery Manager agents. For instructions, see "Setting Up Agents" on page 416.

Local Versus Remote Key Recovery Authorization

Key recovery agents can authorize the recovery of a key locally or remotely. The overview of local and remote authorization provided in this section is intended to help you determine which to use for your organization. You may find it useful to take a look at the Data Recovery Manager agent-specific information in the *CMS Agent's Guide*.

Local Key Recovery Authorization

To initiate key recovery locally, the required number of recovery agents assemble in front of the host system that allows them to access the Data Recovery Manager Agent Services interface. Either a Data Recovery Manager agent or a key recovery agent with a Data Recovery Manager agent certificate accesses the Key Recovery form hosted by the Data Recovery Manager and initiates the key recovery process. All key recovery agents enter their IDs and passwords on the same Recovery Authorization form presented by the Data Recovery Manager. If the information presented is correct, the Data Recovery Manager retrieves the requested key and returns it along with the corresponding certificate in the form of a PKCS #12 package.

By default, key recovery authorization is local.

Remote Key Recovery Authorization

To authorize key recovery remotely, the required number of recovery agents access the Data Recovery Manager Agent Services interface at their own locations and use the Authorize Recovery button to enter each authorization separately.

Before key recovery agents can authorize key recovery remotely, they must be set up to function as Data Recovery Manager agents. This role gives them the privilege to access the Data Recovery Manager's Agent Services interface directly.

In remote key recovery authorization, one of the key recovery agents informs all required recovery agents about an impending remote key recovery process. All recovery agents access the Key Recovery page hosted by the Data Recovery Manager. One of the agents initiates the key recovery process. The Data Recovery Manager returns a notification to each agent. The notification includes a recovery authorization reference number identifying the particular key recovery request that the agent is required to authorize. Each agent uses the reference number and authorizes key recovery separately.

The Data Recovery Manager informs the agent who initiated the key recovery process of the status of the authorizations. When all of the authorizations are entered, the Data Recovery Manager checks the information. If the information presented is correct, it retrieves the requested key and returns it along with the corresponding certificate in the form of a PKCS #12 package to the agent who initiated the key recovery process.

Key recovery agents can switch to remote authorization by deselecting the local authorization option in the Key Recovery form.

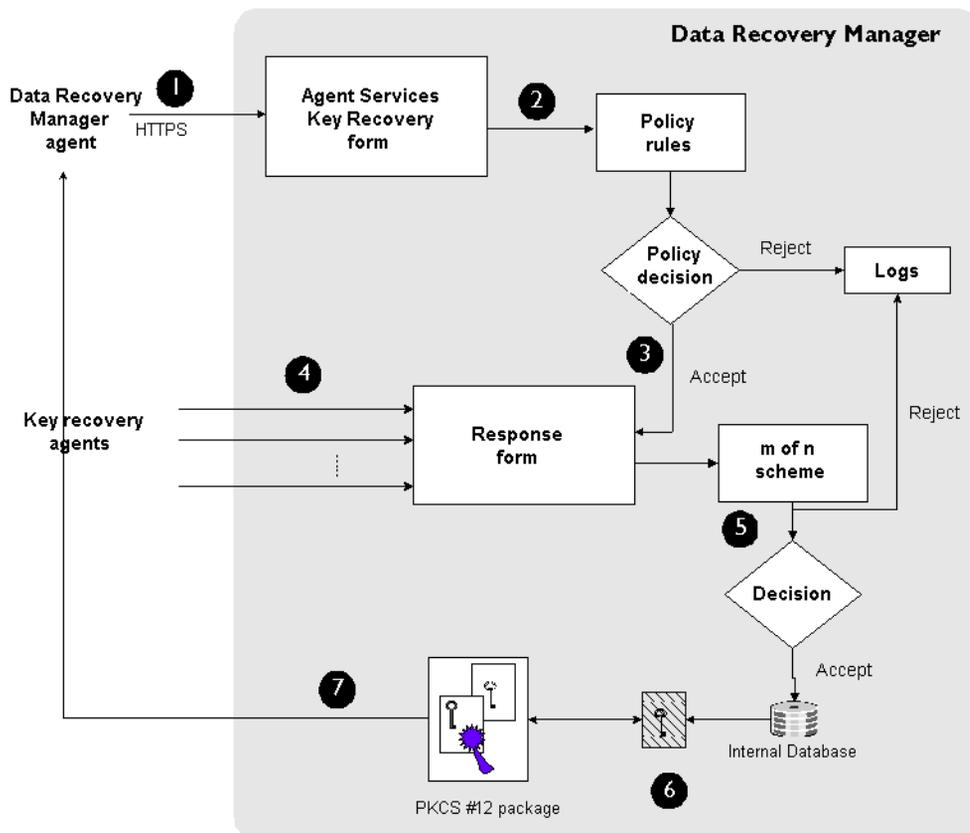
How Agent-Initiated Key Recovery Works

In an agent-initiated key recovery, the key is recovered by the collective efforts of a Data Recovery Manager agent and authorized key recovery agents. You may need to resort to this type of key recovery if the owner of a key cannot be reached and the authorities in the organization need to access that user's encrypted data (for example, S/MIME mail messages).

Upon retrieving the private encryption key (in the form of a PKCS #12 package), the agents may forward the key to the original user, the manager of the original owner, or some other authorities.

Figure 22-2 illustrates how agent-initiated key recovery works.

Figure 22-2 The agent-initiated key recovery process



These are the steps shown in Figure 22-2:

1. The Data Recovery Manager agent accesses the Key Recovery form using the appropriate client certificate, types the identification information pertaining to the person whose encryption private key needs to be recovered, and submits the request.

The request is submitted to the Data Recovery Manager over HTTPS.

2. The Data Recovery Manager subjects the key recovery request to its policy checks.

3. If the request passes all the policy rules, the Data Recovery Manager sends a confirmation HTML page to the web browser the agent used. If the request fails any of the policy checks, the server logs an appropriate error message.

The confirmation page contains information and input sections:

- The information section includes the user's information.
- The input section includes fields for entering the user's certificate corresponding to the key that needs to be recovered, the password for the PKCS #12 package, and key recovery agents' passwords.

The Data Recovery Manager uses the certificate to construct the PKCS #12 package (which includes the user's encryption private key and corresponding certificate), the PKCS #12 password to encrypt the PKCS #12 package, and key recovery agents' passwords to construct the PIN required to unlock its key repository.

4. The key recovery agents verify the information in the confirmation page and enter the certificate in MIME-64 format, the password for the PKCS #12 package, and their individual identifiers and passwords. The Data Recovery Manager agent submits the page to the Data Recovery Manager.
5. The Data Recovery Manager matches the key recovery agent information with its *m of n scheme* (see "Key Recovery Agent Scheme" on page 771). After verifying that the required number of recovery agents entered their passwords, the server uses the agents' passwords to construct the PIN required to access the private key repository.
6. The Data Recovery Manager then retrieves the user's private key from its key repository and decrypts it by using the private component of the storage key pair.
7. The Data Recovery Manager packages the user's certificate and the corresponding private key as a PKCS #12 package and encrypts it with the PKCS #12 password provided by the recovery agent. It then delivers the package to the client the recovery agent used to initiate the key recovery process, and prompts the agent to store the encrypted package. The agent may choose to store the package in the local file system of the client machine (only if it has restricted access) or on a floppy diskette.

The recovery agent can then send the encrypted PKCS #12 package and the corresponding password to an individual by any secure, out-of-band means.

CAUTION The PKCS #12 package contains the private key. To minimize the risk of key compromise, the recovery agent must use any secure, out-of-band means to deliver the PKCS #12 package and password to the key recipient. As an administrator, you should recommend the recovery agent to use a good password for encrypting the PKCS #12 package, and also consider setting up an appropriate delivery mechanism.

Key Recovery Agent Scheme

The *key recovery agent scheme* consists of configuring the Data Recovery Manager to recognize a fixed number of key recovery agents (a minimum of one) and specifying how many of these agents are required to authorize a key recovery request before the archived key is restored. Each recovery agent provides the Data Recovery Manager with a password, which it uses to generate a unique PIN; the Data Recovery Manager uses the PIN to protect its storage key pair, which in turn protects users' keys.

The Data Recovery Manager tracks the key recovery agent password for each agent and allows you to facilitate changing agents' passwords; you do not have direct access to these passwords or the actual storage key password. Each password retrieves only a part of the private storage key.

You first specified the key recovery agent scheme when you installed the Data Recovery Manager.

Changing the Key Recovery Agent Scheme

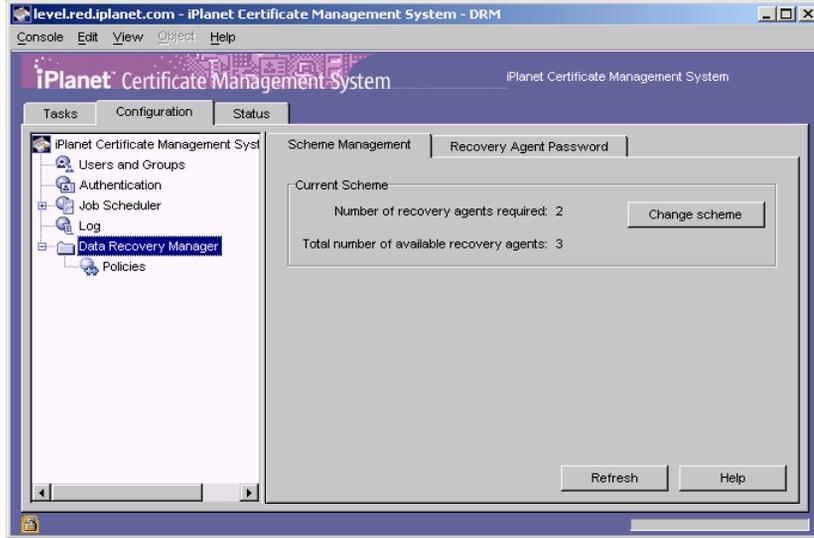
You can change the total number of key recovery agents for a Data Recovery Manager and the number of key recovery agents required to retrieve an end user's encryption private key from the Data Recovery Manager's key repository.

To change the key recovery agent scheme:

1. Access the CMS window (see "Logging In to the CMS Window" on page 351).
2. Click the Configuration tab.

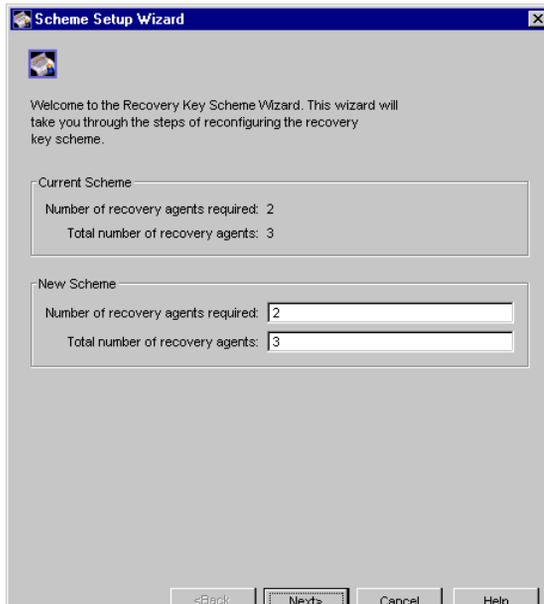
3. In the navigation tree, select the Data Recovery Manager, and in the right pane, click the Scheme Management tab.

The Scheme Management tab shows the current key recovery scheme.



4. Click Change scheme.

The Change Recovery Key Scheme window appears.



5. In the New Scheme section, make the appropriate changes:
 - Number of recovery agents required.** Type the number of agents required to authorize a key recovery process. The number cannot be zero and must be equal to or less than the total number of recovery agents.
 - Total number of recovery agents.** Specify the total number of key recovery agents. The number cannot be less than one and must be equal to or greater than the number of agents required to authorize the key recovery operation.
6. Click Next.
7. For each agent, enter a user name and password, then click Next.

The number of screens depends on the total number of agents you have specified.
8. When you have entered all agent information, click Finish.

You are returned to the Scheme Management tab.

Changing Key Recovery Agents' Passwords

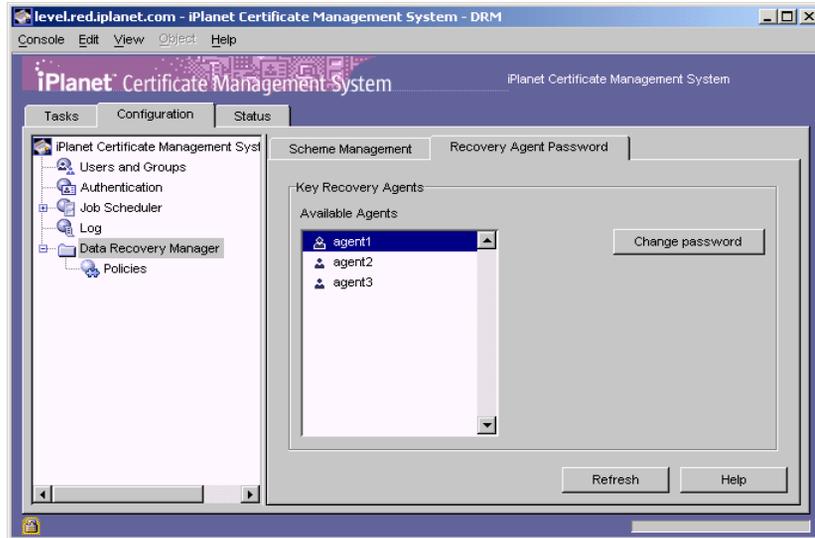
As administrator, you have the responsibility of safeguarding the security of each Data Recovery Manager installed in your PKI setup. One of the safety measures you can implement is to ask your key recovery agents to change their passwords periodically. This way, you will be sure that the required number of agents are available if a key needs to be recovered. If key recovery agents routinely change their passwords, they are less likely to forget them.

The CMS window allows you to view the list of currently designated key recovery agents and, if necessary, change an agent's password.

To change key recovery agents' passwords:

1. Access the CMS window (see "Logging In to the CMS Window" on page 351).
2. Click the Configuration tab.
3. In the navigation tree, select the Data Recovery Manager, and in the right pane, click the Recovery Agent Password tab.

The tab shows current key recovery agents in the Available Agents list.



4. Select the agent whose password needs to be changed, and click Change Password.

The Change Password dialog box appears.



5. Allow the agent to enter the appropriate information.

During installation, the Data Recovery Manager prompts you to enter key recovery agent passwords (by default, they are set to `agent<n>`, where `<n>` can be 1, 2, and so on, depending on the number of agents). The number of passwords you must enter depends on the key recovery agent scheme you chose; for details, see “Key Recovery Agent Scheme” on page 771. If you are changing a password for the first time after installation, in the “Old password”

field you must enter the recovery agent password you specified during installation. Then in the remaining fields, allow the key recovery agent to enter the new password information. If you have more than one key recovery agent, repeat this procedure for all the agents.

Old Password. Type the current password for the key repository.

New Password. Type the new password for the key repository.

Confirm Password. Retype the new password exactly as you typed it in the previous field.

6. Click OK.

You are returned to the Recovery Agent Password tab.

Configuring Key Archival and Recovery Process

By default, the Data Recovery Manager is not configured to archive or recover end users' encryption private keys. This section explains how to set up key archival and recovery processes.

- Step 1. Set Up the Key Archival Process
- Step 2. Set Up the Key Recovery Process
- Step 3. Test Your Key Archival and Recovery Setup

Step 1. Set Up the Key Archival Process

Before proceeding with this section, you should have read “Key Archival Process” on page 761. In particular, you should be familiar with how the key archival process works. If you are not, see “How Key Archival Works” on page 763.

To set up the key archival process, follow these steps:

- Step A. Deploy Clients That Can Generate Dual Key Pairs
- Step B. Connect the Enrollment Authority and the Data Recovery Manager
- Step C. Customize the Certificate Enrollment Form
- Step D. Configure Key Archival Policies

Step A. Deploy Clients That Can Generate Dual Key Pairs

You can use the Data Recovery Manager to archive and recover keys only from clients that support dual key-pair generation, the key archival option, and the CMC protocol. Clients that do not meet this criteria cannot be used with the Data Recovery Manager. To understand why you need to use clients that can generate dual key pairs, see “Clients That Can Generate Dual Key Pairs” on page 760. The same section also points you to an introduction to Netscape Personal Security Manager, which when plugged into Netscape Communicator version 4.7x enables it to support the CMC protocol and generate dual key pairs.

You may have already installed Personal Security Manager—for example, you might have installed it as an OCSP-compliant client when setting up a Certificate Manager to publish CRLs to an OCSP responder; see “Step 2. Install an OCSP-Compliant Client” on page 734.

Step B. Connect the Enrollment Authority and the Data Recovery Manager

Key archival occurs when dual key pairs are generated by the client. The client generates the key pairs when a user requests a certificate by filling out the appropriate certificate enrollment form served by an enrollment authority, which can be either a Certificate Manager or a Registration Manager. When the enrollment authority detects the key archival option in the request, it initiates the key archival process and requests the service of the Data Recovery Manager for archiving the key.

For the enrollment authority to be able to request the service of the Data Recovery Manager, the two subsystems must be configured to recognize, trust, and communicate with each other. When you installed the Data Recovery Manager, you were asked to connect it to a Certificate Manager or Registration Manager. You might have specified some of the configuration information required for the two subsystems to communicate with each other. Also, if the enrollment authority and the Data Recovery Manager are installed in the same CMS instance, certain configurations are done automatically.

However, to ensure that key archival takes place successfully, you must make sure that the Data Recovery Manager is connected to the appropriate enrollment authority. Also verify whether the enrollment authority has been set up as a privileged user, with an appropriate SSL client authentication certificate, in the internal database of the Data Recovery Manager. By default, the Certificate Manager uses its *SSL server certificate* for SSL client authentication, whereas the Registration Manager uses its *signing certificate* for this purpose; for more information, see “Keys and Certificates for the Main Subsystems” on page 450.

Otherwise, follow the instructions in “Setting Up Trusted Managers” on page 423 and set up the enrollment authority as a trusted front end to the Data Recovery Manager.

Step C. Customize the Certificate Enrollment Form

For the enrollment authority to automatically initiate the key archival process at the time key pairs are generated, a certificate request must include the following information:

- The key archival option—this must be included in the certificate enrollment form that your users use to request certificates.
- The Data Recovery Manager’s transport certificate—this must also be included in the certificate enrollment form. The Data Recovery Manager uses it to encrypt the user’s encryption private key with the public key in the transport certificate before sending the user’s key to its key repository. For information about the key repository, see “Where the Keys are Stored” on page 762.

Make sure that the transport certificate, in its base-64 encoded format, is embedded in the form. Otherwise, the Data Recovery Manager will fail to archive users’ keys.

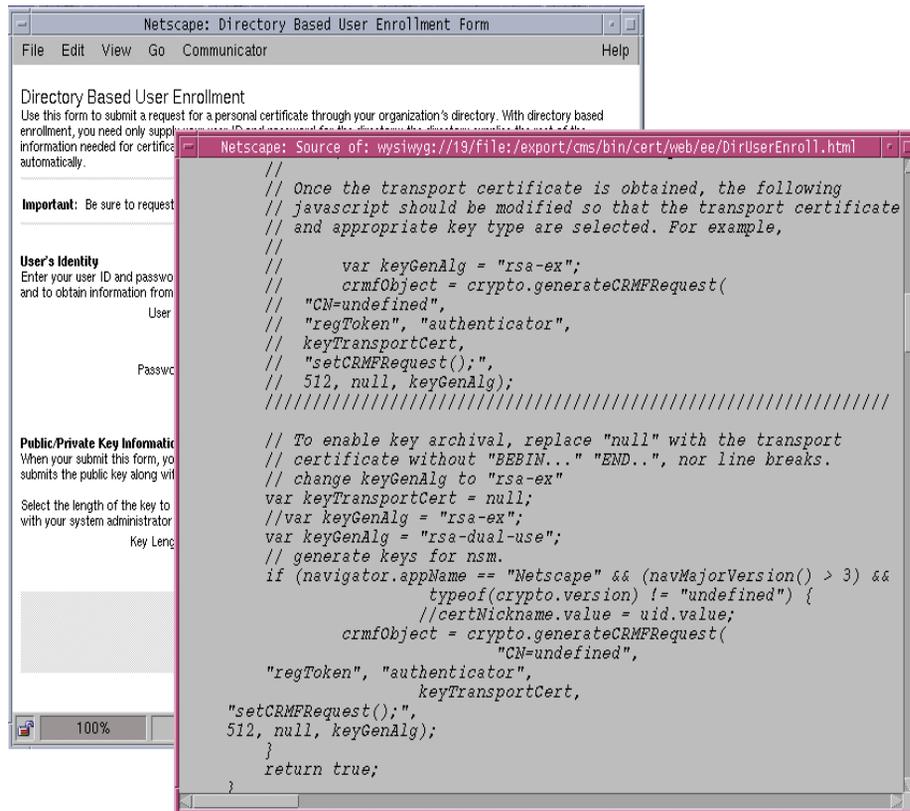
All the end user enrollment forms provided by Certificate Management System—for example, the directory-based enrollment form (`DirUserEnroll.html`), directory- and PIN-based enrollment form (`DirPinUserEnroll.html`), and manual enrollment form (`ManUserEnroll.html`)—contain the necessary JavaScript code for initiating the key archival process. If you are using any of these forms for end-user enrollment, make sure to update the `generateCRMRequest()` JavaScript method. If you plan to use custom enrollment forms for users, be sure to include the required JavaScript code in those forms.

Figure 22-3 shows the default directory-based enrollment form with the information related to the `generateCRMRequest()` JavaScript method highlighted. Note that the JavaScript method includes parameters for specifying various things. You are required to update the following information only:

- The Data Recovery Manager’s transport certificate.
- The algorithm, length, type, and usage for end users’ key pairs. When you update this information, the key archival option is automatically set. For information on specifying the key type, length, and algorithm, see `generateCRMRequest()` in *Javascript API for Client Certificate Management*. This document is located where you extracted Personal Security Manager files after downloading it from the web site.

The steps that follow explain how to do this.

Figure 22-3 Data Recovery Manager's transport certificate in the enrollment form



1. Copy the transport certificate in its base-64 encoded format.

The transport certificate is stored in the Data Recovery Manager's certificate database. If the transport certificate is signed by a Certificate Manager, then a copy of the certificate is also available with the Certificate Manager. Follow the instructions as appropriate.

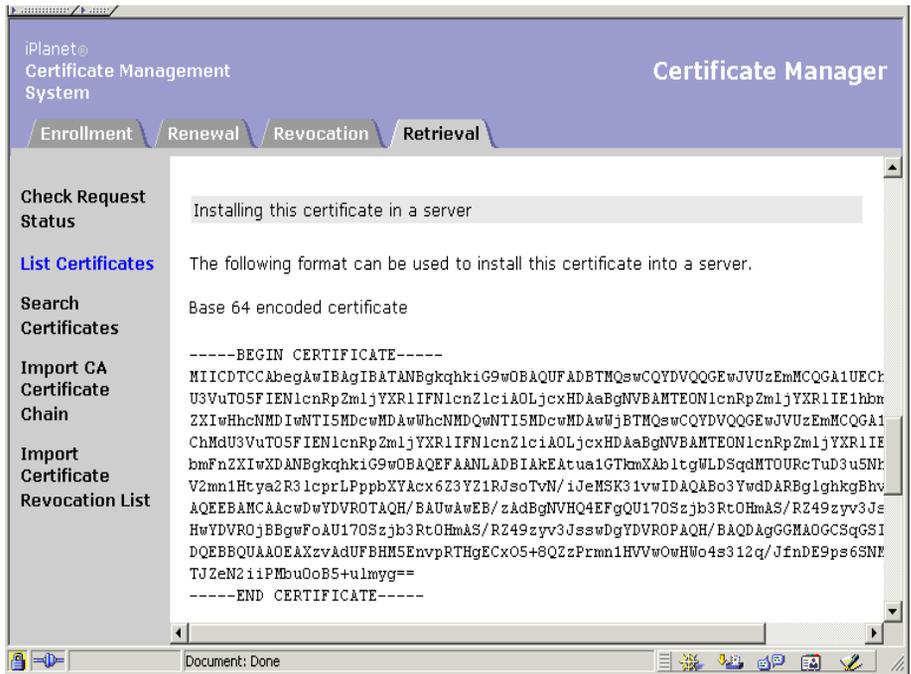
To copy the transport certificate information from a Certificate Manager's database:

- a. Open a web browser window.
- b. Go to the end-entity page hosted by the Certificate Manager.

- c. Click the Retrieval tab.
- d. List or search for the transport certificate.
- e. Click Details, and view the certificate information.

Make sure that the certificate you are looking at is the correct one; the certificate shows the DN that was specified for the transport certificate during the installation of Data Recovery Manager.

- f. Scroll down to the section that says “Installing this certificate in a server.”



- g.** Copy the base-64 encoded certificate, *excluding* the marker lines -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----, to a text file. An example is shown below:

```
MIICDjCCAXegAwIBAgICAFmWdQYJKoZIhvcNAQEEBQAwdzELMAkGA1UEBhMCV
VMxLDAqBgNVBAoTI05ldHNjYXB1IENvbW11bmljYXRpb25zIENvcnBvcnF0aW
9uMREwDwYDVQQLZWhlYXJkY29yZTEuMCUGA1UEAxMeSGFyZG9uU2VydG1
maWNhdGUGU2VydMvYIElJMB4XDk4MTE4OTIzNDIxOVoXDTk5MDUxODIzNDIx
OVowLjELMAkGA1UEBhMCVVMxETAPBgNVBAoTCG5ldHNjYXB1MQwwCgYDVQDE
wNLUmEwXDANBgkqhkiG9w0BAQEFAANLADBIAkEArrbDiYUI5SCdlCKKa0bEBN
1m83kX6bdhytRYNkdHB9
```

To copy the transport certificate information from a Data Recovery Manager's certificate database:

- a.** Open a terminal window in the system that hosts the Data Recovery Manager.
- b.** Use the command-line tool called `certutil` to retrieve the transport certificate from the Data Recovery Manager's certificate database. (For information on the `certutil` tool, see Chapter 11, "Certificate Database Tool" of *CMS Command-Line Tools Guide*.)

First, go to this directory: `<server_root>/cert-<instance_id>/config`

Next, run this command: `<server_root>/bin/cert/tools/certutil -L -d . -n kraTransportCert cert-<instance_id> -a`

The transport certificate appears. View the certificate information. Make sure that the certificate you are looking at is the correct one; the certificate shows the DN that was specified for the transport certificate during the installation of Data Recovery Manager.

- c.** Copy the base-64 encoded certificate, *excluding* the marker lines -----BEGIN CERTIFICATE----- and -----END CERTIFICATE-----, to a text file. The copied information should look like the example below:

```
MIICDjCCAXegAwIBAgICAFmWdQYJKoZIhvcNAQEEBQAwdzELMAkGA1UEBhMCV
VMxLDAqBgNVBAoTI05ldHNjYXB1IENvbW11bmljYXRpb25zIENvcnBvcnF0aW
9uMREwDwYDVQQLZWhlYXJkY29yZTEuMCUGA1UEAxMeSGFyZG9uU2VydG1
maWNhdGUGU2VydMvYIElJMB4XDk4MTE4OTIzNDIxOVoXDTk5MDUxODIzNDIx
OVowLjELMAkGA1UEBhMCVVMxETAPBgNVBAoTCG5ldHNjYXB1MQwwCgYDVQDE
wNLUmEwXDANBgkqhkiG9w0BAQEFAANLADBIAkEArrbDiYUI5SCdlCKKa0bEBN
1m83kX6bdhytRYNkdHB95B
```

- 2.** Update the JavaScript method in the enrollment form.

To do this:

- a. Go to the host system of the enrollment authority and locate the user-enrollment form. The default forms are at this location:
`<server_root>/cert-<instance_id>/web/ee`
- b. Open the enrollment form that you want to use in a text editor.
- c. In the form, locate the `generateCRMRequest()` JavaScript method (see Figure 22-3 on page 778).

- d. Add a variable for the transport certificate.

Below the commented text, add this line:

```
var kraTransportCert =
```

- e. Open the text file that has the Data Recovery Manager's transport certificate (the one you copied earlier) and copy the certificate.
- f. Paste the certificate as the value of the `kraTransportCert` variable.

Paste the certificate in front of the `=` sign, remove any line breaks, enclose the certificate within double-quotation marks (`"`), and end the string with a semicolon (`;`). When deleting line breaks, be sure not to delete any of the characters in the encoded blob.

An example is shown below:

```
var kraTransportCert =
"MIICDjCCAXegAwIBAgICAfMwDQYJKoZIhvcNAQEEBQAwdzELMAkGA1UEBhMC
VVMxLDAqBgNVBAoTI05ldHNjYXB1IENvbW11bmljYXRpb25zIENvcnBvcnF0a
W9uMREwDwYDVQQLEWhIYXJkY29yZTEuMCUGA1UEAxMeSGFyZGZGNvcnUgQ2VydG
lmaWNhdGUGU2VydGVyIEI1JmB4XDTk4MTEwOTIzNDIxOVV5ODIzNDIzNDI
xOVV5ODIzNDIzNDIzNDIzNDIzNDIzNDIzNDIzNDIzNDIzNDIzNDIzNDIzNDI
EwNLUmEwXDANBgkqhkiG9w0BAQEFAANLADBIAGkqEArrbDiYUI5SCdlCKKa0bEB
n1m83kX6bdhytRYNkdHB95Bp85SR";
```

- g. Pass the `kraTransportCert` variable to the JavaScript method.

Replace `null` (the fourth line in the method) with `kraTransportCert`.

- h. Specify the key algorithm and key type (see “`generateCRMRequest()`” in *Javascript API for Client Certificate Management*).

Below is an example that shows how the updated `generateCRMRequest()` method would look:

```
// generate keys for PSM.
if (navigator.appName == "Netscape" && (navMajorVersion() > 3) &&
    typeof(crypto.version) != "undefined") {
    certNickname.value = subject.value;
```

```

        crmfObject = crypto.generateCRMFRequest(subject.value,
            "regToken",
            "authenticator",
            kraTransportCert,
            "setCRMFRequest(";
            512, null, "rsa-ex",
            1024, null, "rsa-sign");
    }

```

The method triggers the client to generate two RSA key pairs—one key of length 512 for encrypting data and another key of length 1024 for signing data.

- i. Save your changes.

Step D. Configure Key Archival Policies

This step is optional.

Unlike Certificate Manager and Registration Manager, no policy plug-in modules are provided for the Data Recovery Manager. If you have implemented any custom policy modules for the Data Recovery Manager's key archival process, you should make sure that they are configured properly. For details on configuring policies for a subsystem, see “Configuring Policy Rules for a Subsystem” on page 613.

Step 2. Set Up the Key Recovery Process

Before proceeding with this section, you should have read “Key Recovery Process” on page 765. In particular, you should be familiar with how the key archival process works. If you are not, see “How Agent-Initiated Key Recovery Works” on page 768.

The Data Recovery Manager supports agent-initiated key recovery process, in which end users' encryption private keys are recovered by designated key recovery agents. This section explains how to set up the key recovery process.

To set up agent-initiated key recovery process, follow these steps:

- Step A. Verify the m of n Scheme
- Step B. Facilitate the Key Recovery Agents to Change the Passwords
- Step C. Determine the Authorization Mode for Key Recovery
- Step D. Customize the Key Recovery Form
- Step E. Configure Key Recovery Policies

Step A. Verify the m of n Scheme

During the installation of the Data Recovery Manager, you were asked to specify the total number of key recovery agents (a minimum of one) and the number of agents (of this total) required to authorize a key recovery operation. This combination is called m of n scheme. For more information about this, see “Key Recovery Agent Scheme” on page 771.

Verify that the current m of n scheme is appropriate for your PKI setup. If it isn't, change the scheme following the instructions in “Changing the Key Recovery Agent Scheme” on page 771.

Step B. Facilitate the Key Recovery Agents to Change the Passwords

During the installation of Data Recovery Manager, after you specified the F of n scheme, you were also prompted to provide unique passwords for each recovery agent. It is quite likely that you specified these passwords yourself instead of it being done by those individuals who have been designated with the key recovery agents' role in your organization. Therefore, you must get the designated recovery agents to change the passwords entered during installation.

- To understand the significance of key recovery agents' passwords, see “Key Recovery Agents and Their Passwords” on page 765.
- To get the recovery agents to change the passwords, follow the instructions in “Changing Key Recovery Agents' Passwords” on page 773.

Step C. Determine the Authorization Mode for Key Recovery

The Data Recovery Manager allows key recovery agents to authorize recovery of an end user's encryption private key locally or remotely. The default configuration is local authorization. It is important that you evaluate both the authorization modes, and choose the one that is appropriate for your organization. For more information about this, see “Local Versus Remote Key Recovery Authorization” on page 767.

If want the key recovery agents to authorize key recovery remotely, be sure to set them up as Data Recovery Manager agents following the instructions in “Setting Up Agents” on page 416.

Step D. Customize the Key Recovery Form

Key recovery agents need an appropriate interface to initiate the key recovery process. By default, the Data Recovery Manager's Agent Services interface includes an HTML form (`recoverKey.html`) that allows key recovery agents to initiate the key recovery process and retrieve users' encryption keys. For details about this form, check *CMS Customization Guide*.

If you want to customize this form to suit your organization, be careful not to delete any of the information that is vital to the functioning of the form; it is recommended that you restrict your changes to the content presented in the form.

Step E. Configure Key Recovery Policies

This step is optional.

Unlike Certificate Manager and Registration Manager, no policy plug-in modules are provided for the Data Recovery Manager. If you have implemented any custom policies for the Data Recovery Manager's key recovery process, you should make sure that they are configured properly. For details on configuring policies for a subsystem, see "Configuring Policy Rules for a Subsystem" on page 613.

Step 3. Test Your Key Archival and Recovery Setup

The steps outlined in this section explain how to verify key archival and recovery process using Netscape Communicator 4.7 with Personal Security Manager, version 1.01.

Step A. Test Your Key Archival Setup

To test whether you can successfully archive a key, follow these instructions.

1. Enroll for dual certificates.

To do this:

- a. Open a web browser window.
- b. Go to the end-entity interface for the enrollment authority.

The default URL is as follows:

```
https://<hostname>:<end_entity_HTTPS_port> or  
http://<hostname>:<end_entity_HTTP_port>
```

- c. In the end-entity interface, open the enrollment form you customized in “Step C. Customize the Certificate Enrollment Form” on page 777.

(Perform a “View Source” to ensure that the browser loads the correct page with the JavaScript method you edited.)

- d. Fill in all the values and submit the request.

The client prompts you to enter the password for your key database.

- e. When you enter the correct password, the client generates the key pairs.

Do not interrupt the key-generation process.

2. Approve the request.

This step is required only if you used the manual enrollment form for requesting the certificate.

- a. Go to the enrollment authority’s Agent Services interface.

The default URL is as follows: `https://<hostname>:<agent_port>`

- b. Click the link that says List Requests.

- c. In the form that appears, select the “Show pending requests” option and click Find.

You should see your request in the list of pending requests.

- d. Make sure the request has the appropriate extensions set for S/MIME (E-mail bit of the Netscape Certificate Type extension selected) and the subject name contains the email information (the value of the E attribute).

- e. Locate and approve the request.

3. Check if the certificates have been issued.

To do this:

- a. Click the List Requests link again.

- b. In the form that appears, select the “Show completed requests” option and click Find.

You should see two new certificates with consecutive serial numbers.

- c. Download the certificates to the web browser.

- d. Go to the security information window of the browser (from the Communicator menu, choose Tools, and then choose Security Info).

- e. Click Certificates and then click Mine.
- f. Verify that the test certificates have been stored in the browser's certificate database.

You will see only one entry for both the certificates you downloaded. If you select the entry and click View, you should see two certificates—one certificate for encipherment and another one for digital signature.

4. Check whether the key has been archived. To do this:
 - a. Go to the Data Recovery Manager's Agent Services interface.
 - b. Click the link that says List Requests.
 - c. In the form that appears, check the "Show completed requests" option and click Find.
 - d. If the key has been archived successfully, you should see the information pertaining to that key. If you don't see the key archived, check the logs and correct the problem before proceeding to the next step.
 - e. If the key has been successfully archived, exit the client completely—that is, from the File menu, select Exit; this will close all client windows.

Step B. Verify the Key

To do this:

1. Open a browser window again.
2. Open the email client, Messenger, and send a signed and encrypted email to yourself.
3. When you receive the email, open it, and check the message to see if it is signed and encrypted.

There should be a security icon at the top-right corner of the message window and it should indicate that the message is signed and encrypted.

Step C. Delete the Certificate

To do this:

1. Open the security information window.
2. Click Certificates and then click Mine.
3. In the list of certificates, select the test certificate you downloaded previously and click Delete.

4. When prompted confirm the delete action.
5. Check your email again. This time, you should not be able to verify the email message because you have deleted the certificates from the client's certificate database.

Step D. Test Your Key Recovery Setup

To test whether you can successfully recover an archived key:

1. Go to the Data Recovery Manager's Agent Services interface.
2. Click the Recover Keys link.
3. In the form that appears, enter any of the following information for the encryption private key that has been archived:
 - The key owner's name
 - The serial number of the key
 - The public key that corresponds to the private key (in the form of base-64 encoded certificate)
 - The instance ID of the enrollment authority that initiated the key archival process

If you need more information about any of the fields in this form, click the Help button.

4. Click Show Key.

If the key has been archived successfully, you should see the information pertaining to that key.

5. Click Recover.
6. In the form that appears, enter the following information:
 - The PKCS #12 password; the Data Recovery Manager uses this password to encrypt the PKCS #12 package (see "How Agent-Initiated Key Recovery Works" on page 768).
 - The base-64 encoded certificate that corresponds to the private key you want to recover; use the enrollment authority's end-entity or agent interface to get this information. If you searched for the archived key by providing the base-64 encoded certificate in (step 4), then you don't have to provide this information.
 - The key recovery agents' passwords.

7. Click Recover.

If you entered the correct information, the Data Recovery Manager returns the private key packaged as a PKCS #12 blob (it contains the recovered key pair and the corresponding certificate) and prompts you to save it. Specify the path and filename for saving the encrypted file. Be sure not to change the default file extension (.p12).

Step D. Restore the Key in the Browser's Database

To do this:

1. Go to the security information window of your browser.
2. Import the *.p12 file (that you saved in the previous step) back into the browser.
3. Open the test email that you couldn't verify after deleting the certificate from the browser's certificate database; you should be able to verify it again.

Managing CMS Logs

Each instance of iPlanet Certificate Management Server (CMS) maintains its own system, error, and audit log files. These files record events related to various CMS activities. By configuring logs, you can customize the contents in the log files.

This chapter explains how to use the CMS window to configure the system, error, and audit logs maintained by Certificate Management System, and how to monitor its activities by viewing log contents.

The chapter has the following sections:

- Introduction to Logs (page 789)
- Configuring CMS Logs (page 797)
- Monitoring CMS Logs (page 803)
- Archiving of Rotated Log Files (page 813)
- Managing Log Modules (page 816)

Introduction to Logs

iPlanet Certificate Management Server (CMS) creates log files that record events related to its activities, such as administration, communications using any of the protocols the server supports, and various other processes employed by all the subsystems the server manages.

This section identifies various logs maintained by Certificate Management System and describes them in detail.

- Logs Maintained by the Server (page 790)
- Services That Are Logged

- Log Levels (Message Categories)
- Log File Locations
- Log File Naming Conventions
- Buffered Versus Unbuffered Logging
- Rotation of Log Files
- Deletion of Log Files

Logs Maintained by the Server

While Certificate Management System is running, it keeps a log of information and error messages on all the components it manages. These messages are broadly categorized into three separate groups and are maintained in three separate log files, as listed in Table 23-1.

During installation, Certificate Management System automatically creates the required log files in your local file system. The server creates common system, error, and audit files for all components that were installed together, and it logs messages to these files. For example, if you installed a Certificate Manager and a Data Recovery Manager together, you will find log messages for both the subsystems in the same log file.

Table 23-1 Types of logs maintained by Certificate Management System

Log type	Description
System	<p>This log records information about requests to the server (all HTTP and HTTPS requests) and the responses from the server. Information recorded in this log includes the IP address of the client machine that accessed the server, operations performed (for example, search, add, edit), and the result of the access (for example, the number of entries returned). This log is on by default.</p> <p>For more information, see “Configuring CMS Logs” on page 797 and “Monitoring System Logs” on page 804.</p>
Error	<p>This log contains the error messages the server has encountered (HTTP errors and errors with the certificate service). This log is on by default.</p> <p>For more information, see “Configuring CMS Logs” on page 797 and “Monitoring Error Logs” on page 806.</p>

Table 23-1 Types of logs maintained by Certificate Management System *(Continued)*

Log type	Description
Audit	<p>This log records messages specific to the certificate service—messages such as certificate requests, certificate renewal and revocation requests, and CRL publication—and enables you to detect any unauthorized access or activity. This log is on by default.</p> <p>For more information, see “Configuring CMS Logs” on page 797 and “Monitoring Audit Logs” on page 808.</p>

Services That Are Logged

All major components and protocols (or services) of Certificate Management System log messages to log files. Table 23-2 lists services that are logged by default. If you want to view messages logged by a specific service, you can customize log settings accordingly. For details, see “Monitoring CMS Logs” on page 803.

Table 23-2 Services logged by Certificate Management System

Service	Description
ACLs	Specifies logged events related to access control lists.
Administration	Specifies logged events related to this server’s administration activities—that is, HTTPS communication between the CMS window and Certificate Management System.
All	Specifies logged events related to all the services.
Authentication	Specifies logged events related to this server’s activity with the authentication module.
Certificate Authority	Specifies logged events related to the Certificate Manager.
Database	Specifies logged events related to this server’s activity with the internal database.
HTTP	Specifies logged events related to the HTTP activity of the server.
Key Recovery Authority	Specifies logged events related to the Data Recovery Manager.
LDAP	Specifies logged events related to this server’s activity with the LDAP directory (used for publishing certificates and CRLs).
OCSP	Specifies logged events related to OCSP.
Others	Specifies logged events related to other activities of this server, such as command-line utilities and other processes.
Registration Authority	Specifies logged events related to the Registration Manager.

Table 23-2 Services logged by Certificate Management System *(Continued)*

Service	Description
Request Queue	Specifies logged events related to the request queue activity of this server.
User and Group	Specifies logged events related to users and groups managed by this server.

Log Levels (Message Categories)

For identification and filtering purposes, events logged by all CMS-supported services are classified into various categories. These are listed in Table 23-3. Each category represents messages that are of the same or a similar nature or that belong to a specific functional area. A particular log, for example the error log, can record entries that fall under one or more of these categories.

In the CMS configuration, each message category corresponds to a specific log level. Log levels are represented by numbers (digits) 1 to 6, each digit indicating the level of logging to be performed by the server—that is, how detailed the logging should be.

- A higher priority level (a larger digit) means less detail because only events of high priority are logged.
- A lower priority level (a smaller digit) means greater detail because more kinds of events are recorded in the log file.

Table 23-3 Classification of log entries or messages

Log level	Message category	Description
0	Debugging	These messages contain debugging information.
1	Informational (default selection for audit log)	These messages provide general information about the state of Certificate Management System. For example, status messages such as “Certificate Management System initialization complete” and “Request for operation succeeded” fall into this category.
2	Warning	These messages are warnings only and do not indicate any failure in the normal operation of the server.

Table 23-3 Classification of log entries or messages (*Continued*)

Log level	Message category	Description
3	Failure (default selection for system and error logs)	These messages indicate errors and failures that prevent the server from operating normally. Examples of messages that fall into this category include failures to perform a certificate service operation (“User authentication failed” or “Certificate revoked”) and unexpected situations that can cause irrevocable errors (“The server cannot send back the request it processed for a client through the same channel the request came from the client”).
4	Misconfiguration	These messages indicate that a misconfiguration in the server is causing an error.
5	Catastrophic failure	These messages indicate that because of an error, the service cannot continue running.
6	Security-related events	These messages identify occurrences that affect the security of the server (for example, “Privileged access attempted by user with revoked or unlisted certificate”).

You can use log levels to filter log entries based on the severity of an event. By default, a level 3 (Failure) is set for all services.

NOTE The log level is additive—that is, specifying a value of 3 causes levels 4, 5, and 6 to be logged. Log data can be voluminous, especially at lower (more verbose) logging levels. Make sure that the host machine has sufficient disk space for all the log files. It is also important to define your logging level, log rotation, log expiration, and server-backup policies appropriately so that all the log files are backed up and the host system doesn’t get overloaded; otherwise, you may lose information.

Log File Locations

For quick access, all the log files—system, error, and audit—are maintained in your local file system. Make sure that your storage capacity is sufficient for all your log files. A log file has the following default location:

```
<server_root>/cert-<instance_id>/logs
```

You can change the default location for logs by modifying it in the configuration.

Log File Naming Conventions

All log files created by Certificate Management System use one or the other of two naming conventions. There is one naming convention for active log files and one for rotated log files.

Active Log File Naming Convention

All active log files created by Certificate Management System use an identical naming convention. The name of an active log file is in the form `<log_type>.log`, where `<log_type>` specifies the log file type—whether it is system, error, or audit.

For example, an active error log file would be named `error.log`.

Rotated Log File Naming Convention

All rotated log files created by Certificate Management System use an identical naming convention. When Certificate Management System rotates an active log file, it renames the current log file and then creates a new log file with the original name. The rotated log file is saved with the original file type and an appended timestamp.

The name of a rotated log file is in the form `<log_type>.timestamp`, where the components of the filename indicate the following:

- `<log_type>` specifies the log file type—system, error, or audit—that has been rotated.
- `timestamp` is a large integer that indicates the date and time the corresponding active log file was rotated. The date and time have the forms `YYYYMMDD` (Year, Month, Day) and `HHmmSS` (Hour, Minute, Second), in that order.

For example, an error log file rotated on July 28, 1998 at 12:36:24 would be named `error.19980728123624`. Note that the timestamp is expressed in standard Unix time: the number of seconds since midnight January 1, 1970.

Buffered Versus Unbuffered Logging

Certificate Management System supports buffered logging for all three types of logs—system, error, and audit. You can choose to configure the server for either buffered or unbuffered logging (see “Configuring CMS Logs” on page 797).

If you configure Certificate Management System for buffered logging, the server creates buffers for the corresponding logs, and it holds the messages in these buffers for as long as possible. The server flushes out the messages to the log files—which are maintained in your local file system—only when either of the following conditions occurs:

- The buffer gets full—the buffer gets full when the buffer size is equal to or greater than the value specified by the `bufferSize` configuration parameter. The default value for this parameter is 512 KB.
- The flush interval for the buffer is reached—the flush interval is reached when the time interval since the last buffer flush is equal to or greater than the value specified by the `flushInterval` configuration parameter. The default value for this parameter is 5 seconds.
- When current logs are read from CMS window—the server retrieves the latest log when it is queried for current logs.

If you configure the server for unbuffered logging, the server flushes out messages as they are generated to the log files. Because the server performs an I/O operation (writing to the log file) each time a message is generated, configuring the server for unbuffered logging decreases performance.

Rotation of Log Files

Certificate Management System supports automatic rotation of log files, which simplifies administration and facilitates backups. You are not required to manually retire the current log file and create a new one to hold subsequent logged events. You can back up all but the current log file in a directory at any time, without stopping the server or manually notifying the server to start a new log file. The parameters that control log rotation are specified in the configuration. To change the log file rotation parameters, see “Configuring CMS Logs” on page 797.

You should periodically archive or back up the rotated log files. For details, see “Archiving of Rotated Log Files” on page 813.

Timing of Log File Rotation

Log files are rotated when either of the following conditions occur:

- The size limit for the corresponding file is reached—the size of the corresponding log file is equal to or greater than the value specified by the `maxFileSize` configuration parameter. The default value for this parameter is 100 KB.

- The age limit for the corresponding file is reached—the corresponding log file is equal to or older than the interval specified by the `rolloverInterval` configuration parameter. The default value for this parameter is 2592000 seconds (every hour).

Both these parameters can be specified from the CMS window; see “Configuring CMS Logs” on page 797.

Location of Rotated Log Files

Rotated log files are stored at the same location where the current or active log files are maintained. To find out the default location of an active log file, see “Log File Locations” on page 793.

Deletion of Log Files

Certificate Management System supports automatic deletion of rotated (or old) log files.

How to Conserve Disk Space

By default, Certificate Management System does not delete rotated log files automatically. Because the rotated log files are also saved in your local file system, these files eventually take up a considerable amount of disk space. You can avoid this problem by doing one of the following:

- Configure the server to automatically delete the rotated log files.
- Manually delete the log files from the local file system.

In either case, if you want to keep specific log files for future use, be sure to archive or back them up before they are deleted. For details, see “Archiving of Rotated Log Files” on page 813.

Timing of Log File Deletion

If you configure Certificate Management System to delete rotated log files automatically, the server deletes these files when the life of the corresponding log file is equal to or older than the interval specified by the `expirationTime` configuration parameter; the interval must be specified in seconds. By default, the rotated log files are not deleted. If you want the files to be deleted, you must change the default value as appropriate. For example, if you want the files to be deleted every 30 days, you would change the value to 2592000 (60x60x24x30).

Configuring CMS Logs

This section explains how to configure Certificate Management System to log messages so that you can monitor the server:

- Step 1. Before You Begin
- Step 2. Modify the Existing Listeners
- Step 3. Delete Unwanted Listeners
- Step 4. Create New Listeners

Step 1. Before You Begin

Before configuring a CMS instance:

- Read section “Introduction to Logs” on page 789.
- Read Chapter 8, “Log Plug-in Modules” of *CMS Plug-Ins Guide*.

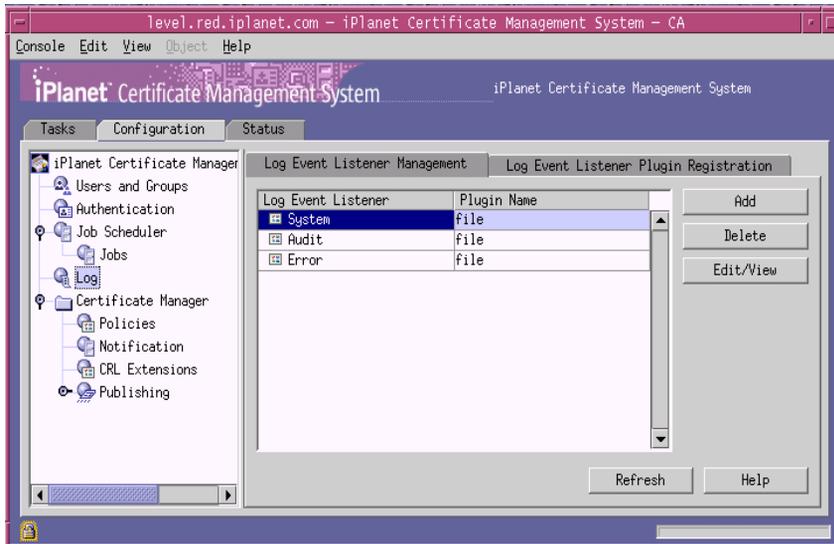
Step 2. Modify the Existing Listeners

When you create a CMS instance, a set of log-event listeners (that you would most likely want to use) are automatically created using the log modules registered by default.

- `Audit`
- `Error`
- `System`
- `NTAudit` (only on a Windows NT system)
- `NTSystem` (only on a Windows NT system)

Note that `Audit`, `Error`, and `System` listeners are created using the `file` module and `NTAudit` and `NTSystem` listeners are created using the `NTEventLog` module.

Figure 23-1 shows the log-event listeners created for a CMS instance installed on a Windows NT system.

Figure 23-1 Default log-event listeners of a Certificate Manager

After installation, you must verify whether you want to use these listeners, check how these listeners are configured, and make the appropriate configuration changes.

You can modify a log-event listener by editing its configuration parameter values; you cannot edit the name of a listener. To change the name of a listener, you need to create a new listener exactly like the listener you want to rename, except with a new name, and delete the old listener.

As a part of editing a listener, you can change its status from enabled to disabled or vice versa by checking or unchecking the `enabled` parameter. Listeners that are in a disabled state do not record any events.

If you don't want to use a listener, delete it from the configuration as explained in "Step 3. Delete Unwanted Listeners" on page 799; alternatively, you may keep it in the disabled state. If you want to create a new listener, you can do so as explained "Step 4. Create New Listeners" on page 800.

To configure audit, error, and system logs for a CMS instance:

1. Log in to the CMS window (see "Logging In to the CMS Window" on page 351).

2. In the navigation tree, select Logs.

On the right pane, the Log Event Listener Management tab appears. It lists the currently configured listeners.

3. In the Log Event Listener list, select a listener that you want to modify.

For the purposes of this instruction, assume that you selected the listener named `Audit`.

4. Click Edit/View.

The Log Event Listener Editor window appears, showing how this listener is configured. An example is shown below.

Log Event Listener Editor

Log Event Listener ID: Audit
Log Event Listener Plugin ID: file

type: audit

enabled:

level: Information

fileName: d:/netscape/server4/cert-demoCA/logs/audit

bufferSize: 512

flushInterval: 5

maxFileSize: 100

rolloverInterval: Monthly

expirationTime: 0

The maximum time in seconds before the buffer is flushed to the file

OK Cancel Help

5. Make the necessary changes and click OK.

You are returned to the Log Event Listener Management tab.

6. Repeat steps 3 through 5 for the remaining rules.
7. Click Refresh.

Step 3. Delete Unwanted Listeners

You can delete any unwanted log-event listeners from the CMS configuration. If you think you might need a listener in the future, instead of deleting it from the configuration you should disable it by unchecking the `enabled` parameter. In this way, you can avoid re-creating the listener in the future.

To delete a listener from the CMS configuration:

1. In the Log Event Listener Management tab, select the listener you want to delete and click Delete.
2. When prompted, confirm the delete action.

The CMS configuration is modified. If the changes you made require you to restart the server, you will be prompted accordingly. Don't restart the server yet; you can do so after you've made all the required changes.

Step 4. Create New Listeners

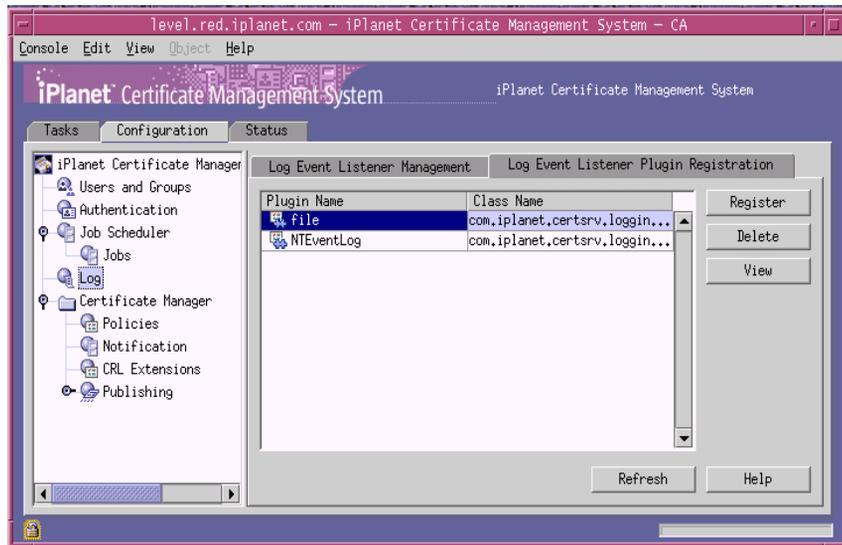
This generally not required. However, if you want to create a new listener, follow the procedure in this section.

Adding a log-event listener to the CMS configuration involves creating a new instance of an already registered log plug-in module, assigning a unique name for the instance, and entering appropriate values for the parameters that define the module you want to create an instance of.

When you add a listener, the CMS configuration gets updated with the relevant information. Keep the following points in mind:

- When naming a listener, be sure to formulate the name using any combination of letters (aA to zZ), digits (0 to 9), an underscore (`_`), and a hyphen (`-`); other characters and spaces are not allowed. For example, you can type `Audit_Log_Listener` or `AuditLogListener` as the instance name, but not `Audit Log Listener`.
- The status of the listener, enabled or disabled, depends on whether you check or uncheck the `enabled` parameter; only an enabled listener records events.

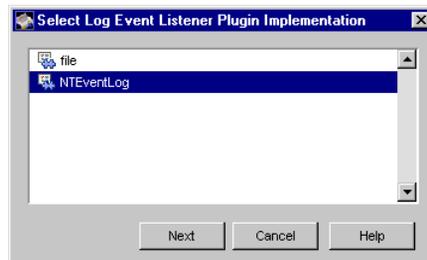
Figure 23-2 shows the log modules registered with a Certificate Manager. If you have registered any custom policy modules (see “Registering a Log Module” on page 816), they too will be available for selection.

Figure 23-2 Default log modules registered with a Certificate Manager

To add a new listener to the CMS configuration:

1. In the Log Event Listener Management tab, click Add.

The Select Log Event Listener Plugin Implementation window appears. It lists registered log modules.



2. Select a plug-in module.

For the purposes of this instruction, assume that you selected the `file` module.

3. Click Next.

The Log Event Listener Editor window appears. It lists the configuration information required for this listener.

4. Enter the appropriate information:

Log Event Listener ID. Type a unique name that will help you identify the listener; be sure to use an alphanumeric string without spaces.

type. Select `audit` to create a listener that records audit logs. For error and system logs, select `system`. For more information, see “Logs Maintained by the Server” on page 790.

enabled. Select this box.

level. From the drop-down list, select a log level. The choices are `Debug`, `Information`, `Warning`, `Failure`, `Misconfiguration`, `Catastrophe`, and `Security`. The default selection is `Failure`. For more information, see “Log Levels (Message Categories)” on page 792.

fileName. Type the full path, including the filename, to the file to write messages. (Make sure that the server has read/write permission to the file.)

bufferSize. Type the buffer size in kilobytes (KB) for the log. The default size is 512 KB. For more information, see “Buffered Versus Unbuffered Logging” on page 794.

flushInterval. Type the interval, in seconds, to flush the buffer to the file. The default interval is 5 seconds.

maxFileSize. Type the file size in kilobytes (KB) for the error log. The default size is 100 KB. For more information, see “Timing of Log File Rotation” on page 795.

rolloverInterval. From the drop-down list, select the frequency at which the server should rotate the active error log file. The available choices are Hourly, Daily, Weekly, Monthly, and Yearly. The default selection is Monthly. For more information, see “Rotation of Log Files” on page 795.

expirationTime. Type, in seconds, the age limit for deleting the rotated log files. The default value is 0 seconds, which indicates that the rotated log files should not be deleted. For more information, see “Deletion of Log Files” on page 796.

5. Click OK.

You are returned to the Log Event Listener Management tab.

6. Repeat steps 1 through 5 and create additional rules, if required.

Monitoring CMS Logs

When you have problems with Certificate Management System that require troubleshooting, you may find it helpful to check the error or informational messages that the server has logged. Also, by examining the log files you can monitor many aspects of the server’s operation.

To facilitate this, the CMS window provides a simple mechanism for viewing the contents of both currently active and rotated audit, system, and error log files. The contents of the log file you choose to view are displayed in the form of a table. Each row is allocated to a specific log entry, with columns containing information such as the date and time the message was logged, the severity of the message, and a general description of the log. Once you open a log file for viewing, you can also do the following tasks:

- Read log file contents partially (by specifying the number of entries to be displayed)
- Filter log entries for specific services (by specifying the source)

This section covers the following topics on monitoring Certificate Management System by viewing log contents:

- Monitoring System Logs
- Monitoring Error Logs

- Monitoring Audit Logs
- Using System Tools for Monitoring the Server (Windows NT Only)

Monitoring System Logs

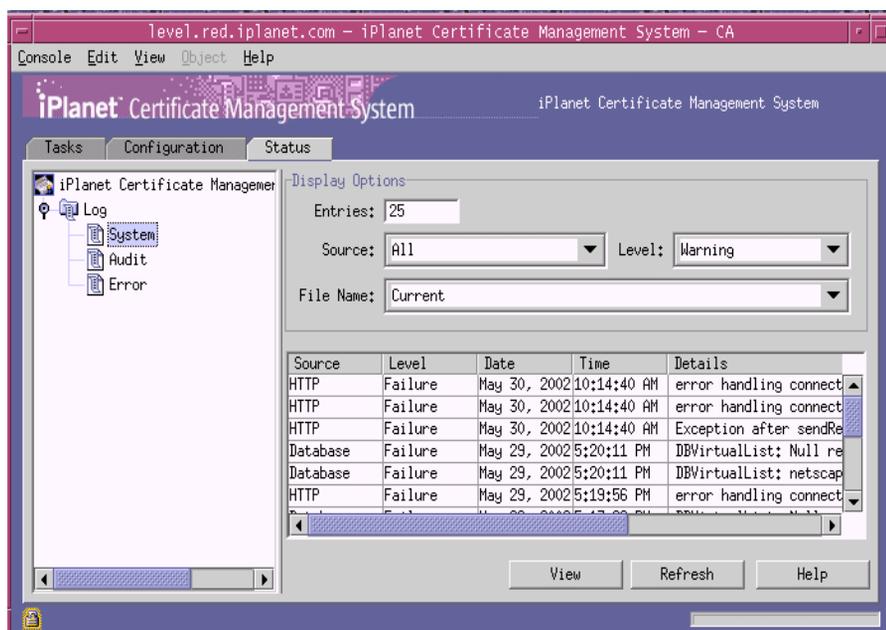
Certificate Management System maintains extensive system logs. These logs record various events and system errors for system monitoring and debugging. A system log records details such as the following:

- Each HTTP access invoked on the server.
- Errors encountered, such as authentication failures, malformed universal resource indicators (URIs), invalid database password indications, and server start-up and shut-down messages.
- Messages related to the status of certificate issuance or revocation, authentication failures for issuing-agent connections, and any errors related to the formatting of requests.

You can view the contents of currently active as well as rotated system log files from the CMS window (see Figure 23-3).

If you have installed Certificate Management System on a Windows NT system, you can configure the server to log messages to Windows NT event log. For details, see “Logging to Windows NT Event Log” on page 811.

Figure 23-3 A sample active system log displayed in the CMS window



To view the contents of an active or rotated system log file:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Status tab.
3. In the navigation tree, under Logs, select System.
4. In the Display Options section, specify your viewing preferences:

Entries. Type the maximum number of entries to be displayed. When this limit is reached, Certificate Management System returns any entries it has located that match the search request. If you enter zero (0), no messages are returned. If you leave the field blank, the server returns every matching entry (no limit) regardless of the number found.

Source. Select the CMS component (or service) for which log messages are to be displayed. Depending on the components that write to this log file, the drop-down list shows one or more of the following: All, Registration Authority, Certificate Authority, Key Recovery Authority, HTTP, Internal

Database, Authentication, Administration, LDAP, Request Queue, ACLs, User and Group, OCSP, and Others. If you choose All, messages logged by all components that log to this file are displayed. For more information, see “Services That Are Logged” on page 791.

Level. Select a message category that represents the log level for filtering messages. For more information on log levels, see “Log Levels (Message Categories)” on page 792.

Filename. Select the log file you want to view. Choose Current to view the currently active system log file. For more information, see “Log File Naming Conventions” on page 794.

5. Click Refresh.

The table displays the system log entries. The entries are in reverse chronological order, with the most current entry placed at the top. Use the scroll arrows on the right edge of the panel to scroll through the log entries.

For each entry you see the following details:

Source. Indicates the CMS component or resource that logged the message.

Level. Indicates the severity of the corresponding entry (explained Table 23-3 on page 792).

Date. Indicates the date on which the entry was logged.

Time. Indicates the time at which the entry was logged.

Details. Provides a brief description of the log.

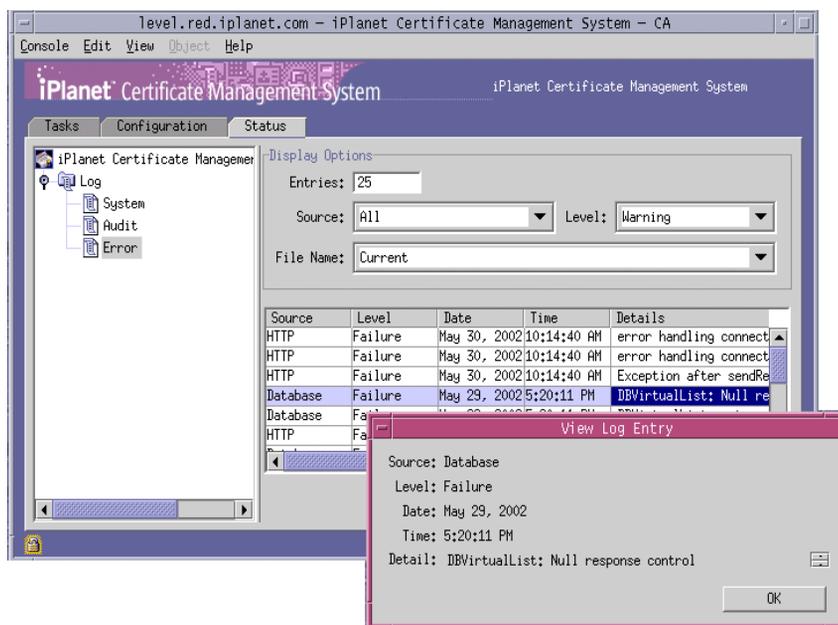
6. To view an entry in its entirety, either double-click it or select the entry and click View.

Monitoring Error Logs

The error log file contains errors the server has encountered since the log file was created; it also contains informational messages about the server, such as when the server was started. Incorrect user authentication is also recorded in the error log. Use the error log to find broken URL paths or missing files.

You can view the contents of currently active as well as rotated error log files from the CMS window (see Figure 23-4).

Figure 23-4 A sample active error log displayed in the CMS window



To view the contents of an active or rotated error log file:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Status tab.
3. In the navigation tree, under Logs, click Error.
4. In the Display Options section, specify your viewing preferences:

Entries. Type the maximum number of entries to be displayed. When this limit is reached, Certificate Management System returns any entries it has located that match the search request. If you enter zero (0), no messages are returned. If you leave the field blank, the server returns every matching entry (no limit) to the client regardless of the number found.

Source. Select the CMS component (or services) for which log messages are to be displayed. Depending on the components that write to this log file, the drop-down list shows one or more of the following: All, Registration Authority, Certificate Authority, Key Recovery Authority, HTTP, Internal

Database, Authentication, Administration, LDAP, Request Queue, ACLs, User and Group, OCSP, and Others. If you choose All, messages logged by all components that log to this file are displayed. For more information, see “Services That Are Logged” on page 791.

Level. Select a message category that represents the level of logging to filter messages. For more information, see “Log Levels (Message Categories)” on page 792.

Filename. Select the log file you want to view. Choose Current to view the currently active error log file. For more information, see “Log File Naming Conventions” on page 794.

5. Click Refresh.

The table displays the error log entries. The entries are in reverse chronological order, with the most current log placed at the top. Use the scroll arrows on the right edge of the panel to scroll through the log entries.

For each entry you see the following details:

Source. Indicates CMS component or resource that logged the message.

Level. Indicates the severity of the corresponding entry (explained in Table 23-3 on page 792).

Date. Indicates the date on which the entry was logged.

Time. Indicates the time at which the entry was logged.

Details. Provides a brief description of the log.

6. To view an entry in its entirety, either double-click it or select the entry and click View.

Monitoring Audit Logs

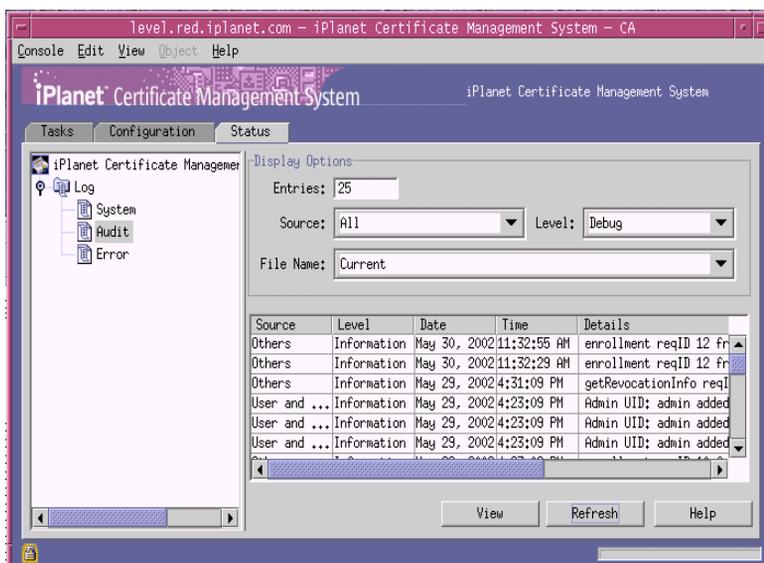
Certificate Management System maintains audit trails for all events—certificate requests, certificate renewal and revocation requests, CRL publication, and so on. These trails enable you to detect any unauthorized access or activity. The audit trails are logged and maintained in a file in your local file system.

If you have installed Certificate Management System on a Windows NT system, you can also configure the server to log audit messages to Windows NT event log. For details, see “Logging to Windows NT Event Log” on page 811.

NOTE You should periodically examine and audit the CMS audit log for unusual activity. When examining the log, note in particular the log entries that fall under the Security-Related Events category (these are labeled Security).

You can view the contents of currently active as well as rotated audit log files from the CMS window (see Figure 23-5).

Figure 23-5 A sample active audit log displayed in the CMS window



To view the contents of an active or rotated audit log file:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Status tab.
3. In the navigation tree, under Logs, select Audit.

4. In the Display Options section, specify your viewing preferences:

Entries. Type the maximum number of entries to be displayed. When this limit is reached, Certificate Management System returns any entries it has located that match the search request. If you enter zero (0), no messages are returned. If you leave the field blank, the server returns every matching entry (no limit) regardless of the number it finds.

Source. Select the CMS component (or resource) for which log messages are to be displayed. Depending on the components that write to this log file, the drop-down list shows one or more of the following: All, Registration Authority, Certificate Authority, Key Recovery Authority, HTTP, Internal Database, Authentication, Administration, LDAP, Request Queue, ACLs, User and Group, OCSP, and Others. If you choose All, messages logged by all components that log to this file are displayed. For more information, see “Services That Are Logged” on page 791.

Level. Select a message category that represents the level of logging to filter messages. For more information, see “Log Levels (Message Categories)” on page 792.

Filename. Select the log file you want to view. Choose Current to view the currently active audit log file. For more information, see “Log File Naming Conventions” on page 794.

5. Click Refresh.

The table displays the audit log entries. The entries are in reverse chronological order, with the most current log placed at the top. Use the scroll arrows on the right edge of the panel to scroll through the log entries.

For each entry you see the following details:

Source. Indicates the CMS component or resource that wrote to the log file.

Level. Indicates the severity of the corresponding entry (explained in Table 23-3 on page 792).

Date. Indicates the date on which this entry was logged.

Time. Indicates the time at which this entry was logged.

Details. Provides a brief description of the log.

6. To view an entry in its entirety, either double-click it or select the entry and then click View.

Using System Tools for Monitoring the Server (Windows NT Only)

If you have installed Certificate Management System on a Windows NT system, you can monitor the server with the system tools provided by Windows NT. This section explains how you can use the system tools.

- Logging to Windows NT Event Log
- Using Event Viewer
- Avoiding Event Log From Getting Filled

Logging to Windows NT Event Log

In addition to logging messages to the log files maintained in your local file system, Certificate Management System can also log audit messages and system errors to the Windows NT Event log. The CMS window allows you to turn this feature on or off and to specify the levels for logging. To configure the server to log messages to the Event log, see “Configuring CMS Logs” on page 797.

Note that by default Certificate Management System is configured to write both audit and system logs to the Windows NT Event log.

Using Event Viewer

Once you configure Certificate Management System to write audit and system logs to the Event log of a Windows NT system, you can use the system’s tool called *Event Viewer* to monitor events related to your server. For more information about the Event Viewer, check your system documentation.

To monitor Certificate Management System by using Event Viewer:

1. In the Administrative Tools program group, double-click the Event Viewer icon.
2. From the Log menu, select Application.

The Application log appears in Event Viewer. In this log, the *source* of any messages from iPlanet Certificate Management Server is the server’s instance ID (if you didn’t change the default value assigned to the `NTEventSourceName` parameter).

3. From the View menu, choose Find to search for one of the iPlanet labels in the log; use Refresh to see updated log entries.

4. Double-click a log entry to see additional information.

The mapping between the CMS log levels and the Windows NT event type is shown in Table 23-4.

Table 23-4 Mapping between Windows NT log event type and CMS logs

Windows NT log event type	CMS log level
Information	Debugging (0)
Information	Informational (1)
Warning	Warning (2)
Error	Failure (3)
Error	Misconfiguration (4)
Error	Catastrophic failure (5)
Error	Security-related events (6)

Avoiding Event Log From Getting Filled

When running Certificate Management System on a Windows NT system, if you don't configure the NT Event Log properly, the event log will get full. When this happens, you'll see an error message (see Figure 23-6) stating that the application log file is full.

Figure 23-6 Error message indicating event log is full

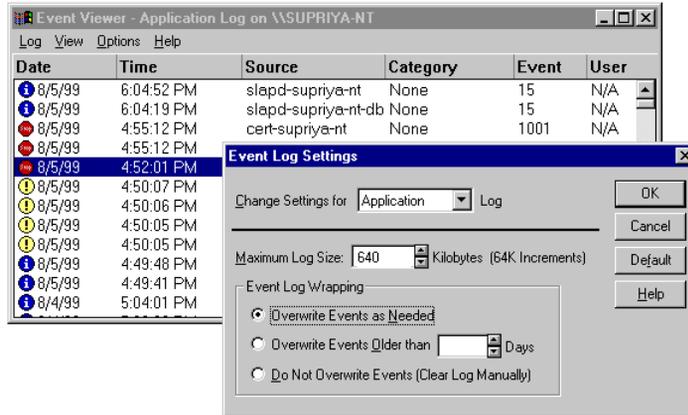


If you see this dialog box, you must clean up the application log immediately. Here's what you should do:

1. From the Start menu on your desktop, select Programs, Administrative Tools (Common), and Event Viewer, in that order.

This opens the Event Viewer window for the system.

2. From the Log menu, select Application.
A checkmark to the left indicates it is selected.
3. From the Log menu, select Log Settings.
This opens the Event Log Settings window.



4. Enter the appropriate values:
 - Change Settings for.** Make sure that the Application log is selected in this box.
 - Maximum Log Size.** Select a reasonable size so that the event log doesn't get full in a short period of time.
 - Event Log Wrapping.** Select the "Overwrite Events as Needed" option.
5. Click OK.
6. Close the Event Viewer window.

Archiving of Rotated Log Files

Log files, especially the audit log file, contain critical information. So it is good practice to periodically archive rotated log files to some archive media. Consider doing this whether you are manually deleting rotated log files or have configured the server to delete files automatically. You can archive log files by copying the entire `log` directory to your archive media.

Certificate Management System does not provide any tool or utility for archiving log files. Use the tools or utilities that your operating system provides for archiving.

Certificate Management System does, however, provide a command-line utility, called `signtool`, that allows you to sign log files before archiving them. This gives you a means of tamper detection. For details, see “Signing Log Files” on page 814.

Signing Log Files

Certificate Management System allows you to digitally sign log files before you archive them or distribute them for audit purposes. This feature enables you to check whether the log files have been tampered with since being signed.

For signing log files, you use a command-line utility called *Signing Tool*; for details about this utility, check Chapter 13, “Signing Tool” of *CMS Command-Line Tools Guide*. To locate an online version of this document, see “Where to Go for Related Information” on page 29. The utility uses information in the certificate (`cert7.db`), key (`key3.db`), and security module (`secmod.db`) databases of Certificate Management System.

Before you begin signing the log files, follow these guidelines:

- Determine the key pair you want to use for signing the log directory. Typically, you should use the Certificate Manager’s (the CA’s) signing key pair. Also find out the nickname of the certificate that corresponds to this key pair.
- If you have deployed many CAs, locate the CMS instance in which the CA you want to use is installed.
- Find out whether the key pair is in an internal or external token. If it is in an external token, make sure the token is currently installed. You will also need to know the password for the token.
- Determine which log files need to be signed. Put all the files that need to be signed in one or more directories. (The utility can sign a directory containing files; it cannot sign individual files.) Make sure these directories are in the host machine in which the CA is installed.
- Determine names for the output files; the output you receive will be a JAR file (which is a signed zip file). You may want to give names that will help you identify these JAR files easily in the future.

When you are ready with all this information, follow the procedure below to sign the log directories:

1. Go to the CMS instance in which the CA whose key pair you want to use for signing is installed.

2. Copy the security module database (`secmod.db` file) from the Administration Server configuration directory to the CMS configuration directory.

The security module database is in this directory:

```
<server_root>/admin-serv/config
```

Copy it to this directory:

```
<server_root>/cert-<instance_id>/config
```

3. Open a terminal window.
4. At the command prompt, run the following command with the appropriate information:

```
signtool -d <secdb_dir> -k <cert_nickname> -Z <output> <input>
```

`<secdb_dir>` specifies the path to the directory that contains the certificate, key, and security module databases for the CA. This must be the same path you used to copy the security module database in step 2.

`<cert_nickname>` specifies the nickname of the certificate you want the utility to use for signing.

`<output>` specifies the name of the JAR file (a signed zip file).

`<input>` specifies the path to the directory that contains the log files.

For example, in a Windows NT system, your command might look like this:

```
signtool -d c:\iplanet\servers\cert-testCA\config -k
testCAsigningcertificate -Z log_err_02_99.jar
c:\archive\logs
```

where `c:\iplanet\servers\cert-testCA\config` is the path to the certificate, key, and security module databases (`secdb_dir`).

`testCAsigningcertificate` is the certificate nickname (`cert_nickname`).

`log_err_02_99.jar` is the name of the JAR file (output).

(`input`) is `c:\archive\logs` is the directory to be signed.

Managing Log Modules

This section explains how to use the CMS window to perform the following operations:

- Registering a Log Module
- Deleting a Log Module

For information on adding or changing policy-specific information in the configuration file, see “Changing the Configuration by Editing the Configuration File” on page 359.

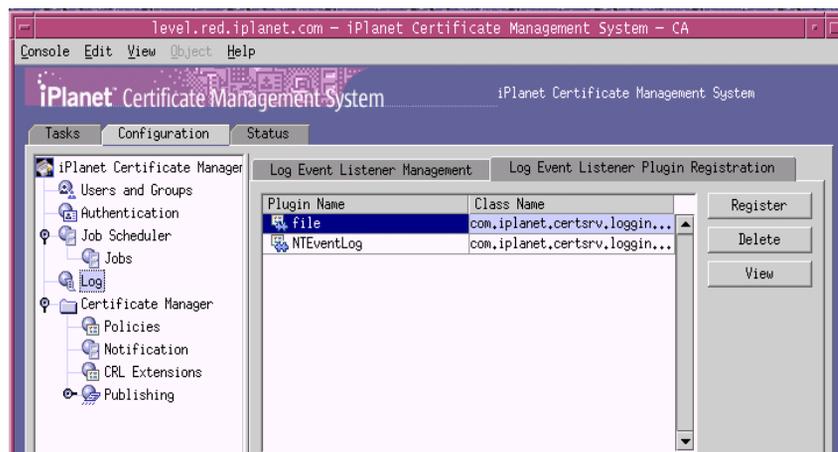
Registering a Log Module

You can register new log plug-in modules using the CMS window. Registering a new module involves specifying the name of the module and the full name of the Java class that implements the log interface.

Before registering a plug-in module, be sure to put the Java class for the module in the `classes` directory (the implementation must be on the class path).

To register a log plug-in module with a CMS instance:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select Logs, and then in the right pane, select the Log Event Listener Plugin Registration tab.



4. Click Register.

The Register Log Event Listener Plugin Implementation window appears.

5. Specify information as appropriate:

Plugin name. Type a name for the plug-in module.

Class name. Type the full name of the class for this module—that is, the path to the implementing Java class. If this class is part of a package, be sure to include the package name. For example, if you are registering a class named `customLog` and if this class is in a package named `com.myCompany`, type `com.myCompany.customLog`.

6. Click OK.

You are returned to the Log Event Listener Plugin Registration tab.

7. To view the updated configuration, click Refresh.

Deleting a Log Module

You can delete unwanted log plug-in modules using the CMS window. Before deleting a module, be sure to delete all the listeners that are based on this module; see “Step 3. Delete Unwanted Listeners” on page 799.

To delete a module:

1. Log in to the CMS window (see “Logging In to the CMS Window” on page 351).
2. Select the Configuration tab.
3. In the navigation tree, select Logs, and then in the right pane, select the Log Event Listener Plugin Registration tab.
4. In the Plugin Name list, select the module you want to delete and click Delete.
5. When prompted, confirm the delete action.

Issuing and Managing Certificates

Chapter 24, “Issuing and Managing Server Certificates”

Chapter 25, “Setting Up CEP Enrollment”

Issuing and Managing Server Certificates

This chapter explains how you can use iPlanet Certificate Management Server (CMS) to issue and manage SSL server certificates.

The chapter has the following sections:

- Certificate Issuance to Servers (page 821)
- Getting Server SSL Certificates for iPlanet Servers (page 824)
- Renewal of Server Certificates (page 831)
- Revocation of Server Certificates (page 831)

Certificate Issuance to Servers

For Certificate Management System to generate a server certificate, it must receive the certificate signing request (CSR) from the server that needs the certificate. This request must be initiated by the administrator of the specific server requiring the certificate.

SSL-enabled servers (or servers that are capable of using certificates for security) provide mechanisms for generating a CSR based on new or existing key pairs. For example, iPlanet servers come with a wizard that walks an administrator through the entire process of requesting a server certificate and installing it in the server's certificate database. For information on this wizard, see "Obtaining and Installing a Certificate" in *Managing Servers with iPlanet Console*.

Once an administrator generates a CSR for a server, he or she must paste it into the appropriate server enrollment form hosted by a Registration Manager or Certificate Manager, and then submit the request. Upon receipt of the request, Certificate Management System responds as follows:

1. Verifies the validity and authenticity of the request.

The authentication mechanism that Certificate Management System uses is based on the authentication mechanism specified in the enrollment form the administrator uses to submit the certificate request. For example, if the enrollment form was configured to employ directory-based authentication, Certificate Management System checks the configured directory for the appropriate information. On the other hand, if the enrollment form specifies manual authentication, the request gets queued and awaits approval by an agent.

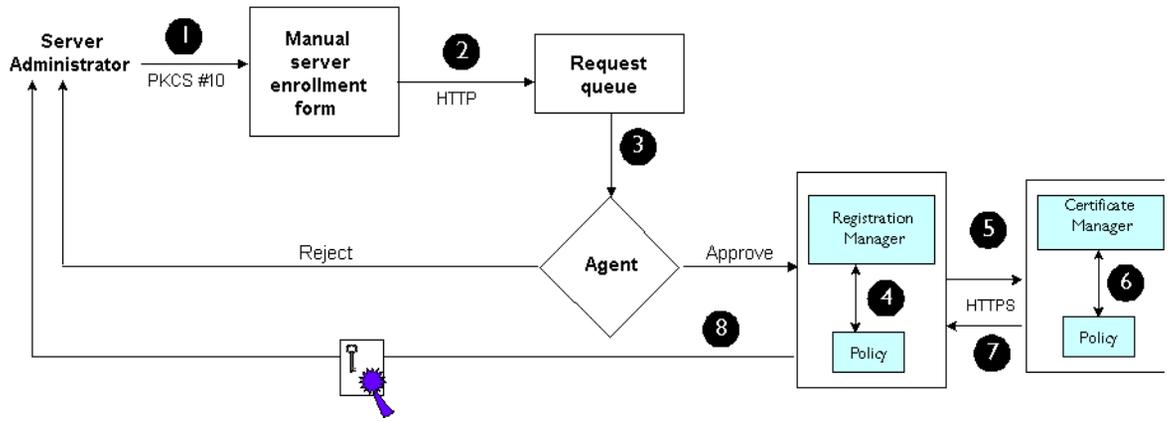
2. Subjects the request to policy checks.

If the request passes all the policy rules, Certificate Management System generates the server certificate and sends it to the email address specified in the server certificate request (the enrollment form includes a field for the administrator to enter this information). Otherwise, Certificate Management System logs an error message.

Upon receipt of the certificate, the server administrator installs the certificate in the server's certificate database.

How the Manual Server Enrollment Process Works

Figure 24-1 illustrates how Certificate Management System issues a server certificate in a deployment scenario involving a Registration Manager acting as an enrollment authority to a Certificate Manager. The server certificate is requested via a manual enrollment form hosted by the Registration Manager.

Figure 24-1 Server (or site) certificate issuance

These are the steps shown in Figure 24-1:

1. The server administrator goes to the manual enrollment form hosted by the Registration Manager, pastes in the certificate signing request in PKCS #10 format, completes the other information in the enrollment form, and submits the form.

(If the enrollment port is HTTPS, the administrator should visit the link that delivers the CA's certificate chain and download the chain into the browser that he or she will use for server enrollment.)

2. The Registration Manager verifies the authenticity of the request. Because the request requires manual authentication, the Registration Manager stores the request in the queue for agent approval.
3. An agent processes the request and either rejects or approves it.
4. The Registration Manager picks up the approved request and subjects it to policy checks.
5. If the request passes the Registration Manager's policy checking, the Registration Manager submits the request to the Certificate Manager for signing. The Certificate Manager verifies the authenticity of the Registration Manager by verifying the certificate presented by it. If it is a trusted Registration Manager, the Certificate Manager accepts the request.
6. The Certificate Manager subjects the request to its own policy checks.

7. If the request passes Certificate Manager's policy, it signs the request immediately and returns the certificate to the Registration Manager. The Registration Manager then delivers the certificate to the administrator. Optionally, the Certificate Manager may publish the certificate to the corporate directory.

If the Certificate Manager's policy requires additional information, the administrator will be directed to return later to pick up the certificate. The administrator may need to query the Registration Manager using the certificate request number to see whether the certificate has been issued. Alternatively, the Registration Manager can be configured to email the user when the certificate is ready for pick up. See "Notifications of Certificate Issuance to End Entities" on page 576.

8. The Registration Manager delivers the server SSL certificate to the email address specified in the enrollment form. Optionally, the Registration Manager may publish the certificate to the corporate directory.

Getting Server SSL Certificates for iPlanet Servers

To enable a server to establish SSL connections, you need to get a certificate that identifies the server. You can get a certificate for a server by submitting a request to Certificate Management System.

To generate the actual request, you (or the server administrator) need to use the server that requires the certificate. This is required because the private key must be stored with the server that will use it.

The following section explains how to request a server SSL certificate for iPlanet servers. The instructions apply mainly to requests from servers other than CMS subsystem server—for example, Web, Administration, and Directory Servers. To request a certificate for a CMS subsystem, follow the instructions in "Getting New Certificates for the Subsystems" on page 507.

Getting Certificates for Version 3.x Servers

To get a certificate for a server in the Netscape version 3.x server family (for example, Netscape Administration Server 3.x) follow the procedure below:

- Step 1. Generate the Server Certificate Request

- Step 2. Submit the Server Certificate Request
- Step 3. Install Your Server's SSL Certificate
- Step 4. Accept a CA as Trusted in Your Server
- Step 5. Verify Your Server's SSL and CA Certificates

Step 1. Generate the Server Certificate Request

To generate the certificate signing request (CSR) for a server:

1. Open a web browser window.
2. Go to the Administration Server, and use the Server Selector to access the Server Manager for your server.
3. Follow the directions presented there to generate a new key pair which you will then get certified (you will use this key pair to generate a certificate signing request).

Alternatively, you can use any other tool provided with your server to generate the key pair; see the documentation for your server.

4. Once you have generated a key pair, follow the directions presented to generate a certificate signing request (CSR).
5. In the Certificate Authority field, enter your own email address.

The server mails the request to the address specified in this field.

6. Submit the form.

The server generates and displays a CSR.

7. Copy the CSR, including the -----BEGIN NEW CERTIFICATE REQUEST----- and -----END NEW CERTIFICATE REQUEST----- marker lines, to a text file. For example:

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

```
MIIBBzCBsgIBADBPmQswCQYDVQQGEwJVUzEoMCYGA1UEChMfTmV0c2NhcGUgRGly
ZWN0b3J5IFB1YmtpY2F0aW9uczEWMBQGA1UEAxMNZHVtcC5tY29tLmNvbTBaMA0G
CSqGSIb3DQEBAQU2nfjiMEYCCQCKsMRaLGdfp4m0OiGcgljG5KgOsYRNvwGYW7kf
W+8mmijDtZRjYNjjcgpF3VnlsbxbclX9LVjjNLC57u37XZdAgEDoAAwDQYJKoZIh
vcNAQEEBQADQCCYUtnUtCVGyNrYGSfydclqiovyx1fRD1z23zg+eBPK7n85UyE4r
5zGZjDsMYr172ytfAFL7DeG83DWzr8Z
```

```
-----END NEW CERTIFICATE REQUEST-----
```

Next, you need to paste this request into the server enrollment form hosted by Certificate Management System.

Step 2. Submit the Server Certificate Request

To submit the server certificate request to Certificate Management System:

1. Open a web browser.
2. Go to the server enrollment form (the page that allows you to submit a server certificate request).

By default, the enrollment forms are at this location:

```
https://<hostname>:<end_entity_HTTPS_port> OR  
http://<hostname>:<end_entity_HTTP_port>
```

3. In the Enrollment tab, unselect Server, select SSL Server.

The form for requesting SSL server certificate appears.

4. Complete the request form with the information that Certificate Management System needs to create a certificate for your server.

In general, you will be required to enter the following information:

- In the certificate request text area, paste the CSR that you copied to the text file, including the -----BEGIN NEW CERTIFICATE REQUEST----- and -----END NEW CERTIFICATE REQUEST----- marker lines.
- In the contact information section, enter values to identify yourself. These values will be used by the CA, if the need arises. For example, if there are any questions or problems with the certificate request, the CA administrator or agent will use this information to contact you. Also, be sure to enter your email address. This is the address where the CA will send the certificate once it has been issued.
- In the additional comments section, enter any additional information that might help the issuing agent process the request. For example, you might want to enter the name of the person who instructed you to obtain a certificate or some other administrative information.

5. Submit the request.

You should receive notification from Certificate Management System or an issuing agent (depending on which enrollment form you used) when your request is processed. The notification will contain your certificate, along with information on how to install the new certificate into your server. The notification may also mention that you need to install the CA's certificate as a trusted CA. Check the notification message for details.

Step 3. Install Your Server's SSL Certificate

To install the server SSL certificate on your server:

1. Open a web browser window.
2. Go to the Administration Server, and use the Server Selector to access the Server Manager for your server.
3. Follow the directions presented there to install the certificate.

In general, you will be required to specify or enter the following information:

- Whether the certificate is for this server. Be sure to select the option that says the certificate is for this server.
 - A name (or nickname) for the certificate. This name will be displayed in the list of certificates installed on this server.
 - The certificate, in base-64 encoded format. Open the email sent to you by the CA, locate and copy the portion that begins with -----BEGIN CERTIFICATE----- and ends with -----END CERTIFICATE-----, and paste it into the text area in the form.
 - The encryption alias. Enter the alias for your server.
4. Follow the prompts and add the certificate to your server's certificate database.
 5. Stop and restart Administration Server for the changes to take effect.

The server decrypts the message, extracts the certificate, and saves it to the directory you specified.

Step 4. Accept a CA as Trusted in Your Server

In both Netscape clients and iPlanet servers, CAs can be either *trusted* or *untrusted*. If a CA is trusted, Netscape clients and iPlanet servers accept the certificates that have been issued by that CA. For the server to accept (during SSL client authentication) client certificates that have been issued by Certificate Management System, you must import its certificate chain into the certificate database of your server.

To view this chain in a format that can be used by iPlanet servers:

1. Go to the home page of Certificate Management System.

By default, the home page is at this location:

`https://<hostname>:<end_entity_HTTPS_port>`

2. Click Accept "This Authority in Your Server."

3. Specify how you want Certificate Management System to display the certificate chain.

You can choose to display the entire certificate chain (in a single block) or individual certificates in the chain. The entire certificate chain is in PKCS #7 format. If you are using an older server that does not recognize the complete certificate chain format, you may need to display each individual certificate in the chain (for example, a version earlier than Netscape server 2.0 releases).

4. Specify how you want to trust this CA.

You can choose to trust only the CA you are accessing or all authorities whose certificates are included in the chain.

5. Click Present Certificate Chain.

If you chose to display the whole chain for importing into your server, the certificate chain is displayed in a format similar to this:

```
-----BEGIN CERTIFICATE-----
MIIBtgYJYIZIAYb4QgIFoIIBPzCCAZ8wggGbMIIBRaADAgEAAgEBMA0GCSqGSIb3
DQEBBAUAMFcxCAJBGnVBAYTA1VTMSwwKgYDVQQKEYNOZXRzY2FwZSBDb21tdW5p
Y2F0aW9ucyBDb3Jwb3JhdG1vb3JlbnQxMjE2LjE2LjE2LjE2LjE2LjE2LjE2LjE2
HhcNOTYxMjE2LjE2LjE2LjE2LjE2LjE2LjE2LjE2LjE2LjE2LjE2LjE2LjE2LjE2
MCoGA1UEChMjTmV0c2NhcnV0c29tbXVuaWNhdG1vb3JlbnQxMjE2LjE2LjE2LjE2
BgNVBAsTEU1zY290aW9ucyBDb3Jwb3JhdG1vb3JlbnQxMjE2LjE2LjE2LjE2LjE2
AOBiQPcK8851jjQXA2GBsaKNFg6pYaM3qhQhM0w5EiY6P1ttmJc5MlPIzZHdlgNd
QLzaNoLMVKjOV5sBp+ffkCAQMwDnnhup9mvbhgh
-----END CERTIFICATE-----
```

6. Open a new web browser window.
7. Go to the Administration Server, and use the Server Selector to access the Server Manager for your server.
8. Follow the directions presented there to install the certificate chain.

In general, you will be required to specify or enter the following information:

- o Whether the certificate is for this server or a trusted CA. Be sure to select the option that says the certificate is for a trusted certificate authority (CA).
- o A name (or nickname) for the certificate chain. This name will be displayed in the list of certificates installed on this server.
- o The certificate chain, in PKCS #7 format. In the original browser window (the window displaying the encoded certificate chain), copy the portion that begins with -----BEGIN CERTIFICATE----- and ends with -----END CERTIFICATE-----, and paste it into the message text area in the form.

9. Save your changes.
10. Stop and restart your Administration Server.

Step 5. Verify Your Server's SSL and CA Certificates

Before activating your server for SSL connections, you can verify whether you have installed your server's SSL and CA certificates correctly.

1. Open a web browser window.
2. Go to the Administration Server, and use the Server Selector to access the Server Manager for your server.
3. Follow the directions there to get to the area that allows you to manage your server's certificates.
4. Scroll to the bottom of the list to find the SSL and CA certificate chain you installed (identified by the nicknames you specified).

If you find both of them, your server is ready for SSL configuration. If not, you must go through the steps again to correctly install whichever certificate is missing.

Getting Certificates for iPlanet Servers

For iPlanet servers, you can use the Certificate Setup Wizard provided by iPlanet Console to get new certificates, renew existing certificates, and install certificates in the database of a server. For information about this wizard, see *Managing Servers with iPlanet Console*. To locate an online version of this book, open the `<server_root>/manual/index.html` file.

Note that there are two ways in which you can submit the certificate signing request to Certificate Management System:

- Submit the request (which is in the form of a base-64 encoded blob) directly from the wizard; in this method, you need not copy the request to a text file.
- Submit the request manually by pasting the request (which is in the form of a base-64 encoded blob) in to the Certificate Manager's server enrollment form; in this method, you need to copy the request when the wizard displays it.

When the wizard generates the certificate signing request for the key size and type you specified, you're presented with the opportunity to choose how you want to submit the request to the CA. The choices include the following:

To CA's email address. This option allows you to send the CSR to the CA administrator's email address. The administrator will then be required to submit the request to the CA by pasting the CSR in the CA's server enrollment form.

To CA's URL. This option allows you to submit the CSR to the CA directly. To submit the CSR to the Certificate Manager, you should enter, depending on the end-entity port you want to use, either of the following URL:

```
http://<CA's_hostname>:<end_entity_port>/enrollment or  
https://<CA's_hostname>:<end_entity_SSL_port>/enrollment
```

Note that the request submitted to the CA's URL gets queued for approval by the Certificate Manager agent.

To submit the server certificate request to Certificate Management System manually:

1. Open a web browser window.
2. Go to the End Entity Services interface of the Certificate Manager (or a Registration Manager that's connected to the Certificate Manager) by entering either of these URLs:

```
https://<hostname>:<end_entity_HTTPS_port> or  
http://<hostname>:<end_entity_HTTP_port>
```

3. In the left frame, under Server, select SSL Server.
4. In the server-enrollment form that appears, enter the required information:

PKCS#10 Request. Paste the base-64 encoded blob, including the -----BEGIN NEW CERTIFICATE REQUEST----- and -----END NEW CERTIFICATE REQUEST----- marker lines, you copied to the text file earlier.

Name. Type your name.

Email. Type your business email address, for example, `jdoe@someCompany.com`.

Phone. Type your business phone number.

Additional Comments. Type any information that will help you identify this request in the future or will help the person who will process this request.

5. Click Submit.

Renewal of Server Certificates

Every certificate issued by Certificate Management System has a validity period that determines its expiration date. The validity period of a certificate is determined by the *validity constraints* policy settings at the time the certificate was issued (see section “ValidityConstraints Plug-in Module” in *CMS Plug-Ins Guide*). For a certificate to be valid beyond its expiration date, it must be renewed. Otherwise, the certificate becomes invalid, and the entity owning the certificate will no longer be able to use it. Also, the expired certificate will take up space in your publishing directory and in the internal database of Certificate Management System.

Note that the Job scheduler component of Certificate Management System enables you to schedule a *job* for removing expired certificates from the publishing directory. For details, see “Configuring a Subsystem to Run Automated Jobs” on page 589.

Certificate Management System allows server administrators to renew their certificates by using the server enrollment form hosted by a Certificate Manager or Registration Manager. The renewal process is similar to the enrollment process in that the administrators must manually generate the certificate-signing request using the server’s key pair, paste that request in the manual enrollment form, and submit the request. For details, see “Certificate Issuance to Servers” on page 821.

For renewing the certificates of a Certificate Manager, Registration Manager, or Data Recovery Manager, see “Renewing Certificates for the Subsystems” on page 515.

Revocation of Server Certificates

Certificate Management System allows a certificate to be revoked by an end user (the original owner of the certificate), a server administrator, or by a Certificate Manager or Registration Manager agent. End users can revoke certificates by using the Revocation form provided in the end-entity services interface. Agents can revoke end-entity certificates by using the appropriate form in the Agent Services interface. Certificate-based (SSL client authentication) or challenge-password-based authentication is required in both cases; for details, see “Authentication of End Users During Certificate Revocation” on page 540.

- An end user can revoke only those certificates that contain the same subject name as in the certificate presented for authentication; if using a challenge password, the user can revoke only the certificate that is associated with that password. After successful authentication, the server lists the certificates

belonging to the end user. The end user can then select the certificate to be revoked or can revoke all certificates in the list. The end user can also specify additional details, such as the date of revocation and revocation reason for each certificate or for the list as a whole. For instructions on how end users revoke their certificates, see the online help available by clicking the Help buttons on the end-entity forms.

- Agents can revoke certificates based on a range of serial numbers or based on one or more subject name components. Upon submission of the revocation request, the agent receives a list of certificates from which she or he can pick the ones to be revoked. For instructions on how agents revoke end-entity certificates, see *CMS Agent's Guide*.

Upon receiving the list of certificates to be revoked, the Registration Manager formulates a CMMF request and sends it to the Certificate Manager. The Certificate Manager marks the corresponding certificate records in its certificate store (maintained in the internal database) as *revoked* and if configured to do so, removes the revoked certificates from the publishing directory and updates the CRL in the publishing directory.

Setting Up CEP Enrollment

iPlanet Certificate Management Server (CMS) can issue certificates to a wide variety of entities, such as web browsers, SSL-enabled servers, routers, virtual private network (VPN) clients, and so on. This chapter explains how you can configure Certificate Management System to issue router and VPN-client certificates.

The chapter has the following sections:

- CEP Enrollment (page 833)
- CEP Enrollment Using the Script (page 834)
- Setting up CEP Enrollment Manually (page 835)
- Certificate Issuance to Routers or VPN Clients (page 845)

CEP Enrollment

Cisco routers support the use of certificates for authentication, encryption, and tamper detection by using the IP Security (IPSec) protocol. Certificate Management System supports Cisco's PKI protocol, the Certificate Enrollment Protocol (CEP); this protocol runs over HTTP and provides its own form of encryption. For an overview of certificate authority support for IPSec, see the information available at this URL: http://www.cisco.com/warp/public/cc/cisco/mkt/security/encryp/prodlit/821_pp.htm

You can issue certificates to routers and CEP-compliant Virtual Private Network (VPN) clients using Certificate Management System. Routers use certificates to authenticate each other and to establish an encrypted IPSec channel between them; all TCP/IP communication passes through this encrypted channel.

Note that Certificate Management System by default supports issuance of certificates to routers and VPN clients using the CEP-based enrollment. However, publishing of these certificates to an LDAP-compliant directory is not turned on by default because routers and VPN clients need to have access to an LDAP directory in order to fully support various functions, such as certificate and CRL retrieval. This section explains how to set up a Certificate Manager to issue certificates to routers and CEP-compliant Virtual Private Network (VPN) clients. The section also describes how to configure the Certificate Manager to publish these certificates and certificate revocation lists (CRLs) to an LDAP-compliant directory.

You may configure the Certificate Manager to publish to any LDAP-compliant directory, but if you do not have one available, you can use the one supplied with Certificate Management System. Certificate Management System comes with iPlanet Directory Server, which is an LDAP-compliant directory. When you install Certificate Management System, two instances of Directory Server are automatically created in the same server group in which Certificate Management System is installed—one of the Directory Server instances is identified as the *configuration directory* and the other *internal database*. For publishing certificates and CRLs you may use the configuration directory, but not the internal database. The internal database is configured for exclusive use by Certificate Management System; see , “Setting Up Internal Database.”

There are two ways to set up CEP enrollment:

- CEP Enrollment Using the Script
- Setting up CEP Enrollment Manually

The sections that follow explain both ways of CEP enrollment in detail. The recommended is to use the interactive script.

CEP Enrollment Using the Script

Certificate Management System provides a menu-driven, interactive script to automate the CEP enrollment process. To invoke the script:

1. Go to the Certificate Manager’s host system.
2. Open a command-line window.
3. Go to this directory: <server_root>
4. Enter either the following, depending on your system, at the prompt:

```
% install/perl bin/cert/tools/cepconfig.pl on UNIX
% install\perl bin\cert\tools\cepconfig.pl on Windows NT
```

The main menu shows up.

```
CEPCONFIG
```

```
This script can be used to configure any instance of CMS.
Configuration tasks include the following:
```

```
- Adding/removing CEP services
```

```
You can configure different services, responding to different URLs
in CMS. This enables different authentication and policy options to
be set for different types of client (e.g. router and VPN)
```

```
- Enabling/disabling LDAP publishing of certificates and CRLs
```

```
Quick-and easy set up the publishing directory.
Turns on publishing.
Adds CRL distribution point to certificates.
```

```
Press ENTER to continue....
```

5. Follow the on-screen instructions to set up CEP enrollment.

Setting up CEP Enrollment Manually

The information covered in this section explains how to set up CEP enrollment manually. Note that the instructions are written with these assumptions:

- That you will publish certificates and CRLs to the configuration directory. For more information about the configuration directory, see *Managing Servers with iPlanet Console*. To locate this document, open the `<server_root>/manual/index.html` file.
- That you will publish certificates and CRLs to the same tree in the configuration directory; you may customize this if you desire. We recommend that you publish to a tree named after the `o` attribute in your CA signing certificate. Router certificates will also need to have an `o` inserted in the subject name; this can be done automatically. This section refers to the name of this tree as `Base DN`.

If you want to publish to any other LDAP-compliant directory, read Chapter 19, “Setting Up LDAP Publishing.”

To set up CEP enrollment manually, follow these steps:

- Step 1. Set up the Directory for Publishing Certificates and CRLs
- Step 2. Configure the Certificate Manager for Publishing Certificates and CRLs
- Step 3. Set Up Automated Enrollment (optional)
- Step 4. Set Up Multiple CEP Services (optional)

Step 1. Set up the Directory for Publishing Certificates and CRLs

Chapter 19, “Setting Up LDAP Publishing contains information on setting up Directory Server for publishing certificates and CRLs—it covers directory schema required for publishing certificates and the attributes to which a Certificate Manager publishes end-entity certificates and CRLs.

For the configuration directory to support publishing of certificates and CRLs, you need to verify two things:

- The Directory Server schema—verify that the directory schema can accommodate router and VPN client certificates. You may need to update the Directory Server’s schema. The reason for this is, if you plan on publishing certificates from routers, they may need to be published with the same DN as their certificate subject names. For example, if the certificate subject name contains `UnstructuredAddress` or `UnstructuredName` components, you may need to add them to the directory schema.

```
unstructuredAddress, 1.2.840.113549.1.9.7, string
unstructuredName, 1.2.840.113549.1.9.8, string
```

To modify the schema you can use the Directory Server window, which can be launched from within iPlanet Console. Alternatively, you can prepare an LDIF file with the changes you want to make and then run the LDAP *modify* command. Check the directory documentation for instructions.

- The Directory Server port—note the port number assigned to the configuration directory; it must be 389. If you installed Certificate Management System with the default choices, you may skip this step; the default port assigned to the configuration directory is 389. To find out the port number assigned to Directory Server, check its configuration file (which is at `<server_root>/slapd-*/slapd.oc.conf`). Alternatively, you can also find and change the port number from iPlanet Console.

Step 2. Configure the Certificate Manager for Publishing Certificates and CRLs

In this step, you configure the Certificate Manager to issue router and VPN-client certificates with *CRL Distribution Point Extension* and to publish the certificates to a directory.

- Create an instance of the mapper plug-in named `LdapExactMapper` and of the publisher plug-in named `LdapUserCertPublisher`. Once you create these instances, you should create a publishing rule for publishing router certificates. For instructions, see “Step B. Add Mappers, Publishers, and Publishing Rules” on page 666.

Note that the publishing rule must be configured to use the mapper and publisher you create for router certificates. In addition, the predicate expression must be set to `HTTP_PARAMS.certType==CEP-Request`.

- Configure CRL publishing details; for instructions, see “Step 4. Configure the Certificate Manager to Publish CRLs” on page 672.
- Identify the directory for publishing. For instructions, see “Step 5. Identify the Publishing Directory” on page 680.
- Create an instance of the policy plug-in named `CRLDistributionPointsExt` (following the instructions in “Step 4. Add New Policy Rules” on page 618) for router certificates. This extension, if present in a certificate, enables the user of the certificate to find revocation information pertaining to that certificate. When you create an instance of the `CRLDistributionPointsExt` plugin, be sure to leave the `issuerName` and `issuerType` fields blank and to enter `HTTP_PARAMS.certType==CEP-Request` in the predicate field.

- Stop the Certificate Manager and edit the configuration file to include the following lines:

```
eeGateway.cep.cep1.appendDN=O=<BASE DN>
eeGateway.cep.cep1.createEntry=true
eeGateway.cep.cep1.entryObjectClass=cep
eeGateway.cep.cep1.url=/cgi-bin/pkiclient.exe
```

A description for each of the above parameters are provided in Table 25-1.

Table 25-1 CEP service-related configuration parameters in the configuration file

Parameter	Description
appendDN	Specifies the DN component appended to the DN the router requests. You must have a constant component in the DN which exists in the certificate to be able to publish.
createEntry	Specifies whether to create an entry in the directory before publishing the certificate. Note that to publish a certificate, an entry must already exist for the DN in the directory. <ul style="list-style-type: none"> • Enter <code>true</code> if you want the Certificate Manager to create an entry if one does not already exist (<code>true/false</code>). • Enter <code>false</code> if an entry already exists in the directory and you don't want the server to create one.
url	Specifies the URL for CEP enrollment. It is used if the router requests a subject name such as <code>unstructuredAddress=1.2.3.4+unstructuredName=fred.siroe.com</code> . You will need to append the DN to add-on <code>O=siroe.com</code> as otherwise publishing to the directory will not work.

Table 25-1 CEP service-related configuration parameters in the configuration file *(Continued)*

Parameter	Description
entryObject Class	<p>Specifies the type of object to assign to the new entry. By default, this is <code>cep</code>, and should not be changed. Note that when <code>createEntry=true</code>, the Certificate Manager will attempt to create an entry for the user. The directory hierarchy must be set up correctly beforehand to accept new entries. To clarify this by an example, if you expect the Certificate Manager to be able to create an entry for a certificate with DN <code>CN=John Doe, OU=Accounting, O=Company, C=US</code>, you must have already created three (3) directory entries for</p> <pre>C=US O=Company, C=US OU=Accounting, O=Company, C=US</pre> <p>You can do this with the help of the <code>ldapmodify</code> command and an LDIF file with the following information:</p> <pre>dn: C=US changetype: add objectclass: top objectclass: country c: US dn: O=Company, C=US changetype: add objectclass: top objectclass: organization o: Company dn: OU=Accounting, O=Company, C=US changetype: add objectclass: top objectclass: organizationalunit ou: Accounting dn: cn=config,cn=ldbm changetype: modify add: nsslapd-suffix nsslapd-suffix: C=US</pre> <p>Also, note that in this case, <code>C=US</code> is the root of your directory, because it is the sole component of the DN. To create a new root, you must first add a new suffix, which is done by adding this to your LDIF file (see the last section of the LDIF file above). For more details, check <i>CMS Agent's Guide</i>.</p>

Once this is done, you may configure the Certificate Manager for automated enrollment. To do so, follow the steps in the next section. If you do not want to configure the server for automated enrollment, restart the server.

Step 3. Set Up Automated Enrollment

As a part of enrolling for a certificate (via CEP), a router administrator or VPN-client user needs to start the enrollment process, which in turn asks the user for information such as the following:

- The CA's identity
- The CEP enrollment URL
- A challenge password
- The serial number and IP address

Some of the information a user enters, such as the serial number and IP address, goes in to the subject name in the CEP request. Information such as the CA's identity and enrollment URL enables the router to connect to the valid CA to make the certificate request. The challenge password, if specified, enables the user to authenticate to the server during enrollment and to revoke the certificate, if needed, by presenting the same password again. (See "Certificate Issuance to Routers or VPN Clients" on page 845.)

You can configure the Certificate Manager to use either the challenge password or the subject name (all or a part of it) as an authentication token during a CEP enrollment, thus enabling users to get router certificates without any action on the part of the Certificate Manager agent.

To aid you in implementing the automated CEP enrollment process, Certificate Management System comes with an authentication plug-in module named `FlatFileAuth`. This plug-in is available in source-code form in the CMS samples package in this directory:

```
<server_root>/cms_sdk/cms_jdk/samples/authentication
```

In order for the Certificate Manager to recognize the `FlatFileAuth` plug-in and use it for authenticating CEP-based certificate requests, you must do the following:

- Register the plugin in the CMS authentication framework; for instructions, see "Registering an Authentication Module".
- Create an instance of the plug-in; for instructions, see "Step 4: Add an Authentication Instance" on page 553.

You can do this either via the CMS window or by adding the required parameters to the Certificate Manager's configuration file (`CMS.cfg`). The configuration parameters of the `FlatFileAuth` plug-in are listed below.

```
eeGateway.cep.cepl.authName=flatfile
auths.instance.flatfile.fileName=<full_pathname_of_password_file>
auths.instance.flatfile.authAttributes=pwd
auths.instance.flatfile.keyAttributes=UNSTRUCTUREDNAME
auths.instance.flatfile.pluginName=flatfilePlugin
auths.instance.flatfile.deferOnFailure=false

auths.impl.flatfilePlugin.class=com.netscape.certsrv.authentication
.FlatFileAuth
```

A description for each of the above listed parameters are provided in Table 25-2.

Table 25-2 Configuration parameters defined in the `FlatFileAuth` plug-in

Configuration parameter	Description
<code>authName</code>	Provides a reference to the <code>auths.instance</code> authentication plug-in described in the <code>auths.instance.*</code> configuration parameters. If you want to turn off automated enrollment for CEP-based requests, delete this parameter from the configuration file.
<code>fileName</code>	Specifies the filename of an authentication-token file. You prepare this file as a part of setting up an automated CEP enrollment as explained in Step 4-B. Be sure to use the full path name.
<code>keyAttributes</code>	Specifies a comma-separated list of attributes in the request which together, uniquely identify an entry in the authentication-token file. Note that these attributes must be present in the request and in the password file for the authentication to succeed.
<code>authAttributes</code>	Specifies a comma-separated list of attributes from the CEP request which must match the attributes specified in the authentication-token file for authentication to succeed. Currently the most useful thing to put in this parameter is <code>pwd</code> , the challenge password from the request.
<code>deferOnFailure</code>	Specifies whether the server should defer CEP requests that fail authentication. <ul style="list-style-type: none"> <code>true</code> specifies that the server should defer CEP-enrollment requests that fail authentication; the deferred requests get queued for agent approval. <code>false</code> specifies that the server should reject CEP-enrollment requests that fail authentication.

During CEP enrollment, all the attributes in the subject name and the challenge password are passed to the `FlatFileAuth` plug-in. The plug-in looks in a prepared file (referred to as the authentication-token file in this document), which consists of a series of entries for each valid enrollee, to determine if the request should be authenticated. For the Certificate Manager to be able to locate the appropriate entry in the authentication-token file before it does any checking of the password, you must identify attributes that are unique in each router request. You do this by setting the `keyAttributes` parameter of the `FlatFileAuth` plug-in implementation to the list of attributes which will be unique in the CEP request.

Table 25-3 lists the default attributes set by the router in the request.

Table 25-3 Default request attributes set by the router

Attribute	Description
UNSTRUCTUREDNAME	Specifies the DNS name of the router (for example, <code>router32.siroe.com</code>). This is always specified in the request.
UNSTRUCTUREDADDRESS	Specifies the IP address of the router (for example, <code>101.22.33.124</code>). This may not be in the request—a user may not want to include this in the subject name of the router certificate, and hence choose not to specify one during enrollment.
SERIALNUMBER	Specifies the serial number of the router (for example, <code>239333</code>). This can sometimes be found on a label on the back of the router. It is also available by typing the <code>show version</code> command. This may not be in the request—a user may not want to include this in the subject name of the router certificate, and hence choose not to specify one during enrollment.

You can identify one or more of these attributes as unique attributes in the authentication-token file. For most cases, specifying the `UNSTRUCTUREDNAME` as the unique attribute will suffice. In this case, you would set the `keyAttributes` parameter as follows:

```
auths.instance.flatfile.keyAttributes=UNSTRUCTUREDNAME
```

However, if this is not unique, you may specify both `UNSTRUCTUREDNAME` and `UNSTRUCTUREDADDRESS` as unique attributes. In this case, you would set the `keyAttributes` parameter as follows:

```
auths.instance.flatfile.keyAttributes=UNSTRUCTUREDNAME,UNSTRUCTURED  
ADDRESS
```

This will force the server to use both these attributes to locate an entry in the authentication-token file. Note that both the attributes must be present in the request for authentication to succeed.

There's an added advantage in determining unique attributes for it allows you to enforce a rule on the attributes that must be present in the CEP enrollment request. For example, if you would like to enforce that a particular router be assigned to an IP address and host name, you could set the `keyAttributes` parameter as follows:

```
auths.instance.flatfile.keyAttributes=UNSTRUCTUREDNAME,UNSTRUCTURED
ADDRESS,SERIALNUMBER
```

Once an entry has been found in the authentication-token file, the server tests the authentication tokens specified in the `authAttributes` parameter against those in the file. Only if they all match, the server grants the request. For the purposes of this discussion, let us assume that you define a single authentication token named `pwd` for the challenge password. In this case, you would set the `authAttributes` parameter as follows:

```
auths.instance.flatfile.authAttributes=pwd
```

In summary, to implement the automated CEP enrollment process, you need to do the following:

1. Decide on authentication credentials for users.

Prepare a list of your CEP enrollees and assign a password to each enrollee.

2. Prepare the authentication-token file with the credentials.

Create a text file with CEP-enrollee information. The format of the authentication-token file must be as follows:

```
<attribute>: <value>
<attribute>: <value>
...
<attribute>: <value>
<attribute>: <value>
```

Each enrolling user is represented by a sequence of attribute-value pairs, terminated by a blank line or end-of-file (EOF). The attributes can be any part of the subject name from the request, for example `SERIALNUMBER`, `CN`, `OU`, `UID`, or the challenge password (`pwd`). An example is shown below:

```
DN: <DN_for_user1>
UNSTRUCTUREDNAME: router32.siroe.com
UNSTRUCTUREDADDRESS: 101.22.33.124
SERIALNUMBER: 239333
pwd: ff93Kd
```

```
DN: <DN_for_user1>
UNSTRUCTUREDNAME: router33.siroe.com
UNSTRUCTUREDADDRESS: 101.22.33.125
SERIALNUMBER: 233455
pwd: 35pww3a
```

Note that if you specify a DN for a CEP enrollee in the authentication file, the Certificate Manager replaces the subject name requested by that user (router or VPN client) with the one specified in the file.

After you've created entries for all the CEP enrollees, save the file. Also note the complete path to the file; you'll be required to specify this in the next step.

3. Configure the Certificate Manager to use the `FlatFileAuth` plug-in to verify predetermined credentials.

Once you have created the authentication file, you should register the `FlatFileAuth` plug-in implementation in the CMS authentication framework and create an instance (named `flatfile`) of the registered plug-in. Be sure to specify the full path to your authentication file and save your changes.

4. Restart the Certificate Manager.

After changing the configuration file, you must restart the server for the changes to take effect. If the server fails to start, check the log file for any error messages.

5. Provide users with the predetermined authentication credentials.

Send the password you determined to the CEP enrollees, asking them to enter it when they are prompted for the challenge password during CEP enrollment.

Step 4. Set Up Multiple CEP Services

This step is optional.

By default, the CEP service runs on this URL: `/cgi-bin/pkiclient.exe`

It is possible to set up multiple instances of CEP, each with a different configuration, each listening on a different URL. This is useful if you have different requirements for different types of users. For example, you might want to have one CEP service that authenticates routers and publishes their certificates to the directory and another CEP service that authenticates VPN clients but does not publish their certificates to the directory.

To set up multiple CEP services, use the example below as a guide.

```

## Router configuration
eeGateway.cep.cep1.appendDN=O=*BASE_DN*
eeGateway.cep.cep1.createEntry=true
eeGateway.cep.cep1.entryObjectClass=cep
eeGateway.cep.cep1.url=/cgi-bin/pkiclient.exe
eeGateway.cep.cep1.authName=flatfile_router

## VPN configuration
eeGateway.cep.cep2.url=/vpnenroll
eeGateway.cep.cep2.authName=flatfile_VPN

## Router authentication parameters in the configuration file
auths.instance.flatfile_router.fileName=
  <full_path_to_the_authentication_file>
auths.instance.flatfile_router.authAttributes=pwd
auths.instance.flatfile_router.keyAttributes=UNSTRUCTUREDNAME
auths.instance.flatfile_router.pluginName=flatfile
auths.instance.flatfile_router.deferOnFailure=true

## VPN authentication parameters in the configuration file
auths.instance.flatfile_VPN.fileName=
  <full_path_to_the_authentication_file>
auths.instance.flatfile_VPN.authAttributes=pwd
auths.instance.flatfile_VPN.keyAttributes=CN,OU,O
auths.instance.flatfile_VPN.pluginName=flatfile
auths.instance.flatfile_VPN.deferOnFailure=false

## FlatFileAuth plugin registered in the configuration file
auths.impl.flatfile.class=com.netscape.certsrv.authentication.
FlatFileAuth

```

When setting up multiple CEP services, you can use the `cepstore` attribute to differentiate one CEP service from another. For example, if you're setting up separate CEP services for router and VPN-client certificates and want to set different extensions in these certificates, you can make that happen with the help of predicates; see Table 18-2 on page 608.

Certificate Issuance to Routers or VPN Clients

In general, issuing a certificate to a router involves the following steps:

- Step 1. Before You Begin
- Step 2. Generate the Key Pair for the Router
- Step 3. Request the CA's Certificate
- Step 4. Submit the Certificate Request to the CA

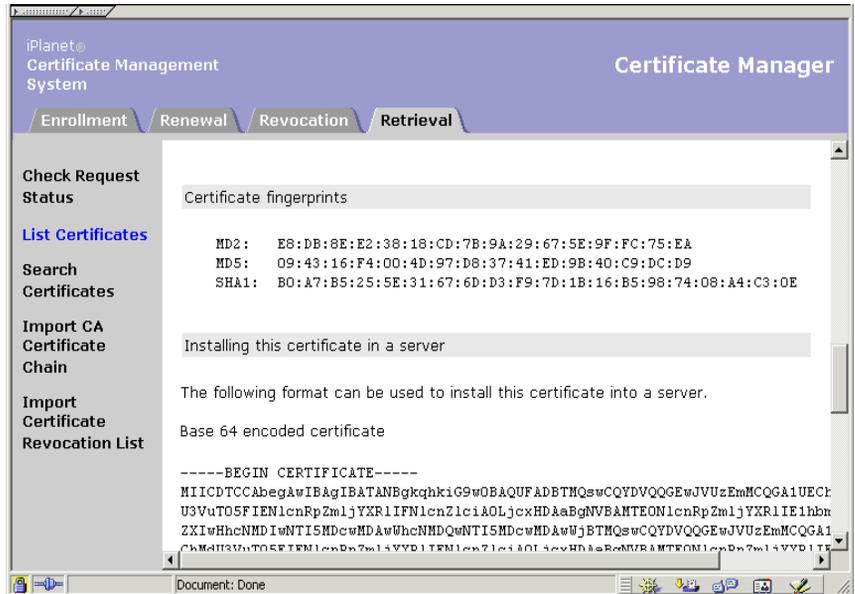
Step 1. Before You Begin

- Decide whether you want to submit the certificate request for your router to the Certificate Manager (CA) directly or through a Registration Manager.
- Open iPlanet Console, and locate the CMS instance that corresponds to the subsystem of your interest. Make sure that the Certificate Manager is started. If you are planning to submit the request via a Registration Manager, make sure that both the Registration Manager and Certificate Manager (to which the Registration Manager forwards its requests) are started.
- Open the CMS window, and verify whether the HTTP port is enabled. If it isn't, enable it; for instructions, see "Configuring Port Numbers" on page 384. If you are requesting the certificate for an earlier version of router software, make sure that the HTTP port number is set to 80; earlier versions of router software can only connect to port 80.
- Note the CEP enrollment URL. The URL is in the form `http://<hostname>:<HTTP_port>/cgi-bin/pkiclient.exe`, where `<hostname>` is in the `<machine_name>.<your_domain>.<domain>` form.
- Note or print the certificate fingerprint information of the Certificate Manager *CA signing certificate*. You will be required to compare this with the fingerprint the router will show on the screen.

To locate the fingerprint information:

- a. Go to the end-entity page hosted by the Certificate Manager.
- b. Click the Retrieval tab.
- c. List or search for the CA signing certificate.
- d. Click Details.

- e. Scroll down to the section that says “Certificate fingerprint.”



- In your router documentation, locate the information specific to requesting certificates for routers. Check the signing algorithm, such as RSA or DSA, and key lengths, such as 512 and 1024, supported by the router. Based on that information, determine the signing algorithm and the key length for the certificate you want to request.
- Find out the password that enables you to access the router in *privileged* mode.
- In your router documentation, locate instructions for requesting certificates. You will be required to run the appropriate commands using this documentation.

Step 2. Generate the Key Pair for the Router

Run the appropriate commands for your router, and generate the key pair. You will be required to provide the signing algorithm, such as RSA or DSA, and the key length, such as 512 or 1024. The longer the key length, the more time the router takes to generate the key pair.

Step 3. Request the CA's Certificate

In this part of the operation, you identify the CA to the router, thus enabling the router to authenticate the CA from which it will request the certificate. You also verify whether the router is talking to the right CA; you do this manually.

Here's what you should do:

1. Run the appropriate command to get the CA certificate.

The command will ask you to specify the following:

- An identity for the CA. You can give any identity; choose something you will remember, since you will be required to provide it when you submit the certificate request.
 - The CA's enrollment URL; this is the enrollment URL you identified in Step 1.
2. The router gets the CA certificate and displays its fingerprint on your screen.
 3. Verify the fingerprint on your screen with the one you noted down in Step 1.

If it matches, the router is talking to the right CA.

Step 4. Submit the Certificate Request to the CA

To submit the certificate request to the CA:

1. Run the appropriate command.

The command will ask you for certain information:

- The CA's identity. You specified this in Step 3.
- Challenge password. If you enter one, write it down; you will be required to specify this password to revoke the certificate.
- The CEP enrollment URL.
- Whether you want to include the router's serial number in the request. If you choose to include the serial number, it will be included in the certificate's subject name.
- Whether you want to include the router's IP address in the request. If you choose to include the IP address, it will be included in the certificate's subject name.

2. This step depends on your CA's configuration for router enrollment.
 - o If the CA to which the router submitted the request employs automatic enrollment (or authentication) for routers, the request will get processed by the CA. The CA may return the certificate to the router in the same transaction. If it doesn't, the router checks with the CA at periodic intervals; in the router configuration you can specify how often the router should poll the CA for the certificate and how many attempts it should make. By default, the router checks the CA every minute.
 - o If the CA to which you submitted the request is configured for manual enrollment (or authentication), the request gets queued and awaits approval by an agent.

NOTE Your router may require additional configuration changes. Be sure to follow the information in your router documentation.

Example

The example below shows the commands and associated outputs for a Cisco router:

```
# To perform certificate enrollment for a router using CEP, you must
be
# in privileged mode, which you do by typing "enable" first, and then
# entering the password.

router> enable
router% config terminal

router(config)#crypto key generate rsa

The name for the keys will be: netscape.mcom.com

Choose the size of the key modulus in the range of 360 to 2048
for your
General Purpose Keys. Choosing a key modulus greater than 512 may
take a few
minutes.

How many bits in the modulus [512]:
Generating RSA keys ...
[OK]

router(config)#crypto ca identity test-ca
```

```
router(ca-identity)#enrollment url
http://ca-hostname.domain.com/cgi-bin/
                                pkiclient.exe

router(ca-identity)#exit

router(config)#crypto ca authenticate test-ca
Certificate has the following attributes:
Fingerprint: 24D34656 EB830C39 DD9E8179 0A4EBA98
% Do you accept this certificate? [yes/no]: yes

router(config)#crypto ca enroll test-ca
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide
this
password to the CA Administrator in order to revoke your
certificate. For
security reasons your password will not be saved in the
configuration.
Please make a note of it.

Password:
Re-enter password:

% The subject name in the certificate will be: router.domain.com
% Include the router serial number in the subject name? [yes/no]:
yes
% The serial number in the certificate will be: 08342063
% Include an IP address in the subject name? [yes/no]: yes

Interface: ethernet0

Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the
fingerprint.

router(config)# exit

router#show crypto ca certificates

CA Certificate
  Status: Available
  Certificate Serial Number: 1
  Key Usage: Not Set
```

Certificate

Subject Name

Name: netscape.mcom.com

IP Address: 208.12.63.193

Serial Number: 08342063

Status: Pending

Key Usage: General Purpose

Fingerprint: 91D70D7F D8BF0DFA E13F00B0 6EA706A0 00000000

Appendixes

Appendix A, “Certificate Download Specification”

Appendix B, “Using SSL with iPlanet Web Server, Enterprise Edition
4.x”

Appendix C, “Export Control Information”

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 855)
- Importing Certificate Chains (page 857)
- Importing Certificates into Netscape Communicator (page 857)
- Importing Certificates into iPlanet Servers (page 858)
- Object Identifiers (page 858)

Data Formats

iPlanet 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 \(page 857\)](#) 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 858](#)), 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 \(page 857\)](#) 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 iPlanet 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 iPlanet Web Server and configuring it to support certificate-based client authentication, see Appendix B, “Using SSL with iPlanet Web Server, Enterprise Edition 4.x.”

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 iPlanet Certificate Management Server) 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 862.

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 iPlanet Certificate Management Server has published to the LDAP directory.

This appendix has the following sections:

- Creating a New Server (page 862)
- Obtaining a Server Certificate (page 863)
- Enabling SSL on the Server (page 869)
- Testing Client Authentication (page 877)

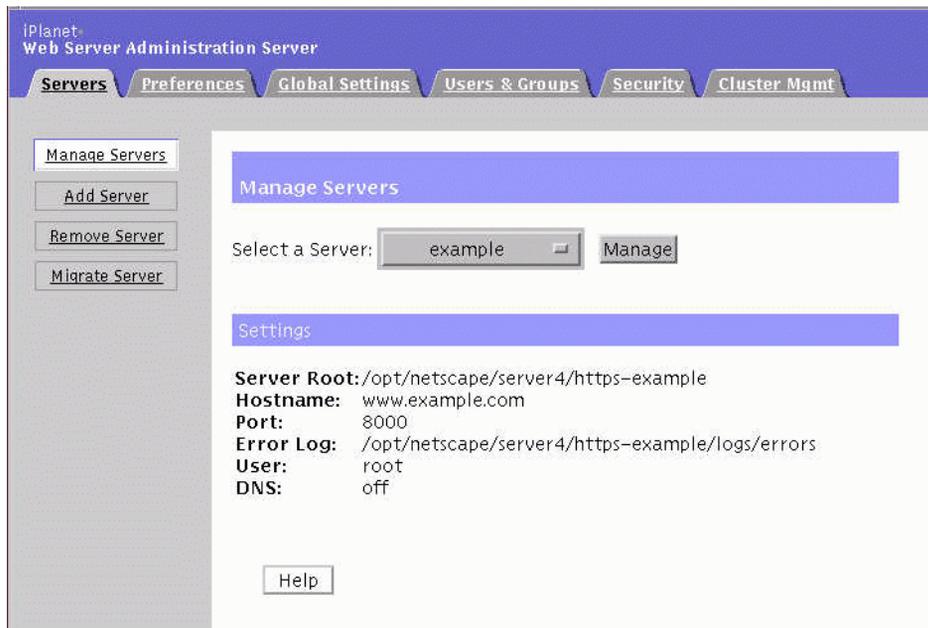
Creating a New Server

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 863. 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 869.

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
- Submitting a Certificate Signing Request
- Importing the Certificate

Creating a Trust Database

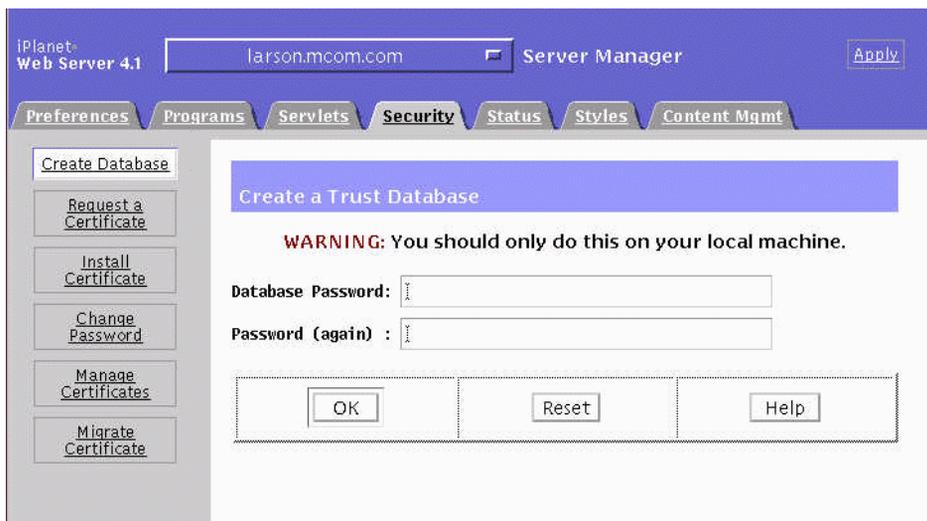
To support SSL, you first create a Trust Database that will contain all of the keys and certificates used by the server (including many pre-installed root certificates from public Certificate Authorities).

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.



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 iPlanet Certificate Management Server 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 laron.mcom.com Server Manager Apply

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: **internal (software)**

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://demoCA.siroe.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.)

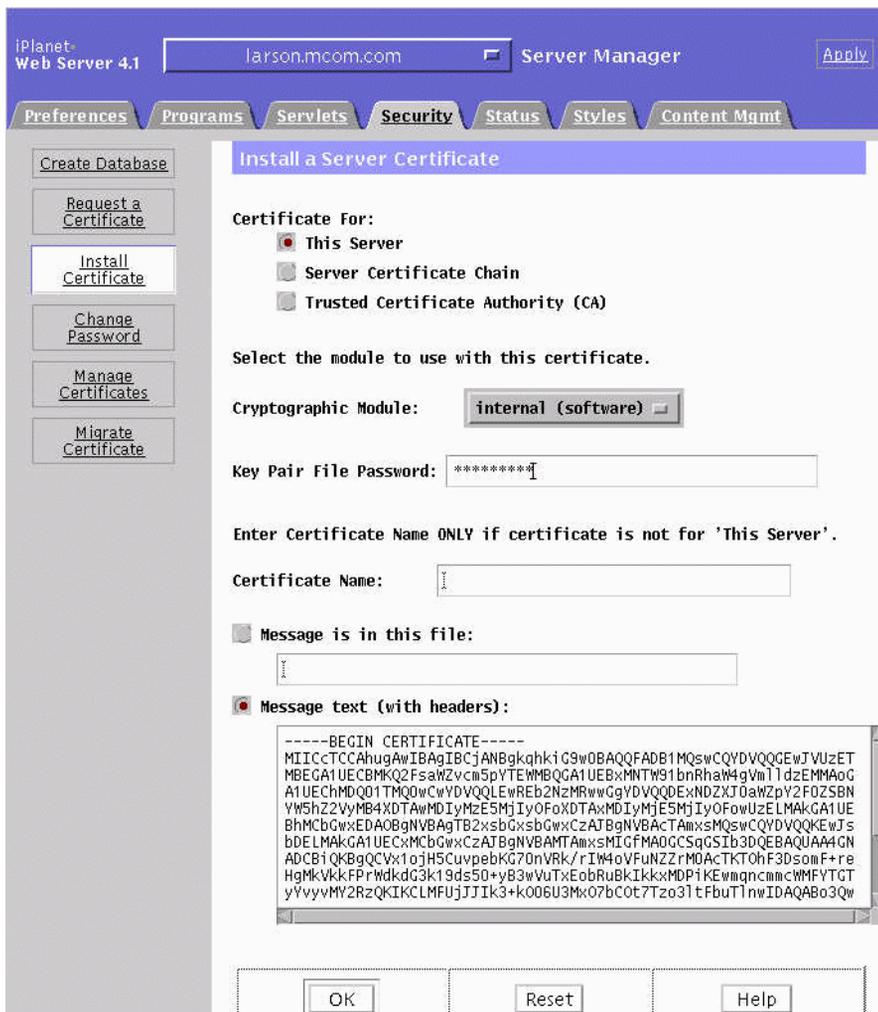
To import the server certificate into the server, follow these steps:

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-----  
MIICeTCCAeKgAwIBAgIChfQwDQYJKoZIhvcNAQEEBQAwdzELMAkGA1UEBhMCVVMx  
LDAqBgNVBAoTIO5ldHNjYXB1IENvbW11bmljYXRpb25zIENvcnBvcYXB1IENvbW1  
1bmljYXRpb25zIENvcnAuMSAwHgYDVQQDFBdtb2cuKG5ldHNjYXB1fG1jb20pLmN  
vbTCBnTANBgkqhkiG9w0BN0nZmUaB3adv7D1TPA==  
-----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.

This page appears.



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 869 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 869. 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 870.

To enable SSL client authentication on all requests, perform only the task in “Enabling Client Authentication for All Requests” on page 871.

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
- Modifying the Configuration File
- Modifying the Access Control Lists

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://demoCA.siroe.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 iPlanet Certificate Management Server 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' configuration page in the iPlanet Web Server Administration Server. The page is titled 'iPlanet Web Server Administration Server' and has several tabs: Servers, Preferences, Global Settings (selected), Users & Groups, Security, and Cluster Mgmt. On the left sidebar, there are buttons for 'Configure Directory Service', 'Restrict Access', 'Cron Control', and 'Configure JRE/JDK Paths'. The main content area is titled 'Configure Directory Service' and contains an 'LDAP Directory Server Configuration' section. The fields are as follows:

- Host Name: ldap.example.com
- Port: 389
- Use Secure Sockets Layer (SSL) for connections?: Yes No
- Base DN: o=example.com
- Bind DN: uid=admin,ou=Administrators,ou=TopologyManagement
- Bind Password: [masked]

At the bottom of the form are three buttons: '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 iPlanet Certificate Management Server 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 iPlanet 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=Sun Microsystems" 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.

The screenshot shows the iPlanet Web Server 4.1 Server Manager interface. The top navigation bar includes tabs for Preferences, Programs, Servlets, Security, Status, Styles, and Content Mgmt. The main content area is divided into two panes.

Access Control Rules for : default

Action	Users/Groups	From Host	Rights	Extra...	Continue
1 Allow	all	anyplace	r-x-li	x	■

■ Access control is on

Current Access deny response is the default file (redirection off) [Response when denied](#)

Buttons: Submit, Revert, Help

User/Group

Anyone (No Authentication)

Authenticated people only

All in the authentication database

Only the following people

Group : List

User : List

Prompt for authentication :

Authentication Methods :

Default Basic SSL Other

Authentication Database:

Default Other:

Default LDAP

Buttons: Update, Reset, Help

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 875.)
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.

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 iPlanet Certificate Management Server. It does not describe the global version of Certificate Management System.

In most cases, the full-strength encryption version (or global version) of Certificate Management System is exportable outside of the United States of America. Certificate Management System 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 Certificate Management System with the same encryption and cryptographic features available in the US and Canada.

The global version of Certificate Management System 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 Certificate Management System be made available to certain persons. For example, local laws may prohibit importing strong encryption, US law may change in the future, or Certificate Management System 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 880)
- SSL Cipher Suite Profiles for Export (page 882)

Approved Export Operations and Key Sizes

Table C-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 882.

Table C-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 iPlanet 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
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

Table C-1 Approved export operations and key lengths *(Continued)*

Description of cryptographic operation	Key length or algorithm strength
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
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

Table C-1 Approved export operations and key lengths *(Continued)*

Description of cryptographic operation	Key length or algorithm strength
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
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 C-2 summarizes the cipher suite profiles approved by the US government for use in the export-controlled version of Certificate Management System.

Table C-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	

Smart Card Login with Windows 2000

This appendix provides detailed instructions for configuring Certificate Server 4.7 to generate certificates that can be used for Smart Card login in a Windows 2000 environment.

Overview

Enabling this feature in Certificate Server 4.7 is a three-part process.

1. Set up the Windows 2000 environment to work with Certificate Server 4.7.
2. Configure Certificate Server 4.7.
3. Customize enrollment forms.

This document provides information resources for Part 1, detailed instructions for Part 2, and some general notes for Part 3.

Part 1. Set Up the Windows 2000 Environment

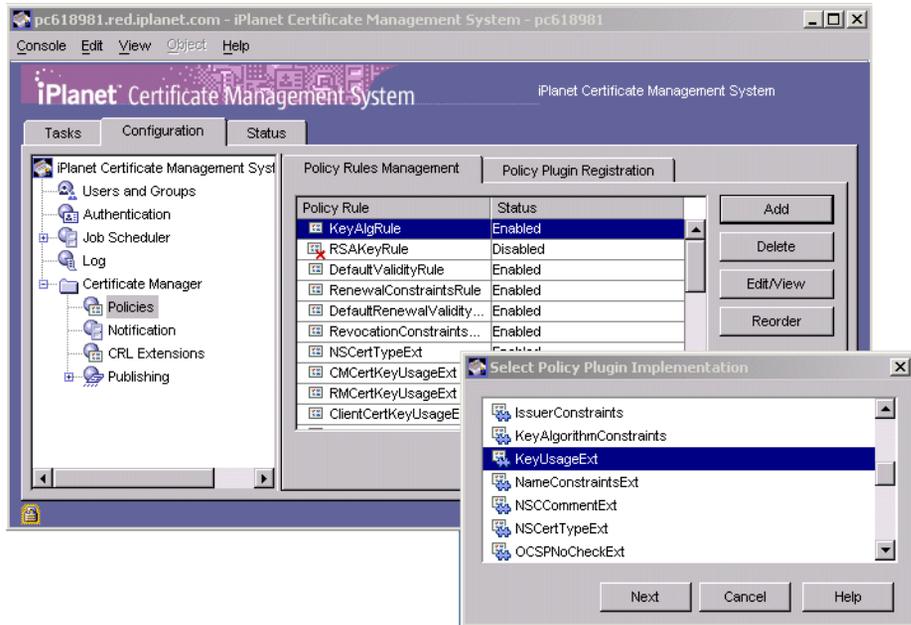
For detailed instructions, see the following documents on the Microsoft website:

- “Requirements for Domain Controller Certificates from a Third-Party CA (Q291010)” at <http://support.microsoft.com/support/KB/articles/Q291/0/10.ASP>
- “Guidelines for Enabling Smart Card Logon with Third-Party Certification Authorities (Q281245)” at <http://support.microsoft.com/support/KB/articles/Q281/2/45.ASP>

Part 2. Configuring Certificate Server 4.7

In this part, you configure a number of policies for the Smart Card Certificate.

1. Set up the Key Usage policy for the Smart Card certificat.
 - a. In the Certificate Server window, in the navigation tree, navigate to the Policy Rules Management tab by clicking Configuration>Policies>Certificate Manager.
 - b. In the Policies Management tab, click Add.
 - c. In the Select Policy Plug-in Implementation window, select KeyUsageExt, and then click Next.



- d. In the Policy Rule Editor window, set the Digital Signature key usage for Smart Card. Enter values as in the following figure, and then click OK:

The screenshot shows the Policy Rule Editor window for a rule named 'smartCardKey'. The window title is 'Policy Rule ID: smartCardKey' and the subtitle is 'Policy Plugin ID: KeyUsageExt'. The main configuration area contains several settings:

- enable**: checked
- predicate**: HTTP_PARAMS.certType == smartCard
- critical**: checked
- digitalSignature**: true
- nonRepudiation**: false
- keyEncipherment**: false
- dataEncipherment**: false
- keyAgreement**: false
- keyCertsign**: false
- criSign**: false
- encipherOnly**: false
- decipherOnly**: false

Below the configuration area, there is a text box containing the text: 'Adds Key Usage Extension; See in RFC 2459 (4.2.1.3)'. At the bottom of the window, there are three buttons: 'OK', 'Cancel', and 'Help'.

2. Set up the Extended Key Usage policy for the Smart Card Certificate.
 - a. In the Policies Management tab, click Add.
 - b. In the Select Policy Plug-in Implementation window, select ExtendedKeyUsage.
 - c. In the Policy Rule Editor window, Set extended key usages for Client Authentication (1.3.6.1.5.5.7.3.2) and Smart Card Logon (1.3.6.1.4.1.311.20.2.2). Enter values as in the following figure, and then click OK:

The screenshot shows a dialog box titled "Policy Rule Editor" for the rule "smartCardExtendedKey". The "Policy Plugin ID" is "ExtendedKeyUsageExt". The "enable" checkbox is checked. The "predicate" field contains the expression "HTTP_PARAMS.certType == smartCard". The "critical" checkbox is unchecked. The "numIds" field contains the value "2". The "id0" field contains the value "1.3.6.1.5.5.7.3.2". The "id1" field contains the value "1.3.6.1.4.1.311.20.2.2". At the bottom, there is a text box that reads "Adds Extended Key Usage Extension. Defined in RFC 2459 (4.2.1.13)". The dialog has "OK", "Cancel", and "Help" buttons.

Policy Rule ID: smartCardExtendedKey
Policy Plugin ID: ExtendedKeyUsageExt

enable

predicate

critical

numIds

id0

id1

Adds Extended Key Usage Extension. Defined in RFC 2459 (4.2.1.13)

OK Cancel Help

3. Set up the Subject Alt Name policy for the Smart Card certificate.
 - a. In the Policies Management tab, click Add.
 - b. In the Select Policy Plug-in Implementation window, select SubjectAltNameExt.
 - c. In the Policy Rule Editor window, Set the user principal name (UPN) for otherName. Enter values as in the following figure, and then click OK:

The screenshot shows the Policy Rule Editor window for the policy rule ID 'smartCardSubAltName' and policy plugin ID 'SubjectAltNameExt'. The window contains the following fields and controls:

- enable**:
- predicate**: HTTP_PARAMS.certType == smartCard
- critical**:
- numGeneralNames**: 1
- generalName0.requestAttr**: HTTP_PARAMS.UPN
- generalName0.generalNameChoice**: otherName (selected from a dropdown menu)

Below the fields, there is a text box with the following text:

This policy inserts the Subject Alternative Name Extension into the certificate. See RFC 2459 (4.2.1.7). * Note: you probably want to use this policy in conjunction with an authentication manager which sets the 'mail' or 'mailalternateaddress' values in the authToken. See the 'ldapStringAttrs' parameter in the Directory-based authentication plugin

At the bottom of the window are three buttons: OK, Cancel, and Help.

4. Set up the Key Usage policy for the domain controller certificate.
 - a. In the Policies Management tab, click Add.
 - b. In the Select Policy Plug-in Implementation window, select KeyUsageExt.
 - c. In the Policy Rule Editor window, set digitalSignature and keyEncipherment. Enter values as in the following figure, and then click OK:

Policy Rule ID: domainControllerKey
Policy Plugin ID: KeyUsageExt

enable	<input checked="" type="checkbox"/>
predicate	HTTP_PARAMS.certType == domainController
critical	<input checked="" type="checkbox"/>
digitalSignature	true
nonRepudiation	false
keyEncipherment	true
ataEncipherment	false
keyAgreement	false
keyCertsign	false
crISign	false
encipherOnly	false
decipherOnly	false

Adds Key Usage Extension; See in RFC 2459 (4.2.1.3)

OK Cancel Help

5. Set up the Extended Key Usage policy for the domain Controller Certificate.
 - a. In the Policies Management tab, click Add.
 - b. In the Select Policy Plug-in Implementation window, select ExtendedKeyUsageExt.
 - c. In the Policy Rule Editor window, enable Client Authentication and Server Authentication in extended key usage. Enter values as in the following figure, and then click OK:

The screenshot shows the Policy Rule Editor window for the ExtendedKeyUsageExt plugin. The window title is "Policy Rule ID: domainControllerExtendedKey" and "Policy Plugin ID: ExtendedKeyUsageExt". The "enable" checkbox is checked. The "predicate" field contains the text "HTTP_PARAMS.certType == domainController". The "critical" checkbox is unchecked. The "numIds" field contains the value "2". The "id0" field contains the value "1.3.6.1.5.5.7.3.2" and the "id1" field contains the value "1.3.6.1.5.5.7.3.1". A scroll bar is visible below the id fields. At the bottom of the window, there is a text box containing the description "Adds Extended Key Usage Extension. Defined in RFC 2459 (4.2.1.13)". At the very bottom, there are three buttons: "OK", "Cancel", and "Help".

Policy Rule ID: domainControllerExtendedKey
Policy Plugin ID: ExtendedKeyUsageExt

enable

predicate HTTP_PARAMS.certType == domainController

critical

numIds 2

id0 1.3.6.1.5.5.7.3.2

id1 1.3.6.1.5.5.7.3.1

Adds Extended Key Usage Extension. Defined in RFC 2459 (4.2.1.13)

OK Cancel Help

6. Set up the Subject Alt Name policy for the Domain Controller certificate.
 - a. In the Policies Management tab, click Add.
 - b. In the Select Policy Plug-in Implementation window, select SubjectAltNameExt.
 - c. In the Policy Rule Editor window, set the domain controller GUID for otherName and DNServer as dNSName. Enter values as in the following figure, and then click OK:

The screenshot shows the 'Policy Rule Editor' window for a rule named 'domainControllerSubjAltName'. The 'Policy Plugin ID' is 'SubjectAltNameExt'. The rule is enabled. The predicate is 'HTTP_PARAMS.certType == domainController'. The rule is not critical. It has two general names: 'generalName0' with request attribute 'HTTP_PARAMS.GUID' and choice 'otherName'; and 'generalName1' with request attribute 'HTTP_PARAMS.DNServer' and choice 'dNSName'. A note at the bottom explains that this policy inserts the Subject Alternative Name Extension into the certificate, referencing RFC 2459 (4.2.1.7), and advises using it with an authentication manager that sets 'mail' or 'mailalternateaddress' values in the authToken parameter of the Directory-based authentication plugin.

Policy Rule ID: domainControllerSubjAltName
Policy Plugin ID: SubjectAltNameExt

enable

predicate HTTP_PARAMS.certType == domainController

critical

numGeneralNames 2

generalName0.requestAttr HTTP_PARAMS.GUID

generalName0.generalNameChoice otherName

generalName1.requestAttr HTTP_PARAMS.DNServer

generalName1.generalNameChoice dNSName

This policy inserts the Subject Alternative Name Extension into the certificate. See RFC 2459 (4.2.1.7). * Note: you probably want to use this policy in conjunction with an authentication manager which sets the 'mail' or 'mailalternateaddress' values in the authToken. See the 'ldapStringAttrs' parameter in the Directory-based authentication plugin

OK Cancel Help

7. Set up the Certificate Template Name policy for the Domain Controller certificate.
 - a. In the Policies Management tab, click Add.
 - b. In the Select Policy Plug-in Implementation window, select GenericASN1Ext.
 - c. In the Policy Rule Editor window, set the domain controller GUID for otherName and DNServer as dNSName. Enter values as in the following figure, and then click OK:

Policy Rule ID: domainControllerTemplate
Policy Plugin ID: GenericASN1Ext

enable

predicate

critical

name

oid

pattern

attribute.0.type

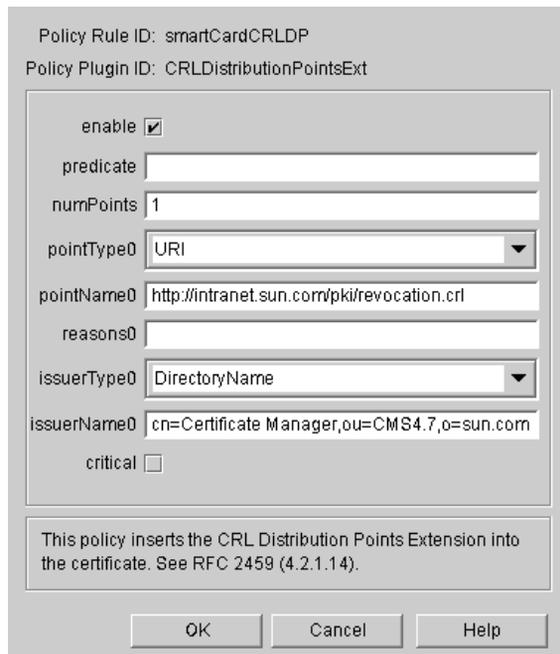
attribute.0.source

attribute.0.value

Adds Private extension based on ASN1. See manual

OK Cancel Help

8. Set up the CRL Distribution Point policy for both Smart Card and Domain Controller certificates.
 - a. In the Policies Management tab, click Add.
 - b. In the Select Policy Plug-in Implementation window, select CRLDistributionPointsExt.
 - c. In the Policy Rule Editor window, set the domain controller GUID for otherName and DNServer as dNSName. Enter values as in the following figure, and then click OK:



Policy Rule ID: smartCardCRLDP
Policy Plugin ID: CRLDistributionPointsExt

enable

predicate

numPoints 1

pointType0 URI

pointName0 http://intranet.sun.com/pki/revocation.crl

reasons0

issuerType0 DirectoryName

issuerName0 cn=Certificate Manager,ou=CMS4.7,o=sun.com

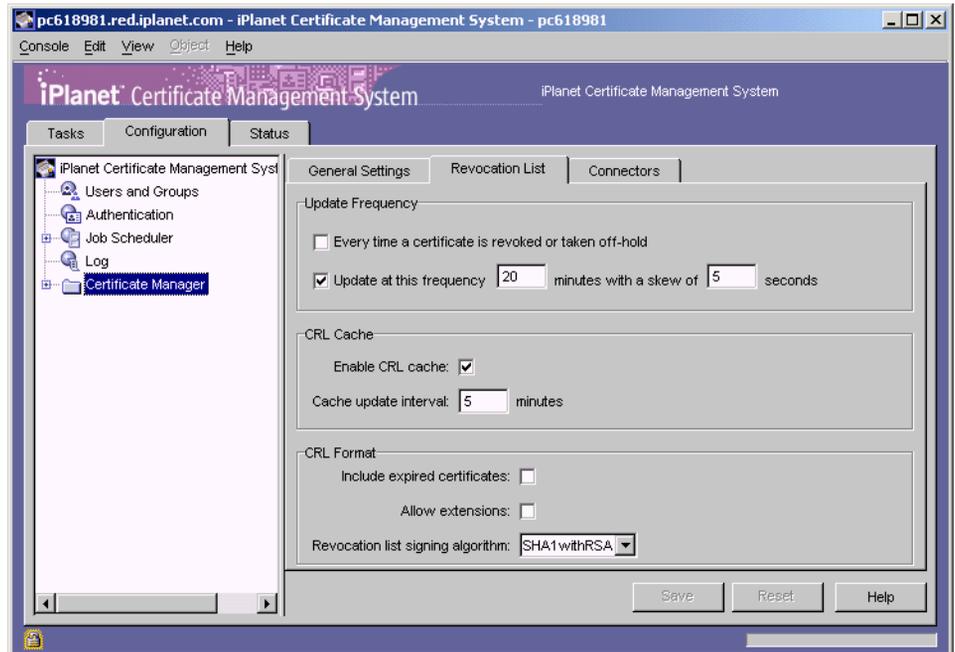
critical

This policy inserts the CRL Distribution Points Extension into the certificate. See RFC 2459 (4.2.1.14).

OK Cancel Help

- d. To configure the CRL Distribution policy for both Smart Card and Domain Controller certificates, enter real values for your deployment in the following fields:
 - o pointName0
 - o issuerName0

9. Make sure that the CRL list is accessible and valid. When the CRL is generated, the Revocation List Update frequency must be set as in the following graphic:



10. Enable the following policies in CMS for all the certificates:

- Subject Key Identifier Extension
- Authority Key Identifier Extension
- Authority Information Access Extension

Part 3. Customization Notes

Once you've set up the Windows 2000 environment and configured Certificate Server to work with it, you can begin issuing certificates. It's likely, however, that you'll want to customize your certificate enrollment forms to better suit your own requirements.

Part 3 is intended to illustrate how certificate issuance for a Windows 2000 environment works with minimal modifications. The notes included here are presented as sample instructions, a starting point from which you can begin to identify your own customization needs and develop custom solutions.

Certificate Server 4.7 provides two sample enrollment forms. One is for the Smart Card Certificate and one is for the Domain Controller Certificate. Both are stored in the following directory:

```
<CMS_Root>/<CMS_Instance>/web/ee.
```

3a. Construct the Domain Controller GUID ASN1 String

In the sample Domain Controller Enrollment form, the value of GUID must be entered. For demonstration purposes, we assume the GUID of the Domain Controller is fb4cdafc-2e1d-4151-a958-b20bfb9e5890. For more information on this value is derived, see

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q224544>.

1. Use the command GenAsn1 provided by CMS4.7 to calculate the ASN1 string for GUID. Example:

```
$ GenAsn1 -o 1.3.6.1.4.1.311.25.1 -t Octet
FC:DA:4C:FB:1D:2E:51:41:A9:58:B2:0B:FB:9E:58:90
BgkrBgEEAYI3GQGgEgQQ/NpM+x0uUUGpWLIL+55YkA==
```

2. Use the output string as the value of GUID in the Domain Controller Certificate Enrollment form. Make sure the value of "GUID-DataType" is "ASN1". Pay special attention to the octet string passed to this command. It is derived from the GUID value.

3b. Construct the UPN Value for the Smart Card Certificate

The User's Principal Name (UPN) field in the Smart Card Certificate Enrollment form accepts an ASN1 string. This string can be generated with the command GenAsn1 as above.

Assume the UPN value for the user of the Smart Card Certificate is user1@sun.com. You can generate the ASN1 string as follows:

```
$ GenAsn1 -o 1.3.6.1.4.1.311.20.2.3 -t UTF8 user1@sun.com
BgorBgEEAYI3FAIDoA8MDXVzZXIxQHN1bi5jb20=
```

Note: The UPN used here, user1@sun.com, must be the Windows logon user ID within that Active Directory domain. And its data type in ASN1 is a UTF8String.

3c. Certificate Enrollment

1. In the Domain Controller Enrollment form, replace the value for `DNServer` with the fully qualified domain name of the Windows Domain Controller machine.
2. Use Microsoft Internet Explorer to enroll for a Domain Controller Certificate from Certificate Server.
3. Export the certificate from Internet Explorer with its private key, and then import it into the Domain Controller's local computer store.
4. As a precaution, remove the Domain Controller Certificate from Internet Explorer after it has exported.
5. In Internet Explorer, enroll for a Smart Card certificate using the Smart Card Certificate Enrollment form.

3d. Certificate Verification

After the certificates are issued, use Internet Explorer to view the extension of `SubjectAltName` in the Smart Card Certificate and in the extensions of `SubjectAltName` and `Certificate Template Name` in Domain Controller Certificate. They should show up in clear text, not in hexadecimal or ASCII raw data.

About otherName in Subject Alt Name Extension

Originally only ASN1 sequence was supported for otherName type in Subject Alt Name extension. In CMS4.7, we support another ASN1 string without the leading character of `0x30`, which is mainly for Windows 2000 support, and ASCII string. To specify the different types in the HTML form, a hidden variable of `"xxx-DataType"` is added, where `"xxx"` is the parameter being used in the extension. The 3 possible values for the data type are

- Sequence, for ASN1 sequence with the leading `"0x30"` character. The old CMS versions support this format and this is the default in CMS4.7.
- ASN1, for ASN1 string without the leading `"0x30"` character.
- String, for ASCII string input.

About otherName in Subject Alt Name Extension

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 *attribute = value*, where *attribute* consists of a tag, such as `o` (organization) or `uid` (user ID), and *value* consists of a value, such as “Sun Microsystems” 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 Confidential 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 iPlanet 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 or a person or organization using certificate-issuing server software (such as 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 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.

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 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 iPlanet 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 iPlanet 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 *registration*.

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.

iPlanet Console The Java application used to set up and manage iPlanet servers.

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 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 *cryptographic module* or *cryptographic service provider*) 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.

PCKS #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 iPlanet Administration Server.

server root The directory used to store CMS and other iPlanet 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 1. In CMS, a password that simplifies the way you sign on to iPlanet Certificate Management Server 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 `jdoe@iPlanet.com`, or a computer can identify itself as a site called `www.iPlanet.com` 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

- accelerators 477
- active logs
 - default file location 793
 - frequency for rotating 795
 - message categories 792
 - naming convention 794
 - See also* logging
- adding
 - administrators 413
 - agents 416
 - automated process 416
 - manual process 417
 - extensions
 - to CA certificates 487
 - to CRLs 675, 701, 739
 - to end-entity certificates 618
 - new authentication instances 553
 - relationship with enrollment forms 559
 - new jobs 593
 - new log event listeners 800
 - new policy rules 618
- Administration Server 342
 - and demo 108
 - NT setup 196
 - relationship to iPlanet Console 342
 - relationship to server root 342
 - starting 343
 - from iPlanet Console 343
 - from the command line 343
 - from the Windows NT Services panel 343
 - stopping 344
 - from iPlanet Console 344
 - from the command line 344
 - from the Windows NT Services panel 344
 - Unix setup 192
- administrator
 - defined 54
- administrator/agent, initial enrollment 136–139, 277–280
- administrators
 - common tasks 348
 - deleting 448
 - designated group 409
 - modifying 444
 - group membership 446
 - login information 444
 - port used for operations 382
 - See also* ports
 - role defined 396
 - setting up 413
 - tools provided
 - CMS window 346
 - iPlanet Console 340
- agent
 - defined 54
- agent enrollment 277–283
- Agent Services interface 68
 - for Certificate Manager agents 69
 - for Data Recovery Manager agents 71
 - for Online Certificate Status Manager agents 71
 - for Registration Manager agents 70
 - URL for 383
- agents
 - authorizing remote key recovery 767
 - deleting 448

- designated groups 410
- modifying 444
 - certificate information 445
 - group membership 446
 - login information 444
- port used for operations 383
 - See also* ports
- revocation checking of certificates 536
- role defined 397
- setting up 416
 - automated process 416
 - manual process 417
- SSL client certificates for 399
 - See also* Agent Services interface
- archiving
 - rotated log files 813
 - users' encryption private keys 759
- Audit log
 - defined 791
 - how to configure 798
 - how to monitor 808
 - logging to Windows NT event log 811
 - See also* logging
- authentication
 - client, with Enterprise Server 3.x 861–878
 - configuration parameters 361
 - decisions for deployment 185
 - defined 531
 - during certificate enrollment 537
 - during certificate renewal 465, 537, 538
 - during certificate revocation 540
 - for administrators 532
 - for agents 534
 - managing from CMS window 556
- authentication instances
 - adding new 553
 - relationship with enrollment forms 559
 - configuration parameters 361
 - deleting 569
 - modifying 570
 - naming convention 554
- authentication modules 55, ??–55, 77, 81–93
 - deleting 574
 - registering new ones 572

B

- buffered logging 794
- built-in plug-in modules
 - See* plug-in modules

C

- CA
 - built-in OCSF service 46
- CA chaining 36
- CA cloning 288
- CA decisions, for deployment 175–179
 - CA renewal 178–179
 - distinguished name 175
 - extensions 177–178
 - root versus subordinate 176
 - signing certificate 176
 - signing key 176
- CA hierarchy 36
 - root CA 36
 - subordinate CA 36
- CA scalability 37, 288
- CA signing certificate 176, 451, 635
 - changing trust settings of 526
 - configuration of 199–202, 212–213
 - deleting 525
 - getting a new one 478, 507
 - nickname 451
 - renewing 478, 515
 - viewing details of 523
- CEP 77, 93–95, 96, 100
- CEP enrollment 833
 - manual 835
 - port number for 846
 - setting up multiple services 844
 - URL 846
 - using a script 834
- certificate chains
 - installing in the certificate database 493
 - why you should install 529
- certificate database
 - how to manage 523
 - what it contains 523

- where it's maintained 523
 - Certificate Database tool 507, 515
 - certificate enrollment
 - authentication during 537
 - Certificate Enrollment Protocol (CEP) 833
 - certificate issuance
 - to routers 833, 845
 - an example 849
 - to servers 821
 - manual enrollment 822
 - Netscape 3.x servers 824
 - to VPN clients 833
 - certificate life-cycle management 48, 84, 98–102
 - Certificate Management System (CMS)
 - access to subsystems 99
 - architecture 73–77
 - command-line utilities 65
 - identifier 193, 196
 - overview of 42–43
 - servlets 55
 - standards supported by 77–79
 - Certificate Manager
 - as root CA 36
 - as subordinate CA 36
 - built-in OCSP service 46, 289
 - CA hierarchy 36
 - CA scalability 288
 - chaining to third-party CAs 36
 - clone CA 174
 - cloning 37, 288
 - configuration of 199–203
 - configuring
 - SMTP settings for notifications 587, 598, 599
 - to use separate SSL server certificates 500
 - to use specific ciphers 506
 - connecting to a Data Recovery Manager 436
 - Data Recovery Manager and 170–175
 - Data Recovery Manager and Registration
 - Manager and 172–175
 - demo and 108
 - enabling interaction with end entities 563
 - enabling OCSP service 723
 - features of 45
 - installed by itself 167
 - interface for agents 69
 - introduced 44, 45
 - key pairs and certificates
 - CA signing certificate 451
 - CRL signing certificate 453
 - getting new ones 507
 - list of 451
 - OCSP signing certificate 453
 - protecting 450
 - remote administration server certificate 457
 - renewing existing ones 515
 - SSL server certificate 455
 - wTLS CA signing certificate 452
 - logging to Windows NT event log 811
 - manual updates to publishing directory 686
 - master CA 174
 - Registration Manager and 168–169
 - serial number range 565
 - specifying IP address for 387
 - what to do if not responding 335
 - what to do when exhausts all serial numbers 565
- certificate renewal 831
 - authentication during 465, 537, 538
 - of server certificates 831
 - certificate request
 - result of policy processing 613
 - certificate revocation
 - authentication during 540
 - reasons for 636
 - who can do this 636
 - Certificate Setup Wizard 478
 - using to install certificate chains 493
 - using to install certificates 493
 - supported data formats 494
 - using to request certificates 479
 - certificates
 - Certificate Manager 182
 - Data Recovery Manager 184
 - for subsystems, summarized 182–185
 - for wireless applications 229, 232
 - how to revoke 636
 - installing 855–859
 - life-cycle management 98–102
 - management formats and protocols 77–78
 - Online Certificate Status Manager 184
 - publishing of 629
 - publishing to files 691
 - publishing to LDAP directory 629, 639

- required schema 642
- Registration Manager 183
- revocation reasons 636
- serial numbers
 - what to do when a CA exhausts all 565
- SSL server, for CMS subsystems 182
- X.509 specification 79
- challenge password 465, 538
- changing
 - CMS instance name 308, 309
 - character set for the name 193, 196, 287, 292
 - format for the name 308
 - group members 446
 - port numbers 384, 722
 - See also* ports
 - trust settings in certificates 526
 - why would you change 526
- changing passwords 323, 336
- checking CMS status 334
- cipher suites for export 882
- ciphers
 - configuring 506
 - defined 504
 - list of 504
 - step-up program for browsers 505
 - supported on the server side 504
 - which ones to choose 505
- client authentication, with Enterprise Server
 - 3.x 861–878
- clone CA 174
- cloning 37
- cloning a CA 288
- CMC 78
- CMMF 77
- CMS administrator
 - defined 54
- CMS agent
 - defined 54
- CMS certificates
 - renewal 450
- CMS data
 - where it's stored 389
- CMS feature list 34
- CMS instance
 - changing the name 308, 309
 - character set for the name 193, 196, 287, 292
 - format for the name 308
 - creating multiple instances 286
 - removing 309
 - viewing information 306
 - file location 307
 - installation date 307
 - on/off/unknown status 308
 - security level 308
 - version number 307
- CMS instances
 - ports and 186–188
 - server groups and 166, 186–188
- CMS key pairs and certificates 450
- CMS watchdog 335
- CMS window
 - Configuration tab 347
 - configuring authentication 556
 - configuring jobs 589
 - configuring network settings 381, 389
 - configuring policies 613
 - how to launch 351
 - introduction 346
 - managing logs 803
 - Status tab 350
 - Tasks tab 347
 - using to manage policies 613
 - using to schedule jobs 589
 - who can launch 351
- CMS. *See* Certificate Management System, Cryptographic Message Syntax
- command-line utilities 65
 - for adding extensions to CMS certificates 487
 - killproc tool 335
- configuration
 - road map 376
 - ways to modify 359
- configuration directory
 - demo and 108
 - NT setup 193, 195
 - Unix setup 191
- configuration directory server
 - Unix setup 190
- configuration file 355
- copying from one instance to another 358
- effects of installation on 356

- format 360
- format for localizable values 361
- guidelines for editing 360
- how subsystem-specific parameters are distinguished 360
- location 358
- name 355
- sample 363
- shared parameters 356
- ways to modify
 - by editing the file 359
 - from CMS window 359
- what is ignored by the server 360
- what it controls 355
- when created 355
- Configuration tab 347
 - tasks you can accomplish 348
- configuring logs 797
 - Audit log 798
 - Error log 798
 - System log 798
 - See also* logging 797
- connecting subsystems 405, 423
 - connection types 406
 - connectors 406
 - why would you do this 405
- conventions used in this book 27
- core features 34
- creating
 - administrators 413
 - agents 416
 - automated process 416
 - manual process 417
- creating multiple CMS instances 286
- CRL
 - complete 289
- CRL Distribution Point extension 639
- CRL signing certificate 453, 635
 - nickname 453, 503
- CRLs
 - Certificate Manager support for 46
 - defined 635
 - issuing or distribution points 638
 - publishing of 39, 635
 - publishing to files 691
 - publishing to LDAP directory 637, 639

- required schema 642
 - publishing to online validation authority 181, 714, 732
 - supported extensions 636
 - supported versions 636
 - when automated updates take place 635
 - when generated 636
 - who generates it 635
- CRMF 77
- Cryptographic Message Syntax (CMS) 78

D

- data formats for installing certificate chains 494
 - binary 494
 - text 494
- data formats for installing certificates 494
 - binary 494
 - text 494
- Data Recovery Manager
 - Certificate Manager and 170–175
 - Certificate Manager and Registration Manager and 172–175
 - configuration of 205–??
 - configuring
 - to use separate SSL server certificates 500
 - to use specific ciphers 506
 - connecting to a Certificate Manager 436
 - connecting to a Registration Manager 424
 - features of 48
 - interface for agents 71
 - introduced 44, 48, 52
 - key pairs and certificates
 - getting new ones 507
 - list of 460
 - protecting 450
 - remote admin server certificate 462
 - renewing existing ones 515
 - SSL server certificate 462
 - storage key pair 461
 - transport certificate 461
 - logging to Windows NT event log 811
 - recovery agents for 208–??
 - setting up

- key archival 775
 - key recovery 782
 - specifying IP address for 387
 - transport certificate 205–208
 - what to do if not responding 335
 - database, internal CMS 109
 - deleting
 - authentication instances 569
 - authentication modules 574
 - certificates from the token 525
 - precaution 525
 - job modules 601
 - jobs 593
 - log event listeners 799
 - log modules 817
 - mapper modules 711
 - policy modules 628
 - policy rules 618
 - privileged users 448
 - publisher modules 711
 - rotated log files 796
 - demo 105–161
 - first user certificate for 136–139
 - installation of 105–161
 - Installation Wizard and 123–136
 - overview of 108–112
 - passwords for 111–112
 - port numbers for 110
 - software installed for 110
 - using 139–161
 - using an LDAP directory with 146–161
 - verifying installation 139–144
 - deployment planning 165–188
 - authentication decisions 185
 - CA decisions 175–179
 - CA renewal/CA renewal 178–179
 - distinguished name 175
 - extensions 177–178
 - root versus subordinate 176
 - signing certificate 176
 - signing key 176
 - certificate decisions
 - Certificate Manager 182
 - Data Recovery Manager 184
 - Online Certificate Status Manager 184
 - Registration Manager 183
 - enrollment scenarios 84–97
 - file-based publishing decisions 180
 - firewall considerations 84
 - hardware token decisions 179
 - LDAP publishing decisions 180–181
 - policy decisions 185–186
 - port assignments 186–188
 - SSL server certificate decisions 182
 - storage key 184
 - subsystem certificate decisions 182–185
 - topology decisions 166–175
 - directory
 - schema for PINs 547
 - distinguished name (DN)
 - for CA 175, 176
 - for CA signing certificate 200
 - for Data Recovery Manager transport certificate 206
 - for Online Certificate Status Manager signing certificate 211
 - for Registration Manager signing certificate 204
 - documentation
 - conventions followed 27
 - where to find 29
 - downloading certificates 855–859
 - DSA 176
- ## E
- email resolver 577
 - end entities
 - enabling interaction with a Certificate Manager 563
 - enabling interaction with a Registration Manager 566
 - enrollment scenarios for 84–97
 - enrollment, steps in 81–83
 - forms for 101
 - forms provided for 72
 - generating PINs for 547
 - life-cycle management and 98–102
 - port used for operations 383
 - See also* ports
 - end-entity certificates
 - renewal 831

- revocation 831
- End-Entity Services Interface
 - introduced 72
- enrollment forms
 - specifying authentication 559
- enrollment scenarios 84–97
 - custom authentication, customer database 86
 - custom authentication, Kerberos 89–91
 - firewall considerations 84
 - manual authentication 87–89
 - PIN-based authentication 91–92
 - routers 96–97
 - VPNs 93–95
- enrollment, initial administrator/agent 277–280
- Enterprise Server 3.x, using SSL with 861–878
- Error log
 - defined 790
 - how to configure 798
 - how to monitor 806
 - See also* logging
- event log
 - logging audit and system messages 811
- event-driven notifications 65
- export control information 879–882
- extensions
 - adding to a CA certificate 487
 - CA certificates and 201–202
 - CAs and 177–178
 - CMS policy modules for 58
 - SSL server certificate 215–216
 - tool for joining 487
 - tools for generating 487
 - transport certificate 207
- external tokens
 - defined
 - installing 466, 468
 - viewing contents of 523

F

- filenames
 - for active log files 794
 - for rotated log files 794

- FIPS PUBS 140-1 78, 465
- firewalls 84
- flush interval for logs 794
- fonts used in this book 27

G

- gateway
 - agent, for demo 136
 - end user, for demo 136
- generating PINs for end entities 547
- getting new certificates for subsystems 507
- groups
 - changing members 446
 - defined 409
 - for administrators 409
 - for agents 410
 - for trusted managers 412
 - where they're maintained 409

H

- hardware accelerators 477
- hardware requirements for CMS installation 106
- hardware token decisions, for deployment 179
- hardware tokens
 - See* external tokens
- host name
 - for mail server used for notifications 587, 599
- how to check whether CMS is on or off 334
- how to renew CMS certificates 450
- how to revoke certificates 636
- how to search for keys 762
- HTML forms
 - for agents 68
 - for end entities 72

- I
- installation 217–283
 - additional instances 286
 - demo 105–161
 - first user certificate for 136–139
 - Installation Wizard and 123–136
 - NT installation script for 114–122
 - overview of 108–112
 - passwords for 111–112
 - Unix installation script for 112–114
 - using 139–161
 - verifying 139–144
 - hardware requirements 106
 - location of
 - NT setup 193
 - Unix setup 190
 - overview 217
 - port considerations 186–188
 - software requirements 106
 - Solaris requirements 106, 107
 - system requirements 106–107
 - Windows NT requirements 107
 - worksheet 189–216
- installation date 307
- installation script
 - information requested by 190–196
 - NT
 - complete instructions 224–227
 - running for demo 114–122
 - worksheet for 193–196
 - Unix
 - complete instructions 221–224
 - running for demo 112–114
 - worksheet for 190–193
- Installation Wizard
 - initial configuration steps 196–199
 - procedures for using 227
 - running for demo 123–136
- installing certificates 855–859
- installing external hardware tokens
 - Level 2 466
 - Level 3 468
- installing multiple CMS instances 286
- instances, CMS
 - agents for additional 280–283
 - creating additional 286
- internal CMS database 109
- internal database
 - default host name 391
 - precaution for changing the host name 391
 - defined 389
 - how to distinguish from other Directory Server instances 390, 392
 - introduced 54
 - name format 390, 392
 - schema 390
 - what you shouldn't do 390
 - what is it used for 389
 - when installed 390
- internal tokens
 - viewing contents of 523
- IP address 387
- IP addresses, and port assignments 188
- iPlanet Console
 - checking CMS status 334
 - demo and 108
 - how to launch 344
 - in Unix 345
 - in Windows NT 345
 - installing multiple CMS instances 286
 - introduction 340
 - opening CMS window 351
 - relationship to Administration Server 342
 - removing a CMS instances 309
 - restarting Certificate Management System 332, 726, 751, 752
 - starting Administration Server 343
 - starting Certificate Management System 327
 - starting Installation Wizard from 227
 - stopping Administration Server 344
 - stopping Certificate Management System 330
 - viewing CMS instance information 306
- issuing certificates
 - to routers 833, 845
 - an example 849
 - to servers 821
 - manual enrollment 822
 - Netscape 3.x servers 824
 - to VPN clients 833

J

- Java 2 76
- Java/JNI 76
- JavaScript policy processor 626
- job modules
 - deleting 601
 - registering new ones 600
- job scheduler 61
- jobs
 - adding new 593
 - configuration parameters 362
 - created during installation 591
 - deleting 593
 - managing 589
 - managing from CMS window 589
 - modifying 590
 - naming 593
 - naming convention 593
 - setting frequency 597
 - turning on scheduler 597
- JSS 76

K

- key archival 761
 - how it works 763
 - how keys are stored 762
 - how to set up 775
 - PKI setup required 759
 - where keys are stored 762
 - why you should archive 761
- key features 34
- key length 176
- key pairs and certificates used by CMS 450
- key recovery 765
 - designated agents
 - See key recovery agents
 - how to set up 782
 - interface for agents 766
 - local vs. remote 767
- key recovery agents
 - passwords 765
 - significance 765

- when specified the first time 765
 - responsibilities 765
 - role defined 765
- KEYGEN tag 78
- killproc tool 335

L

- LDAP 78
- LDAP directory
 - configuration, demo and 108
 - DN pattern for authentication 147
 - internal CMS database, demo and 109
 - publishing decisions 180–181
 - testing authentication with 146–161
- LDAP publishing
 - advantages 630
 - defined 629
 - manual updates 686
 - when to do 687
 - who can do this 686
 - See CRLs
- linked CA 36
- linking subsystems
 - See connecting subsystems
- List Certificate page, configuring 424
- local vs. remote key recovery 767
- location of
 - active log files
 - CMS configuration file 358
 - CMS documentation 29
 - rotated log files 796
- log event listeners
 - adding new 800
 - created during installation 797
 - deleting 799
 - modifying 798
 - naming convention 800
- log modules
 - deleting 817
 - registering new ones 816
- logging
 - buffered vs. unbuffered 794

- configuring
 - Audit log 798
 - Error log 798
 - System log 798
- log files
 - archiving rotated files 813
 - automatic deletion 796
 - automatic rotation 795
 - default location 793
 - location of rotated files 796
 - naming convention for active logs 794
 - naming convention for rotated logs 794
 - significance of deleting files 796
 - signing rotated files 814
 - timing of rotation 795
- log levels 792
 - default selection 793
 - how they relate to message categories 792
 - how they're represented 792
 - significance of choosing the right level 793
 - what it means 792
- managing from CMS window 803
- monitoring
 - Audit log 808
 - Error log 806
 - System log 804
 - using system tools in Windows NT 811
- parameters in the configuration file 362
- services that are logged 791
- types of logs 790
 - Audit 791
 - Error 790
 - System 790
- privileged users 395
- publisher plug-in modules 709
- schedulable jobs 589
- mapper modules
 - deleting 711
 - list of 63
 - registering new ones 709
- mappers
 - created during installation 661
 - modifying 661
- mapping certificates to directory entries 63
- master CA 174
- message templates for notifications 578
- migration. See "upgrading from a previous version."
- modifying
 - authentication instances 570
 - jobs 590
 - log event listeners 798
 - mappers 661
 - policy rules 614
 - privileged user's group membership 446
 - privileged-user information 444
 - publishers 663, 664
- monitoring logs 803
 - Audit log 808
 - Error log 806
 - System log 804
 - things you can monitor 803
 - using system tools in Windows NT 811
 - See also* logging

M

- m of n secret sharing 765
- mail server used for notifications 587, 599
- managing
 - certificate database 523
 - job plug-in modules 599
 - log plug-in modules 816
 - mapper plug-in modules 709
 - policies 613
 - policy plug-in modules 626

N

- naming convention
 - for active logs 794
 - for authentication instances 554
 - for CMS instances 193, 196, 287, 292
 - for internal database instances 390, 392
 - for log event listeners 800
 - for policy rules 618
 - for rotated logs 794
 - for schedulable jobs 593
- nickname
 - for CA signing certificate 451

- for CRL signing certificate 453, 503
- for OCSP signing certificate 453
- for remote administration server certificate 457
- for signing certificate 459, 463
- for SSL server certificate 455, 459, 462, 463
- for transport certificate 461
- for wTLS signing certificate 452

notifications

- configuring the mail server 598
 - host name 587, 599
 - port 587, 599
- customizing 578
 - templates 580
- event-driven 575
 - when certificates are issued 576
 - when new requests are queued 577

notifications, event-driven 65

NSS 76

O

OCSP 49

OCSP client 718

OCSP responder 181, 714

- defined 49

OCSP server 181, 714

OCSP service

- enabling in Certificate Manager 723

OCSP signing certificate 453

- nickname 453

Online Certificate Status Manager

- configuration of 211
- interface for agents 71
- introduced 44, 49
- key pairs and certificates
 - list of 463
 - protecting 450
 - remote admin server certificate 464
 - signing certificate 463
 - SSL server certificate 463
- logging to Windows NT event log 811

online certificate validation authority

- defined 49

operating systems supported 106

P

password cache 336

password-quality checker 323, 337

passwords

- changing cached 323, 336
- See also* single signon passwords

PIN Generator tool

- delivering PINs to users 568
- directory schema requirements 547
 - changing 3.x directory schema 547
 - changing 4.x directory schema 547
- generating PINs 547

PIN present constraints policy 550

PKCS #10 79

PKCS #11 74–76, 79

PKCS #11 support

PKCS #7 79

PKI. *See* installation script.

PKI. *See* Public-Key Infrastructure.

PKI. *See* distinguished name (DN).

pkiclient.exe 846

PKIX 78

plug-in modules

- for authentication 55
- for logs
 - managing 816
- for mappers
 - managing 709
- for policy 619
 - managing 626
- for publishers
 - managing 709
- for publishing
 - list of 63, 64

policies in JavaScript 626

policy

- built-in plug-in modules 619
- configuration parameters 362
- defined 604
- managing 613

- managing from CMS window 613
- processor 612
 - how it applies rules 613
 - JavaScript 626
 - result of processing 613
 - when used 612
- what can you use it for 604
- policy modules 55, 57, 61, 83
 - decisions for deployment 185–186
 - deleting 628
 - registering new ones 626
- policy rules
 - adding new 618
 - configuration parameters 362
 - created during installation 614
 - defined 605
 - deleting 618
 - how policy processor applies them 613
 - modifying 614
 - naming convention 618
 - predicates in 606
 - reordering 623
 - significance of ordering 623
 - See also* predicates
 - types of 605
 - what each rule does 605
- port numbers
 - assignment of 186–188
 - for demo 110
 - IP addresses and 188
- ports 381
 - changing numbers 384, 722
 - for agent operations 383
 - for end-entity operations 383
 - turning on/off HTTP port 386
 - for remote administration 382
 - for the mail server used for notifications 587, 599
 - how to choose numbers 382
- predicates
 - attributes for 608
 - expression support 606
 - operators for 606
 - sample expressions 606, 608
 - what are they 606
 - why would you use 606
- privileged users 395, 396
 - deleting 448
 - groups 409
 - modifying privileges 444
 - certificate information 445
 - group membership 446
 - login information 444
 - setting up 413
 - administrators 413
 - agents 416
 - trusted managers 423
 - types 396
 - administrators 396
 - agents 397
 - determining factor 396
 - trusted manager 405
 - types or roles 396
- protecting private keys 450
- Public-Key Infrastructure (PKI) 43
- publisher modules
 - deleting 711
 - list of 64
 - registering new ones 709
- publishers
 - created during installation 661
 - modifying 663, 664
- CRLs
 - publishing
 - See also* LDAP publishing
- publishing
 - of certificates 629
 - to files 691
 - to LDAP directory 629, 639
 - of CRLs 635
 - to files 691
 - to LDAP directory 637, 639
 - to online validation authority 181, 714, 732
 - See* LDAP publishing
- publishing certificates to directory entries 64
- publishing directory
 - defined 52, 629
- publishing rules
 - created during installation 661

R

- reasons for revoking certificates 636
- recovering users' private keys 765
- registering
 - authentication modules 572
 - job modules 600
 - log modules 816
 - mapper modules 709
 - policy modules 626
 - publisher modules 709
- Registration Manager
 - Certificate Manager and 168–169
 - Certificate Manager and Data Recovery Manager and 172–175
 - configuration of 203–??
 - configuring
 - List Certificates page 424
 - SMTP settings for notifications 598
 - to use separate SSL server certificates 500
 - to use specific ciphers 506
 - connecting to another subsystem 424
 - enabling interaction with end entities 566
 - features of 48
 - interface for agents 70
 - introduced 44, 47
 - key pairs and certificates
 - getting new ones 507
 - list of 459
 - protecting 450
 - remote admin server certificate 460
 - renewing existing ones 515
 - signing certificate 459
 - SSL server certificate 459
 - logging to Windows NT event log 811
 - specifying IP address for 387
 - what to do if not responding 335
- Remote admin server certificate 460
- remote admin server certificate 462, 464
- Remote administration server certificate 457
 - nickname 457
- removing unwanted CMS instances 309
- renewal of certificates
 - See certificate renewal
- renewal of CMS certificates 450
- renewing certificates of subsystems 515
- reordering policy rules 623
 - significance of ordering 623
- restarting
 - Certificate Management System 332
 - from iPlanet Console 332, 726, 751, 752
 - from the command line 333
- revocation checking of agent certificates 536
- revocation-status checking for agent certificates 402
- revoking certificates 831
 - reasons 636
 - who can do this 636
- road map to configuring subsystems 376
- roles
 - administrator 396
 - agent 397
 - determining factor 396
 - key recovery agents 765
 - trusted manager 405
- root CA 36
- root versus subordinate CA 176
- rotated logs
 - naming convention 794
- rotating log files 795
 - archiving files 813
 - conserving disk space 796
 - how to set the time 795
 - signing files 814
- routers
 - getting certificates for 833, 845, 849
 - port used for requesting 846
- RSA 176

S

- scalability 288
- schedulable jobs
 - See jobs
- scheduling
 - jobs 589
- secret sharing of storage key pair 765
- security level 308
- server certificate 213–216

- server certificate renewal 831
- server group 166
- server groups 166
- server instance
 - finding out details 306
- server name
 - changing 308
- server root 166
 - default for Unix 307
 - default for Windows NT 307
 - defined 307
 - how many on a single host 307
 - relationship with Administration Server 342
- server status
 - off 308
 - on 308
 - unknown 308
- server's on/off status 334
- servlets, CMS 55
- setpin.conf file 548
- setting CRL extensions 675, 701, 739
- setting up
 - key archival 775
 - key recovery 782
- setup script 161
- signing
 - rotated log files 814
- signing algorithms 46
- signing certificate 459, 463
 - CA 176, 199–203, 212
 - changing trust settings of 526
 - deleting 525
 - getting a new one 478, 507
 - nickname 459, 463
 - Online Certificate Status Manager 209–??
 - Registration Manager 203–211
 - renewing 478, 515
 - viewing details of 523
- signing key, for CA 176
- single sign-on password 216, 335
- single signon password
 - changing cached passwords 323, 336
 - starting CMS without 323
 - what it does 323
 - what it protects 322
- when required 322
 - when specified 323
 - why change periodically 323
- Smart Card
 - CMS customization notes 893
 - configuring CMS 4.7 884
 - configuring login with Windows 2000 883
 - otherName in Subject Alt Name extension 895
 - setting up Windows environment 883
- SMTP settings 587, 598, 599
- software requirements for CMS installation 106
- Solaris
 - requirements for installation 107
- Solaris requirements for installation 106
- specifying IP address 387
- SSL 79
 - cipher suites approved for export 882
 - server certificate 213–216
 - using with Enterprise Server 861–878
- SSL server certificate 455, 459, 462, 463
 - changing trust settings of 526
 - deleting 525
 - getting a new one 478, 507
 - nickname 455, 459, 462, 463
 - renewing 478, 515
 - viewing details of 523
- starting
 - Administration Server 343
 - from iPlanet Console 343
 - from the command line 343
 - from the Windows NT Service panel 343
 - Certificate Management System 322
 - from iPlanet Console 327
 - from the command line 328
 - from the Windows NT Services panel 329
 - information required 322
 - iPlanet Console 344
 - in Unix 345
 - in Windows NT 345
- Status tab 350
 - tasks you can accomplish 350
- stopping
 - Administration Server 344
 - from iPlanet Console 344
 - from the command line 344
 - from the Windows NT Services panel 344

- Certificate Management System 330
 - from iPlanet Console 330
 - from the command line 331
 - from the Windows NT Services panel 332
- storage key pair 461
 - secret sharing 765
- storage key, for Data Recovery Manager 184
- stronger encryption for export browsers 505
- subject name 189
- subordinate CA 36
- subsystem certificate decisions 182–185
- subsystem certificate decisions, for deployment
 - Certificate Manager 182
 - Data Recovery Manager 184
 - SSL server 182
- Sun ONE 23
- support for
 - OCSP client 718
 - publishing of CRLs 39
- System log
 - defined 790
 - how to configure 798
 - how to monitor 804
 - logging to Windows NT event log 811
 - See also* logging
- system requirements for CMS installation 106–107

T

- Tasks tab 347
 - tasks you can accomplish 347
- templates
 - for notifications 578
 - customizing 580
 - token list 581
 - templates
 - for automated notifications 578
- timing log file deletion 796
- timing log rotation 795
- tokens
 - changing password of 477
 - deleting certificates from 525
 - external 465

- See also* external tokens
 - internal 465
 - managing 476
 - viewing contents of 523
 - viewing which tokens are installed 476
 - what are they 464
- topology decisions, for deployment 166–175
- transport certificate 461
 - changing trust settings of 526
 - deleting 525
 - getting a new one 478, 507
 - nickname 461
 - renewing 478, 515
 - viewing details of 523
 - when used 764
- transport certificate, for Data Recovery Manager 205–208
- trusted managers
 - certificate for SSL client authentication 408
 - connectors for linking 406
 - deleting 448
 - designated group 412
 - access rights 412
 - modifying 444
 - certificate information 445
 - group membership 446
 - login information 444
 - role defined 405
 - setting up 423
- type styles used in this book 27

U

- unbuffered logging 794
- uninstalling Certificate Management System 311
 - from the command line 311
 - using Windows NT Add/Remove Programs utility 311
- upgrading from a previous version 41
 - from version 4.2 SP2 to 4.7 313
 - migration tool
 - to version 4.2 SP2 316
- user/group directory
 - NT setup 194

- user/group directory server
 - Unix setup 191
- users
 - privileged 395
- utilities, command-line 65

V

- version number 307
- viewing
 - contents of a token 523
- viewing CMS instance information 306
- VPN clients
 - getting certificates for 833

W

- watchdog 335
- when the server was installed 307
- why should you revoke certificates 636
- Windows NT event log
 - logging audit and system messages 811
- Windows NT, requirements for installation 107
- wireless CA certificate 229, 232
- wireless certificates 229, 232
- wizard
 - See Certificate Setup Wizard
- writing policies in JavaScript 626
- wTLS CA signing certificate 452
 - nickname 452
- wTLS certificates 229, 232

X

- X.509 certificates 79