

ITProTM
SERIES

WindowsITPro

 **eBooks**

Keeping Your Business
SAFE from Attack:

Encryption and Certificate Services

By Jan De Clercq

Microsoft[®]



Contents

Chapter 6: Building a Windows PKI	114
Introduction	114
Assessing the Organizational Needs for a PKI	115
Analyzing Business Requirements	115
PKI-Enabled Applications	116
Insource or Outsource?	116
Designing, Planning, and Implementing an Outsourced PKI Solution	118
Designing, Planning, and Implementing an Enterprise Windows PKI	119
PKI Policy Definition	119
Defining Your PKI Topology	121
Individual CA Specifications	121
Preliminary Planning	122
CA Hardware Sizing	122
Offline CAs	123
CA Installation Options	123
CA Role	124
CA Keys and Certificate	124
CA Naming Conventions	126
The CA Database	128
Other CA Installation Options	129
CA Configuration Options	129
Revocation Settings	130
AIA Settings	134
Other Certificate Characteristics	135
CA Administrative Delegation and Role Separation	135
CA Server Hardening	139
CA Fault Tolerance	139
Defining Public Key Policy Settings	140
Conclusion	142

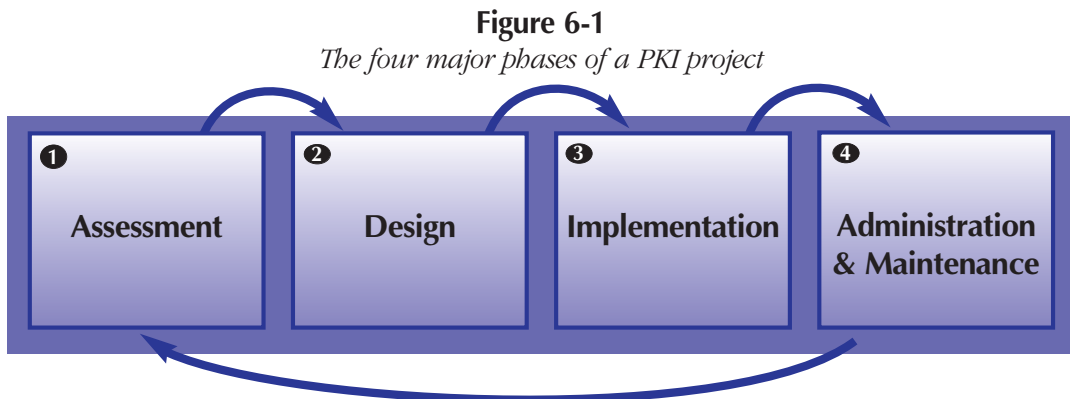
Chapter 6:

Building a Windows PKI

In the previous chapters of this ebook, we explained some of the technical nuts and bolts of Windows Server 2003 PKI. In this chapter, we look at the different steps you must consider when you plan, design, and build a Windows-rooted public key infrastructure (PKI).

Introduction

Like any other IT project, a PKI project can be split into four key phases: assessment, design, implementation, and management (administration and maintenance). Figure 6-1 illustrates these phases.



A PKI project can be iterative:

- During the implementation phase, for example, issues might arise that require a new assessment and changes to the original design.
- During the assessment phase, the current and future security requirements of an organization are analyzed. This analysis can comprise running a security audit, performing a penetration test, or just analyzing existing processes. The assessment phase also includes a business-requirement analysis.
- During the design phase, the technological and nontechnological design of the PKI solution are the focus. Nontechnological design topics include the creation of certificate policies and certification practice statements (CPSs).
- During the implementation phase, before the rollout, is the development of customized PKI-enabled applications or PKI software plug-ins; and then the rollout of the PKI solution and its integration with the existing IT environment are taken care of.

Once the PKI is installed and deployed across your enterprise, you must manage and maintain it. In the management phase, you must set up the support model for the PKI (the Help desk), the PKI administrator, user training, and the management of the PKI components.

Assessing the Organizational Needs for a PKI

During a PKI assessment, you must analyze your company's current and future security needs. As part of the assessment, you can organize a security audit or a penetration test if you need to gather some extra information. The next sections focus on three key areas of the assessment phase: the business-requirement analysis, the decision about whether to insource or to outsource the PKI infrastructure, and the analysis of the applications that need PKI-based security.

Analyzing Business Requirements

Core business needs, such as advanced security requirements for information storage and network communication, typically drive the rollout of a PKI in an enterprise. The size of the investment a company makes in a PKI will depend upon the criticality of the business problem the organization wishes to resolve with the PKI, or the importance of the business processes whose security level the company wants to improve by installing a PKI.

Some business security requirements do not need a certificate-based security solution and so can be resolved with simpler arrangements than a PKI. Also, not every certificate-based solution requires an enterprise PKI. For some solutions, buying a limited set of certificates from a commercial certification authority such as Verisign or Thawte is enough. This approach can also be a cost-effective solution for smaller organizations that require only a couple of certificates—for example, to secure access to their Web site using Secure Sockets Layer (SSL) and a server-side SSL certificate.

The business your organization is in may impose special requirements in the following areas:

- *Availability.* What level of availability does the PKI need within the organization? The answer to this question affects the design of the Certification Authorities (CAs), the CA databases, and the directories that are used in the PKI solution.
- *Scalability.* How scalable must the PKI be? Will it have to deal with rapid growth of the number of required certificates? Does planning have to take into account company mergers and acquisitions? Will more and more PKI-based applications be deployed in the future? Good advice is to always plan for growth. Remember that PKI is a fundamental security service that can be used by many applications and users.
- *Performance.* PKI and its public-key cryptography operations create an additional performance load for computer systems. Is this extra load acceptable? Should the existing hardware be upgraded? Should you install additional hardware that speeds up PKI operations?
- *Cost.* PKI products come in different flavors, with different features, and in different price classes. Smaller organizations may opt to buy certificates from a commercial CA instead of designing and operating an internal PKI. A key price advantage of Windows PKI is that it comes with the Windows OS software. You can use PKI and PKI-enable your applications without needing to buy some specialized third-party PKI software.
- *Legal and regulatory compliance.* Some organizations may want to create a PKI and PKI-enabled applications for regulatory and legal compliance reasons. Examples of regulations that can drive the creation of a PKI are the Health Insurance Accountability and Portability Accountability Act of 1996 (HIPAA) and the federal securities-related Sarbanes-Oxley Act (in the United States), and the revised international capital framework, Basel II (in Europe).

PKI-Enabled Applications

A PKI is an infrastructure, and many Windows applications can take advantage of PKIs to provide strong security services to their users. Among these applications are networking systems, VPN systems, enterprise resource-planning (ERP) software, document signing, and smart card-based applications.

You can build the following PKI-enabled applications on top of a Windows PKI:

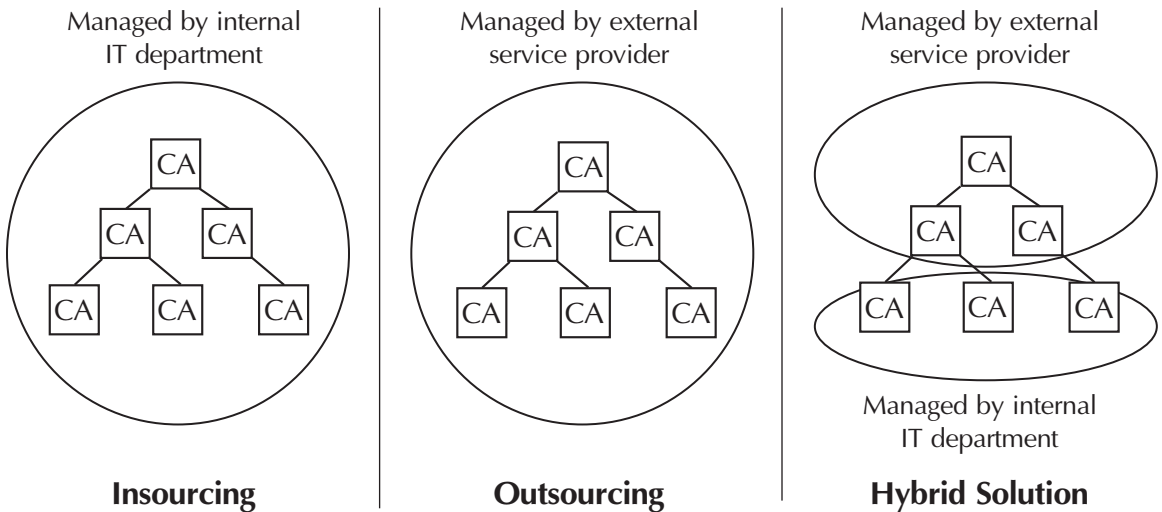
- *Secure Web.* You can use certificates for strong authentication using the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.
- *Secure mail.* Signing and sealing electronic mail messages using S/MIME also builds on public-key cryptography.
- *File-system encryption.* Windows 2000 and later Microsoft OSs come with the Encryption File System (EFS) extension to the NTFS version 5 file system.
- *Code signing.* Code signing protects against the downloading of malicious code. The Microsoft code-signing technology is known as Authenticode.
- *Document signing.* Document-signing technology provides the capability to add a digital signature to, for example, a Word document.
- *Smart card logon.* Smart card logon provides strong two-factor authentication in a Windows 2000 or Windows Server 2003 domain environment.
- *Virtual private networking (VPN).* Windows 2000 and later Microsoft OSs support the IPsec tunneling protocol, which can use certificates to authenticate IPsec tunnel endpoints.
- *Remote access authentication.* Both the Windows 2000 and the Windows Server 2003 RAS support the Extensible Authentication Protocol (EAP), which can deal with certificate-based Transport Layer Security (TLS) authentication.
- *Wireless authentication.* Windows Server 2003 and Windows XP support certificate-based authentication for wireless network access.
- *Secure SMTP site connections.* You can connect Windows 2000 and Windows Server 2003 Active Directory (AD) sites using asynchronous SMTP connections. When you do so, the bridgehead domain controllers authenticate to one another using certificates. This setup also protects the confidentiality and integrity of AD replication traffic.
- Any custom PKI-enabled application that uses CryptoAPI.

Large enterprises typically build a PKI to PKI-enable several of the above-listed applications. Typical PKI application needs for smaller organizations are Web-server authentication based on the SSL/TLS protocols and secure email based on the S/MIME standard.

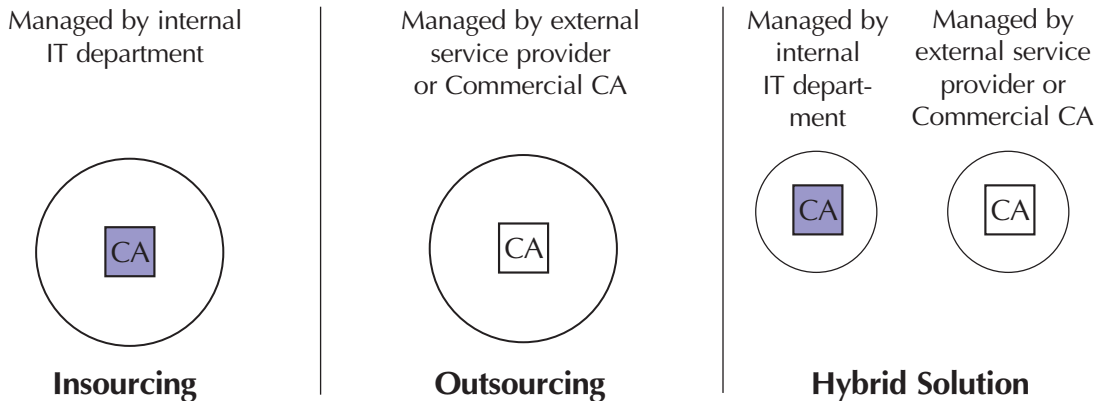
Insource or Outsource?

You have three choices for implementing and managing a PKI: to insource, to outsource, or to use a hybrid approach. Figure 6-2 illustrates models of these three approaches for both large and small organizations.

Figure 6-2
Insourcing and outsourcing PKI models



Large Organizations



Small Organizations

Insourced PKI solutions are solutions that you install, implement, administer, and maintain, all by yourself. Insourcing means that your IT department must take the lead in implementing all related PKI technologies: CA hardware, CA database, remote automations (RAs), directories, and the communication links among all participating entities. This path offers you complete independence. You can create your own liability rules and security policies, and you can decide how to implement, administer, and maintain the PKI.

For a large enterprise, insourcing typically means that you will design, create, and operate an internal PKI hierarchy that consists of several CA server levels. For a small enterprise, insourcing typically comes down to the design, creation, and operation of a single internal CA.

Outsourced solutions turn most of these responsibilities over to another company. The degree of outsourcing can range from a little (generating some server certificates) to a lot (outsourcing multiple CA services that are dedicated to your company). Outsourcing is often the best solution for smaller companies, or for those businesses without the required funds and resources to install and maintain a proper PKI.

For a large enterprise, outsourcing typically means that an external organization will design, create, and operate the company's PKI hierarchy. For a small enterprise, outsourcing typically comes down to hiring an external company for the design, creation, and operation of a single CA—or simply for buying a couple of certificates from a commercial CA.

Hybrid solutions combine insourcing and outsourcing: Your company maintains some of the CAs, and another company maintains the rest. The company itself, for example, can implement and manage the CAs who are issuing certificates for applications with high security needs. The implementation and management of CAs who are issuing certificates to applications with less robust security needs can then be outsourced. For a large enterprise, this model typically means that part of the company's PKI hierarchy is designed, created, and operated by an external organization, and another part is designed, created, and operated internally. It can also mean that, for certain certificate types an internal PKI is used, and, for others, a commercial CA. The same principle applies to small enterprises, but again on a much smaller scale.

Table 6-1 can help you choose between insourced and outsourced PKI approaches, depending on your organization's size and other requirements.

Table 6-1 Advantages and disadvantages: insourcing versus outsourcing

	Insourcing	Outsourcing
Pros	<ul style="list-style-type: none"> • Offers more and tighter control over certificate policy definition, certificate and key management, certificate issuance, key archival, and key recovery. • Offers tighter integration options: integration with enterprise directory and in-house applications. • Means potentially stronger trust relationships with partners because of in-house certificate policy control. 	<ul style="list-style-type: none"> • Can leverage expertise of PKI experts. • Requires less effort for planning, design, administration, and maintenance. • Can be more cost-effective for a small enterprise. • Can be operational in a short period of time. • Requires less in-house expertise.
Cons	<ul style="list-style-type: none"> • Is more expensive (e.g., cost of planning, design, administration, and maintenance). • Requires more in-house expertise. • Means possible complex integration and deployment. • Requires more time to plan, design, and deploy. 	<ul style="list-style-type: none"> • Offers less policy control and fewer enforcement capabilities. • Offers fewer integration options. • Can be more costly for a large enterprise.

Designing, Planning, and Implementing an Outsourced PKI Solution

For an outsourced PKI, the design, planning, and implementation steps are much simpler and shorter than those for an insourced PKI. As noted, outsourcing is typically the approach smaller organizations that don't need and want the overhead of designing and operating an internal PKI take.

In these outsourced scenarios, you must include the following steps in the design, planning, and implementation of your PKI-based solution:

1. *Define the certificate needs.* What kind of certificates do I need? For which applications, servers, users, devices?
2. Define the certificate characteristics. What are the certificate content requirements? Do any of the PKI-enabled applications have special certificate requirements? These requirements can be related to the certificate lifetime, the content of X.509 certificate extensions, and so on.
3. Perform a commercial CA market analysis. Which commercial CA can offer the certificate services you specified in the previous steps? At what price?
4. Negotiate with a commercial CA, or subset of commercial CAs, and decide upon which CA can best serve your needs. Obviously, this step also will take cost factors into account. During the negotiation, make sure that you consult the following data regarding the operation of the commercial CA:
 - CPSs and certificate policies
 - Legal constraints—for example, liability statements
 - Operational procedures of the commercial CA:
 - How is the commercial CA's root CA certificate distributed as a trust anchor in my organization?
 - How does certificate revocation work? How is revocation information made available to the users of the commercial CA?
 - What happens if a certificate expires? How will certificate renewal work?
 - What happens in case of a disaster (commercial root CA trust compromise, unavailability of the commercial CA service, etc.)
5. *Provide basic PKI training for the people who will deal with the certificates you bought from the commercial CA.* At the least, these individuals should understand the basics of PKI, how to link a certificate to an application, how to set up revocation checking, what to do if the certificate expires, and how to distribute the commercial CA's root CA trust anchor.

Designing, Planning, and Implementing an Enterprise Windows PKI

In this section, we focus on the design, planning, and implementation steps you must consider when you design an internal enterprise PKI. This is the strategy you will need to use when you take the insourcing or hybrid approaches outlined previously.

In the sections that follow, we focus on the different steps you must consider when you are designing, planning, and implementing a Windows-rooted enterprise PKI. With the exception of the next section, the following text focuses on the technical parts of a Windows PKI design, plan, and implementation.

PKI Policy Definition

An important nontechnical aspect that technology-focused PKI planners often overlook or neglect is the definition of the CPSs and the certificate policies. Both document categories are derived from a company's security policy. Although we have discussed these terms earlier in the ebook, we review PKI policy definition, CPSs, and certificate policies in this section in light of the chapter's focus.

CPSs and certificate policies help the PKI user determine the level of trust that he or she can put in the certificates issued by a CA that is part of the PKI. When you are dealing with a PKI that is

used to secure applications associated with highly confidential or valuable information, the availability of policies is critical. The creation of a CPS and a certificate policy is less critical when you are using a PKI to secure applications that have less robust security requirements. In those circumstances, adding some extra clauses to the employment agreement regarding the use of PKI and certificates in your company can work.

The security policy is a high-level document that defines a set of rules regarding the availability and use of security services within an organization. The security policy reflects an organization's business and IT strategy, and provides a context for an enterprise's security services. In the context of PKI, the security policy answers high-level PKI questions such as "What applications should be secured with certificates?" and "What kind of security services should be offered using certificates?"

A certificate policy focuses on certificates and the CA's responsibilities regarding these certificates. The policy defines certificate characteristics such as certificate use, enrollment procedures, liability issues, and so on. The X.509 standard defines a certificate policy as "a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements." You can download the X.509 standard from <http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-X.509>.

A certificate policy answers the following questions:

- For what applications can the certificate be used?
- How can a user enroll for the certificate?
- How are users identified when they request a certificate?
- What is a certificate's lifetime?
- How is renewal defined? Is a new key pair generated every time a certificate is renewed?
- What key lengths and ciphers are used to generate the certificate?
- Where is the private key stored? How is it protected? Can it be exported?
- What is the CA's liability when its private key is compromised?
- How should users react when they lose their private keys?

A certificate policy is defined by a group of people known as the *policy authority* within your organization. This group should consist of representatives of the different key departments of your organization: management, legal, audit, human resources, and so forth. In most organizations, the policy authority members are also members of the group that defined the security policy. This distinction assures that the certificate policy is in line with the security policy.

The CPS translates certificate policies into operational procedures for CAs. Whereas a certificate policy focuses on a certificate, a CPS focuses on a CA. Both the European Electronic Signature Standardization Initiative (EESSI) and the American Bar Association (ABA) define a CPS as "a statement of the practices that a certification authority employs in issuing certificates."

A CPS answers the following questions:

- What certificate policy or policies does the CA implement?
- What are the policies for issuing certificates? How are certificates issued? Are they issued directly to users, or published into a directory? What types of certificates can the CA issue, and to which users?
- Who can administer the CA? What subtasks are delegated to the different administrators?
- What are the revocation policies? How is certificate revocation handled?
- When is a certificate revoked? Where are CRLs published? How often are CRLs updated?

- How is access to the CA physically and logically secured?
- Who is responsible for backing up the CA?
- What is the quality of the CA certificate and private key? What is the lifetime of the CA keys and the certificate? What is the CA key length?
- Where and how is the CA private key stored?
- What is the procedure for CA rollover?

Members of your IT department, people who are operating and administering the IT infrastructure, and the people who defined the certificate policy should define the CPS. You can find good examples of a CPS on the Web sites of the following commercial CAs:

- Globalsign: <http://www.globalsign.net/repository>
- Verisign: <http://www.verisign.com/repository/CPS>

A reference to the certificate policy and CPS to which the CA adheres are made available in the CA certificate. To do this, the CA embeds a unique certificate policy Object Identifier (OID) (see also the following sidebar) in its certificate's CertificatePolicies extension. This OID will allow the user of the certificate to reject certificates that were issued under a policy to which the user does not adhere. You can add more detailed policy information by including a URL pointer to the CPS, or a short text notice in the certificate extensions.

More information about certificate policies and CPSs is available in documents you can download from the following Web sites:

- RFC 2527 (also known as PKIX Part 4), "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," available from <http://www.ietf.org>
- Entrust white paper, "Certificate Policies and Certification Practice Statements," available from <http://www.entrust.com/resources/pdf/cps.pdf>

Defining Your PKI Topology

When you design your Windows PKI topology, you must do the following, which we discuss throughout the next subsections:

- *Decide on the number of CAs.* Your organization may require multiple CAs for scalability, business, geographical, CA policy, or political reasons.
- *Choose a PKI trust model.* Windows Server 2003 PKI supports the hierarchical, networked, hybrid, and constrained trust models.
- *Map the trust model to the Windows domain and site model.* The PKI trust model is totally independent of the Windows domain trust model. A single CA can span multiple domains, and a single domain can host multiple CAs.
- *Define relationships with external CAs or PKIs.* To provide PKI interoperability between and among different PKIs, Windows Server 2003 PKI supports cross-certification and certificate trust lists (CTLs). No matter which of these two you use, you must always define a trust policy. Will the trust be unidirectional or bidirectional? What will be the constraints of the trust?

Individual CA Specifications

CAs are key PKI components, so you need to spend enough time to create a detailed design for each individual CA. Part of the CA parameters is set during the installation; another part is set after

installation, in the CA configuration phase. Table 6-2 shows the different installation and configuration options you must consider in this context. In addition to these installation and configuration options, you must consider CA hardware sizing.

Table 6-2 CA installation and configuration options

Preliminary Planning	During CA Installation	As Part of CA Configuration
CA hardware sizing	CA role (root, subordinate, standalone, enterprise)	Revocation policy
CA architecture (exit and policy modules)	CA key and certificate properties	Supported certificate types (certificate templates)
Offline CA?	CA naming conventions	Certificate characteristics
Advanced CA private-key protection—Hardware Security Module (HSM)	CA data-storage locations	Enrollment policy (identification options, who can enroll for what)
	Reference to CPS	Recovery agent configuration
	CA X.509 certificate extensions	CA server hardening

Preliminary Planning

Before a CA is installed, you must make sure that you think about the CA hardware sizing, its architecture (are special exit and/or policy modules required?), whether the CA will be online or offline, and whether you will provide advanced CA private-key protection. Next we will discuss only CA hardware sizing and the concept of an offline CA. You can review the discussion of the other topics in previous chapters of this ebook.

CA Hardware Sizing

Table 6-3 provides some hardware sizing guidelines for a Windows Server 2003 CA. Microsoft claims that the scalability of the Windows Server 2003 CA is unlimited. Microsoft tested Windows Server 2003 PKI on a single four-processor, Intel-based computer, issuing more than 35 million certificates.

Table 6-3 Hardware sizing guidelines

Hardware Parameter	Comment
Processor	This is the most important CA resource. A powerful state-of-the-art CPU is strongly advised. Multiple CPUs will also enhance CA performance.
Memory	Microsoft recommends 512 MB; 256 MB is a minimum.
Disks	<ul style="list-style-type: none"> • RAID configuration is advisable. • Use separate physical disks for CA database and log files. • As for the CA database size, each issued certificate requires approximately 16 KB, and each archived private key requires approximately 4 KB.

Offline CAs

To minimize the risk of CA private-key compromise, you might want to set up offline CAs. Within a certificate hierarchy, for example, it is advisable to take the nonissuing CAs (root CAs and intermediate CAs) offline. Making a CA an offline CA can include different scenarios, such as the following:

- Take the CA off the network.
- Protect the CA from the rest of the network by putting it behind a firewall or a router.
- Shut down the CA service.
- Shut down the machine that hosts the CA.
- Install the CA on a standalone Windows server, and set it up as a standalone CA.
- Remove a CA server's hard disk, and store it in a vault to which only a limited number of people have access.
- Provide strong CA private-key protection by storing the CA's private key on a Hardware Security Module (HSM).

You must bring an offline CA online to issue certificates and CRLs, and every time the CA's certificate must be renewed. PKI users also must be able to access an offline CA's CRLs and CA certificate using CRL distribution points (CDP) and Authority Information Access (AIA) certificate pointers. When setting up an offline CA, you must make sure that the CDP and AIA pointers of both the CA certificate and all the certificates the CA issues refer to an online location. We explain CDP and AIA configuration in more detail later in this chapter.

When the offline CA is not connected to the network, you can use the following procedure to obtain a certificate for the new subordinate CA:

- During the subordinate CA installation, select "save the request to a file." The Microsoft Certificate Services installation wizard will inform you that the installation is incomplete. Put the request file (*.req) on a floppy disk, and transport the floppy to the offline CA.
- Bring the offline CA online. Open the subordinate CA's request file from the offline CA, and copy the text starting with "BEGIN NEW CERTIFICATE REQUEST" and ending with "END NEW CERTIFICATE REQUEST." Paste this text into the "Submit a saved request" page (Advanced certificate request) of the offline CA's Web enrollment interface. Submit the request, and save the newly generated certificate to the floppy disk. In Windows Server 2003, you also can submit a certificate-request file to a CA from the MMC Certification Authority snap-in: Right-click the CA object, and select All Tasks\Submit new request.... This action will let you select the certificate-request file from some file system location.
- Transport the floppy to the subordinate CA. Then, from the Certification Authority snap-in, right-click the CA object and select "Install CA certificate." The CA certificate will be installed, and the subordinate CA service will be started.

CA Installation Options

During CA installation, you need to think about the following CA-related parameters: the CA role (enterprise or standalone, root or subordinate), the CA's keys and certificate properties, the CA naming conventions, the CA X.509 certificate extensions, and the CA's database specifications. In the following sections, we cover only the topics that we did not discuss in earlier chapters of this ebook.

CA Role

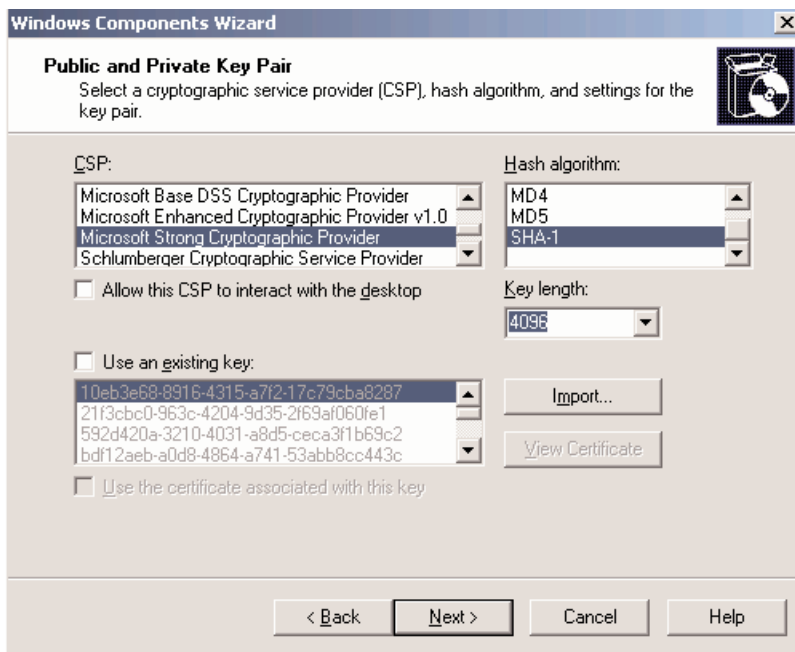
The very first screen of the CA installation wizard will ask you whether you want to install the CA as a standalone CA or enterprise CA, or as a root CA or subordinate CA. We discussed these installation options in detail in Chapter 3.

CA Keys and Certificate

Next during CA installation, you must choose the CA key length, the Cryptographic Service Provider (CSP), and the hash functions the CA will use for its cryptographic operations (as illustrated in Figure 6-3). You can choose these options only if you have checked the “Use custom settings to generate the key pair and CA certificate” CA installation option.

Figure 6-3

The CA key and certificate options during CA installation



When you are installing a root CA, you can also set the lifetime of its certificate—you can do this from the *Validity period* field in the CA Identifying Information screen shown in Figure 6-4.

Figure 6-4
Certificate lifetime and key length in a typical PKI hierarchy

The screenshot shows the 'CA Identifying Information' dialog box from the Windows Components Wizard. The title bar reads 'Windows Components Wizard'. The main title is 'CA Identifying Information' with the subtitle 'Enter information to identify this CA.' There is a CD-ROM icon in the top right corner. The dialog contains the following fields and controls:

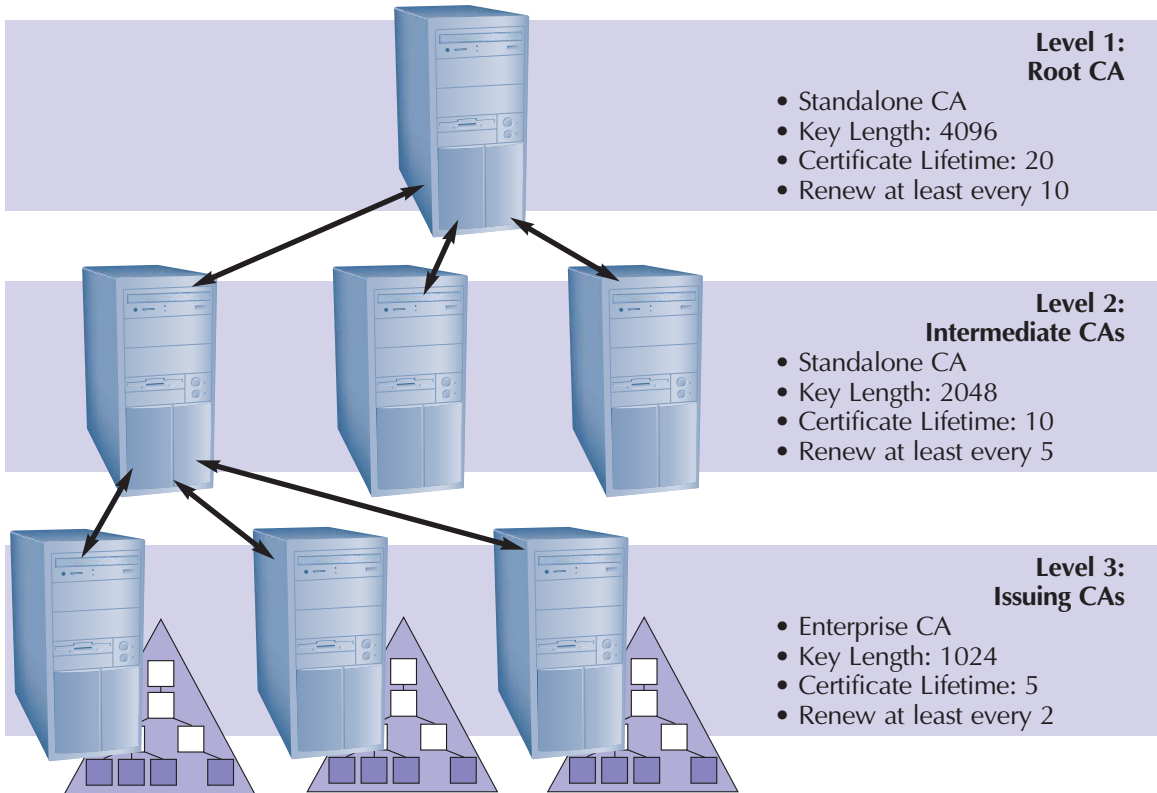
- Common name for this CA:** A text box containing 'Research Root CA'.
- Distinguished name suffix:** A text box containing 'DC=dc,DC=net'.
- Preview of distinguished name:** A text box containing 'CN=Research Root CA,DC=dc,DC=net'.
- Validity period:** A numeric input field with '5' and a dropdown menu set to 'Years'.
- Expiration date:** A text field showing '9/24/2008 4:18 AM'.
- Navigation buttons:** '< Back', 'Next >', 'Cancel', and 'Help'.

When you are installing a subordinate CA, the Validity period field says “Determined by parent CA.” In that case, the subordinate CA certificate’s lifetime is dependent upon a V2 certificate template setting (if the parent CA is an enterprise CA) or the registry key value (if the parent CA is a standalone CA). We explained how to change these settings in Chapter 3.

Figure 6-5 gives an example of how the CA certificate lifetime, key lifetime, and key length might be defined for different CAs in a PKI hierarchy. Notice that the deeper you go in the certification hierarchy, the shorter the certificate lifetime, key lifetime, and key length become.

A primary point to remember is that the CA is the heart of your security system, and if its private key is compromised, so is the entire PKI. Protect against attacks by choosing the longest key possible—at least 1,024 bits—and by storing the CA private key in a secure place. We discussed secure private-key storage in Chapter 2. Figure 6-4 shows you the screen on which you specify the CA name and certificate lifetime (Validity period) during the CA installation.

Figure 6-5
CA naming and certificate-lifetime options



CA Naming Conventions

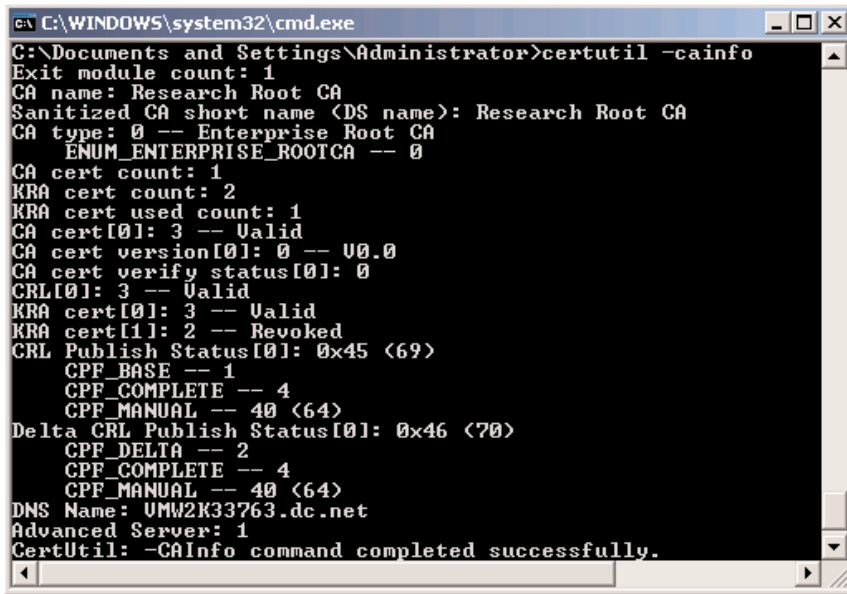
The CA installation wizard prompts you for the CA's identification information: the CA common name and distinguished name suffix. Make sure you agreed on the naming conventions before you start the installation. The naming choices you make during installation not only affect the CA, but also are reflected in the CA's common name that is stored in AD and in every certificate the CA issues. Make sure the CA common name is a short name, user- and administrator-friendly. For example, you certainly shouldn't choose a CA common name that's the CA's Fully Qualified Domain Name (FQDN).

Besides the CA's common name, Windows PKI generates a sanitized CA name, which is the short CA name, not including any non-ASCII characters and ASCII punctuation characters. The sanitized name is needed for file names, key container names, and AD object names that cannot handle a CA name that includes special characters. Sanitized names are limited to 64 characters; if a CA's name is longer than 64 characters, it is truncated and appended with a hash that is calculated over the truncated part. This name also is referred to as the CATruncatedName in Microsoft documentation.

To retrieve a CA's sanitized name, run `certutil` with the `-cainfo` switch. As Figure 6-6 shows, this command also brings up other interesting CA configuration information.

Figure 6-6

Using certutil to check the CA's sanitized name



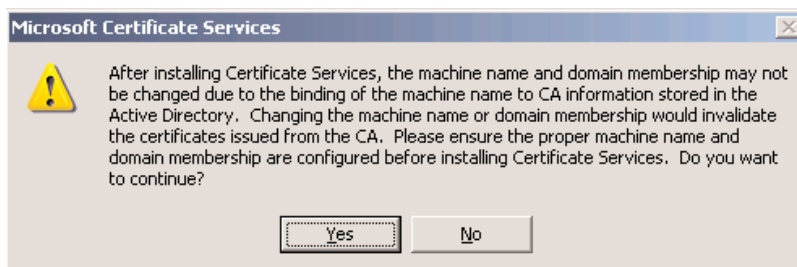
```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator>certutil -cainfo
Exit module count: 1
CA name: Research Root CA
Sanitized CA short name (DS name): Research Root CA
CA type: 0 -- Enterprise Root CA
        ENUM_ENTERPRISE_ROOTCA -- 0
CA cert count: 1
KRA cert count: 2
KRA cert used count: 1
CA cert[0]: 3 -- Valid
CA cert version[0]: 0 -- U0.0
CA cert verify status[0]: 0
CRL[0]: 3 -- Valid
KRA cert[0]: 3 -- Valid
KRA cert[1]: 2 -- Revoked
CRL Publish Status[0]: 0x45 (69)
  CPF_BASE -- 1
  CPF_COMPLETE -- 4
  CPF_MANUAL -- 40 (64)
Delta CRL Publish Status[0]: 0x46 (70)
  CPF_DELTA -- 2
  CPF_COMPLETE -- 4
  CPF_MANUAL -- 40 (64)
DNS Name: UMW2K33763.dc.net
Advanced Server: 1
CertUtil: -CAinfo command completed successfully.
  
```

Once you have installed a CA, you can change neither the CA server's name nor its Windows domain membership. To change the server name or domain membership after installation, you must uninstall Certificate Services, change the server's name or domain membership, and then reinstall Certificate Services. The CA installation program warns you about this problem, as Figure 6-7 illustrates.

Figure 6-7

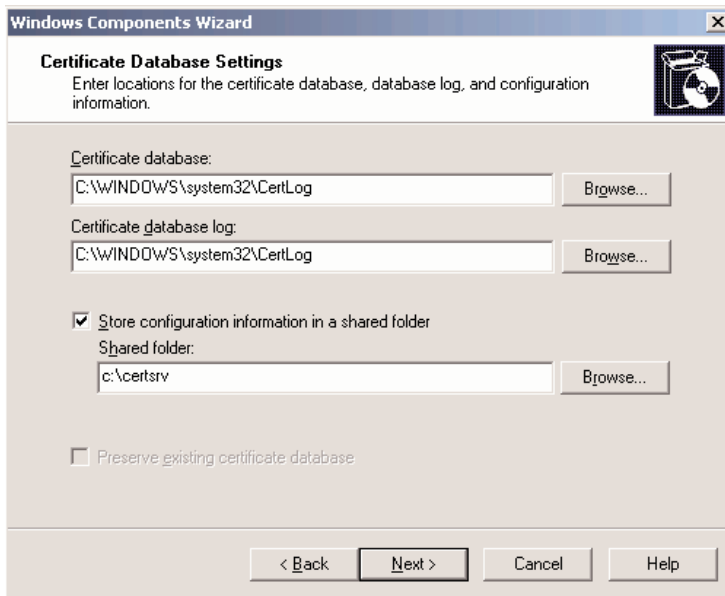
CA installation warning



The CA Database

The CA installation wizard lets you specify the location of the CA database (which is also referred to as the certificate database) and its log files (as illustrated in Figure 6-8).

Figure 6-8
CA database installation options



You also will be asked whether you want to store the CA's configuration information on the file system. When you check this option, the CA installation program will copy the CA's naming information and the CA's certificate to the file system. The configuration directory is automatically shared as certconfig.

The Windows CA database is a JET Blue database. The database has been designed to support an unlimited number of certificates, and it uses the same database engine—the Extensible Storage Engine (ESE) (esent.dll)—that is used in Microsoft Exchange and AD. Just as for any other JET database, a best practice is to split the database and its log files across different physical disk drives. By default, all CA database files are located in the CertLog subdirectory of the system directory. To find out the location of your CA database files, type

```
certutil -databaselocations
```

at the command prompt, or go to the Storage tab in the CA properties dialog box, available from the Certification Authority snap-in. Table 6-4 lists the CA database files.

Table 6-4 Windows Certificate Server database files

Database File	Goal
<CA name>.edb	The CA store
edb.log	The transaction log file for the CA store
res1.log	Reservation log file, to store transactions if disk space is exhausted
res2.log	Reservation log file, to store transactions if disk space is exhausted
edb.chk	Database checkpoint file
tmp.edb	Temporary CA store

The CA database layout used in Windows Server 2003 PKI is different from the layout that was used in previous releases. When you upgrade from Windows 2000 to Windows Server 2003, the database format is automatically converted. You can look at the layout of the CA database by typing

```
certutil -schema
```

at the command prompt.

Other CA Installation Options

During CA installation or certificate renewal, you also can add several customized X.509 extensions and properties to the CA certificate:

- CDP extensions
- Cross-certificate distribution-point extensions
- AIA extensions
- Extended key usage (EKU) extensions
- Basic constraint extensions
- Issuance-policy constraint extensions
- Application-policy constraint extensions
- Name-constraint extensions
- CA certificate-renewal settings
- CRL and delta CRL validity settings

To set these certificate extensions and properties, you must create a policy-statement file called `capolicy.inf` and store it in the Windows system directory before the installation of the CA. Chapter 3 contains a sample `capolicy.inf` file.

CA Configuration Options

Once a CA is installed, you must configure it. In the configuration phase, you must consider revocation settings, AIA settings, policy and exit-module properties, certificate template settings, delegation of administrative control, identification options, key recovery agent (KRA) settings, and CA server hardening. Again, in the following sections we do not come back to topics that we discussed extensively in previous chapters—for example, CA identification options, enrollment interfaces, certificate templates, and KRA settings.

Revocation Settings

In a Windows PKI design, you must think about the following revocation-related parameters: the CRL and delta CRL lifetime and publication interval, and the number and type of CDPs.

All of these parameters are CA-dependent and can be configured once for each CA. If your PKI-enabled applications have different CRL or delta CRL requirements, you can install multiple CAs. For example, you can install one CA with a short publication schedule, used for an application with high security requirements, and another one with a longer schedule, used for an application with lower security requirements.

CRL and Delta CRL Lifetime and Publication Interval You can set the CRL and delta CRL publication intervals from the Properties dialog box of the Revoked Certificates container in the Certification Authority snap-in or from the registry (as outlined next). In the Certification Authority snap-in, you cannot disable automatic CRL publication; you can do this, however, for delta CRLs. You can disable automatic CRL publication from the registry by setting the CRLPeriod-Units parameter to 0. For delta CRLs, you must set the CRLDeltaPeriod-Units parameter to 0. When you disable automatic CRL or delta CRL publication, you must fall back to manual publication (which we explained in Chapter 5).

It is a best practice to disable delta CRL publication and set a long CRL publication interval for offline CAs. In that case, the administrative overhead, or publishing CRLs and delta CRLs, is not outweighed by the number of revoked certificates.

If a CA cannot publish its CRL on time, the revocation information is not updated, and the old CRL will expire. Most PKI-enabled applications consider a certificate invalid if the CRL has expired or is unavailable. This is why a CRL must at least be valid for the time it takes for a CA recovery in case of a hardware or software failure. For example, a one-hour CRL publication interval is very likely not enough to perform a complete CA hardware and software restoration. This situation also explains why you must bring an offline CA online at regular intervals for CRL publication. See also “Publishing the CRL of an Offline CA” later in this chapter.

In Windows PKI, the CRL lifetime and publication interval are, although closely related, not the same. The CRL lifetime is derived from the publication interval and is by default 10 percent longer than the publication interval. This differential allows Windows PKI to deal with the replication delay of CRLs that are published in AD. You can set the CRL overlap in the registry of a CA server using the CRLOverlapPeriod, CRLOverlapUnits, and Clockskewminutes parameters. These parameters are located in the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\<CA Name>\ registry container.

The following example illustrates the use of the various parameters. Imagine you must deal with an AD replication delay of four hours. To deal with this delay, you can set the CRL overlap period to five hours. The following are the registry settings you would use in this example. These settings result in a CRL lifetime of one week, five hours, and 10 minutes, which is the sum of the CRLPeriod, CRLOverlapPeriod, and Clockskewminutes parameters. The CRL will be published every week, as specified in the CRLPeriod and CRLPeriodUnits parameters.

<i>CRLPeriod</i>	<i>REG_SZ = Weeks</i>
<i>CRLPeriodUnits</i>	<i>REG_DWORD = 1</i>
<i>CRLOverlapPeriod</i>	<i>REG_SZ = Hours</i>
<i>CRLOverlapUnits</i>	<i>REG_DWORD = 5</i>
<i>ClockSkewMinutes</i>	<i>REG_DWORD = a</i>

<i>CRLDeltaPeriod</i>	<i>REG_SZ = Days</i>
<i>CRLDeltaPeriodUnits</i>	<i>REG_DWORD = 1</i>
<i>CRLDeltaOverlapPeriod</i>	<i>REG_SZ = Minutes</i>
<i>CRLDeltaOverlapUnits</i>	<i>REG_DWORD = 0</i>

Setting the *CRLOverlapUnits* parameter in the registry to 0 activates the default algorithm—CRL lifetime is 10 percent longer than the publication interval—for the calculation of the CRL overlap. The same default interval is true for delta CRL parameters. In this example, the default algorithm has been activated for delta CRLs.

CRL Distribution Points Windows PKI supports three CDP types: Lightweight Directory Access Protocol (LDAP), file system, and HTTP-based CDPs. CDPs are used for both CRL and delta CRL distribution. For CRL downloads within your AD infrastructure, LDAP CDPs are the best choice. If you share PKI-enabled applications and certificates with external entities that do not have access to your AD, or with entities that are not using a Windows OS, consider alternative CDP locations, such as Web pages, in which case you use HTTP-based CDPs. Make sure that these CDPs do not reveal internal namespaces to the external entities.

The following is a procedure for publishing the CRL of an offline CA:

- Bring the offline CA online.
- Start the Certification Authority snap-in and publish the CRL to the local file system.
- Copy the CRL to a floppy disk or another removable medium.
- Shut down the CA.
- Publish the CRL at the different CDPs. For file-system and HTTP CDPs, copy the CRL from the floppy disk or other removable medium to the appropriate file-system location. For LDAP CDPs, copy the CRL to the file system, and then use *certutil*, together with the *-dspublish* switch to publish the CRL to AD.

To ensure revocation information redundancy, and to make CRLs available through more than one location, it is a best practice to define multiple CDP types (LDAP, HTTP, and so forth). An AD-rooted LDAP CDP automatically provides redundancy (but only on the AD level) because the CRL is replicated in the AD to all domain controllers in the forest. Also, an LDAP CDP refers to a location in the AD configuration-naming context, which is available on all AD domain controllers (DCs). To provide the same level of redundancy with HTTP CDPs, add a virtual Web server name that points to several physical Web servers for the CDP.

Root CA certificates must have an empty CDP. The CDP point is always defined by the certificate issuer (because this is also the entity that is issuing the CRL). Because the root's certificate issuer is the root CA itself, it does not make sense to include a CDP in the root CA certificate.

The CDPs of an offline CA must point to a location different from the server that is hosting the offline CA. Otherwise, users would never be able to download an offline CA's CRLs.

To make these CDP changes to a root CA's and an offline CA's certificate, you must define the CDP settings in a *capolicy.inf* configuration file and make this file available on the CA machine during the CA installation. We explained the *capolicy.inf* file and how to use it in Chapter 3.

The CDPs of the certificates a CA issues are configured in the Extensions tab of the CA Properties dialog box, which is available from the Certification Authority snap-in. You can also change these settings from the command line using the following certutil command:

```
certutil -setreg CA\CRLPublicationURLs "<List of CDPs>"
```

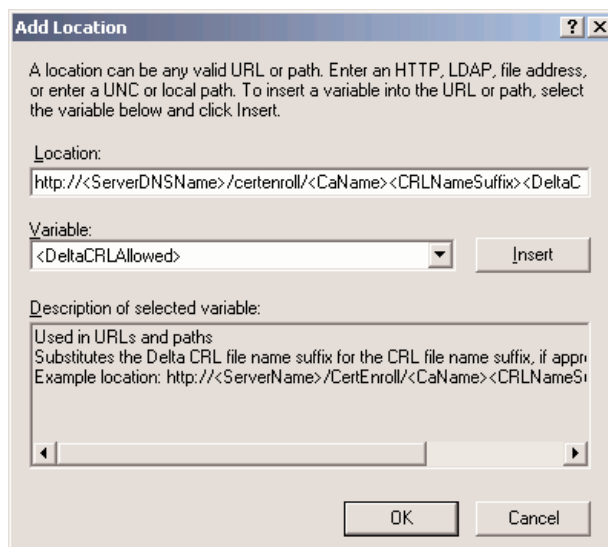
To define CDPs, you can use text string variables that are formatted using the replaceable parameter syntax. Microsoft added these variables to make the life of a CA administrator easier. The variables, also referred to as *replacement tokens*, are defined in Table 6-5.

Table 6-5 Replaceable parameter syntax

Text String Variable	Number	Variable Value
<ServerDNSName>	%1	DNS name of CA
<ServerShortName>	%2	NetBIOS name of CA
<CaName>	%3	Name of CA
<CertificateName>	%4	Certificate name
<ConfigurationContainer>	%6	Location of AD configuration container
<CaTruncatedName>	%7	Sanitized name of CA
<CRLNameSuffix>	%8	CRL base file name and renewal extension
<DeltaCRLAllowed>	%9	Substitutes delta CRL name suffix for CRL name suffix
<CDPObj ectClass>	%10	Used in LDAP URLs for CDP extension
<CAObj ectClass>	%11	Used in LDAP URLs for AIA extension

In Windows Server 2003 PKI, Microsoft offers a new interface (illustrated in Figure 6-9) that uses the replacement tokens to facilitate the creation of CDPs.

Figure 6-9
Defining CDPs using the Replaceable Parameter Index



In the registry, these tokens are translated into number variables. You must use the number variables when you are using the replacement tokens in a capolicy.inf configuration file or in a batch file. You can use the same tokens for AIA and cross-certificate distribution-point definition.

If an offline CA is not connected to the network and you provide an AD-rooted CDP, you must manually change the value of the %% replacement token. The %% replacement token holds the location of the AD configuration container. You can make the change by typing

```
certutil.exe -setreg ca\DSConfigDN <path to AD configuration naming context>
```

For example,

```
certutil.exe -setreg ca\  
DSConfigDN CN=Configuration,DC=mydomain,DC=com
```

The following are examples of an LDAP, a file system, and an HTTP CDP as they are defined in the CA properties (using the replaceable parameter syntax), and of how these CDPs will show up in the CDP certificate extension of a certificate issued by a CA named ResearchCA located on a machine called myserver. The HTTP CDP is published on a Web server called mywebserver.mydomain.net.

```
Ldap:/// CN:<CATruncatedName><CRLNameSuffix>,CN=<ServerShortName> ,CN=CDP,CN=Public Key  
Services,CN=Services,<ConfigurationContainer>,<CDPObject Class>
```

will show up as

```
URL=Ldap:/// CN=ResearchCA,CN=Myserver,CN=CDP,CN=Public%20Key%20Servi  
ces,CN=Services,CN=Configuration,DC=mydomain,DC=net?cert  
ificateRevocationList?base?objectClass=cRLDistributionPoint  
File://\<ServerDNSName>\CertEnroll\ <CaName><CRLNameSuffix><DeltaCRLAllowed>.crl
```

will show up as

```
URL=file://\mywebserver.mydomain.net/CertEnroll/ResearchCA.crl  
Http://<ServerDNSName>/CertEnroll/ <CaName><CRLNameSuffix><DeltaCRLAllowed>.crl
```

will show up as

```
URL=http://mywebserver.mydomain.net/CertEnroll/ResearchCA.crl
```

Other Revocation-Related Settings By default, a Windows CA automatically removes expired certificates from a CRL. In some PKA scenarios, maintaining expired certificates on the CRL is desirable. To do so, you can use the following certutil command:

```
certutil -setreg ca\CRLFlags +CRLF_PUBLISH_EXPIRED_CERT_ CRLS
```

By default, a Windows CA maintains expired CRLs in the CA database and in the AD. Doing this is a best practice for long-term validation and auditing purposes. You can remove expired CRLs from the CA database by using the certutil command listed below. This command applies only to the very last CRL the CA publishes for a given CA certificate.

```
certutil -setreg ca\CRLFlags + CRLF_DELETE_EXPIRED_CRLS
```

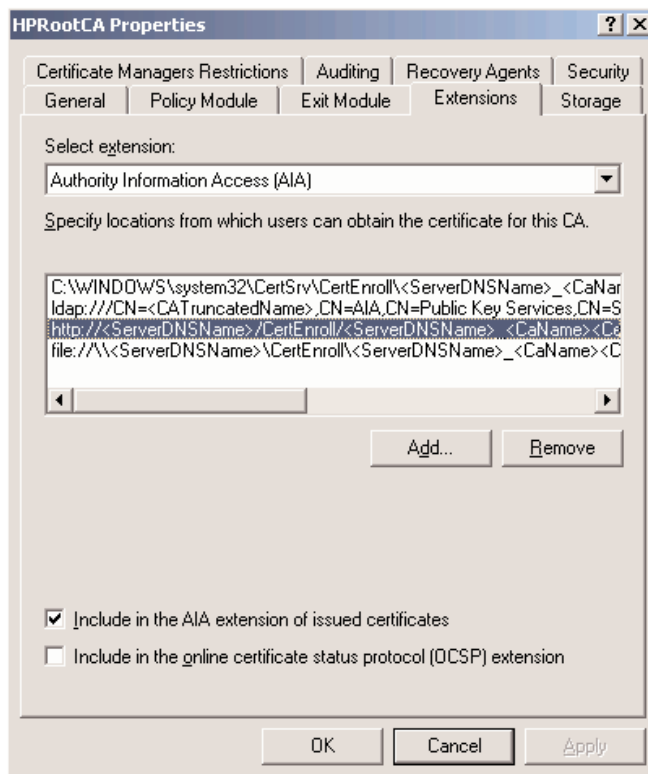
AIA Settings

A certificate's AIA fields hold pointers to storage locations for CA certificates. As explained in Chapter 5, these settings play an important role in the certificate validation process.

For the certificates a CA issues, the AIA settings—like the CDP settings—are configured from the Extensions tab in the CA Properties dialog box, available from the Certification Authority snap-in (illustrated in Figure 6-10).

Figure 6-10

Configuring AIAs from the CA Properties dialog box



You can also use the following certutil command line:

```
certutil -setreg CA\CACertPublicationURLs "<list of AIAs>"
```

As for CDPs, you can specify the AIA settings in a CA's proper certificate in the capolicy.inf file; we explained how to do this in Chapter 3. In any case, when configuring AIA settings, you can also use the replaceable parameter syntax and the replacement tokens as we defined them earlier.

Also for CDPs, the AIAs of the certificates an offline CA issues must point to a location different from the server that is hosting the offline CA. Otherwise, users would never be able to download an

offline CA's certificate. If the offline CA is not connected to the network and you provide an AD-rooted AIA, make sure you change the value of the %6 replacement token (location of the AD configuration container) by typing

```
certutil.exe -setreg ca\DSConfigDN <path to AD configuration naming context>
```

For example,

```
certutil.exe -setreg ca\  
DSConfigDN CN=Configuration,DC=mydomain,DC=com
```

An AIA storage location can hold more than a single CA certificate. An LDAP-rooted AIA storage location can be used to download, at once, all CA certificates available from the AIA. This option is possible because, in the AD, AIA CA certificates are stored in a multivalued attribute of a CA object called caCertificate. When you are dealing with an HTTP-rooted or file system-rooted AIA storage location, only a single CA certificate can be downloaded at once. This limitation exists because HTTP and file-system AIA pointers must point to individual CA certificates.

Other Certificate Characteristics

The bulk of the certificate characteristics (with the exception of the CDP and AIA extensions) can be configured from the MMC Certificate Templates snap-in. Remember that this option is true only for version 2 certificate templates. We covered certificate templates and their properties in Chapter 2.

Two very important certificate characteristics are the certificate lifetime and the renewal period. When planning for these characteristics, you must consider the following:

- *The trust your organization has in the certificate subjects.* If you are issuing certificates to users of your corporate extranet, the certificate lifetime should be shorter than the lifetime when you are issuing certificates to users of your corporate intranet. Generally, the level of trust that an organization has in its internal users is higher than the level of trust it has in the external users of the corporate IT infrastructure.
- Certificate lifetime has an impact on the number of certificate renewal requests that are sent across your network. Environments with limited network bandwidth (e.g., when users in a remote site are connecting to a CA across a slow WAN), can be a reason to lengthen certificates' lifetimes.

CA Administrative Delegation and Role Separation

Windows Server 2003 PKI provides a role-based administration model. Also, it allows for role and task separation between and among the different CA administrators. As we explain next, this role-based model piggybacks on the Windows access-control model (for permissions and user rights). This new administration model is in line with the role definitions defined in the *Certificate Issuing and Management Components (CIMC) Family of Protection Profiles Version 1.0*. You can download this standards document from http://csrc.nist.gov/pki/documents/CIMC_PP_20011031.pdf.

To assign a PKI role to a user or group, you must assign the role's corresponding security permissions or user rights to the user or group. You can set the permissions from the Security tab in the Properties dialog box of the CA object (accessible from the Certification Authority snap-in), as illustrated in Figure 6-11. You can set user rights from the MMC Group Policy Object snap-in.

Figure 6-11
Setting CA object permissions

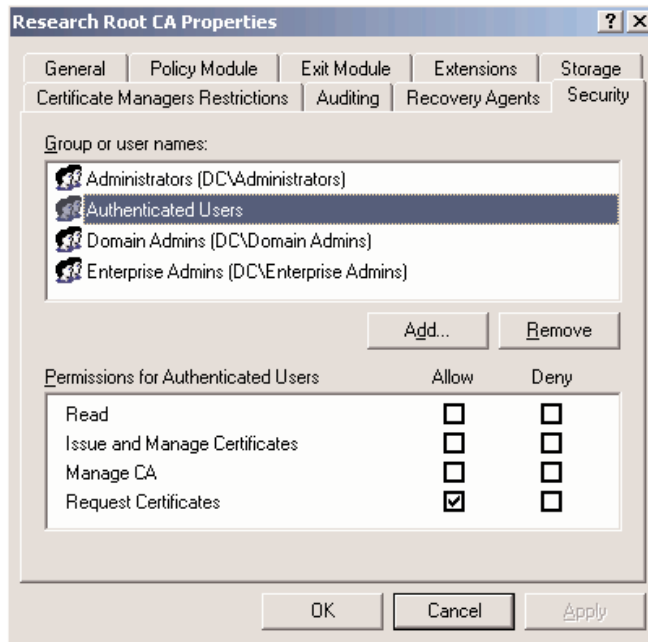


Table 6-6 shows the administrative roles and their associated permissions/user rights available in Windows Server 2003 PKI.

Table 6-6 Windows Server 2003 PKI administrative roles

Administrative Role (Certificate Issuing and Management components [CIMC] Equivalent)	Associated Permissions— User Rights	Meaning
CA Administrator	Manage CA permission.	Configure and maintain the CA. This includes the ability to assign all other CA roles and renew the CA certificate.
Certificate Manager (Officer)	Issue and manage certificates' CA permission.	Approve certificate enrollment and revocation requests.
Backup Operator (Operator)	Back up files and directories and restore files and directories user rights.	Perform system backup and recovery.
Auditor (Auditor)	Manage auditing and security logs user right.	Configure, view, and maintain audit logs.
Enrollees	Request certificates' CA permission.	Request certificates from the CA. Enrollees are clients who are authorized to make such requests.
Read	Read CA permission.	Read records from the CA database. For authorized entities.

Table 6-7 shows the tasks associated with every administrative role.

Table 6-7 Windows Server 2003 PKI administrative roles and associated tasks

Activity	Local Administrator	CA Administrator	Certification Manager	Backup Operator	Auditor
Install CA	X	—	—	—	—
Configure policy and exit module	—	X	—	—	—
Stop and start the Certificate Services service	—	X	—	X (stop only)	—
Configure extensions	—	X	—	—	—
Configure roles	—	X	—	—	—
Renew CA keys and certificates	X	—	—	—	—
Define key recovery agents (KRAs)	—	X	—	—	—
Configure Certificate Managers restrictions	—	X	—	—	—
Delete single row in database	—	X	—	—	—
Delete multiple rows in database	X	—	—	—	—
Enable role separation	X	—	—	—	—
Issue and approve certificates	—	—	X	—	—
Deny certificates	—	—	X	—	—
Revoke certificates	—	—	X	—	—
Reactivate certificates placed on hold	—	—	X	—	—
Enable, publish, or configure CRL schedule	—	X	—	—	—
Recover archived key	—	—	X	—	—
Configure audit parameters	—	—	—	—	X
Audit logs	—	—	—	—	X
Back up system	—	—	—	X	—
Restore system	—	—	—	X	—
Read CA database	X	X	X	—	X
Read CA configuration information	X	X	X	X	X

On a default Windows Server 2003 CA installation, the CA roles are assigned and modified by local administrators on a standalone machine, or, when the CA is part of a domain, by enterprise administrators and domain administrators.

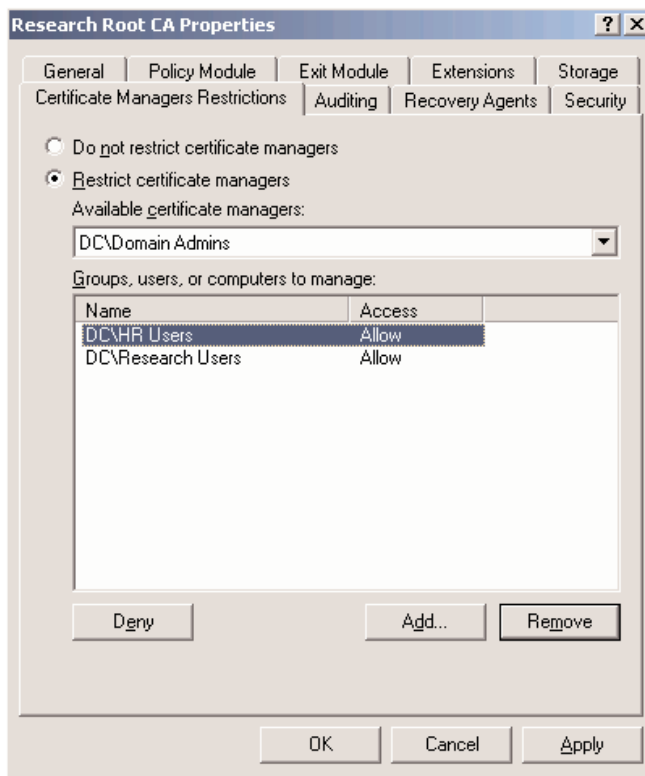
CA administrators have the Manage CA and Issue and Manage Certificates permissions on the CA object. Local administrators, enterprise administrators, and domain administrators are CA administrators by default on an enterprise CA. Only local administrators are CA administrators by default on a standalone CA. If the standalone CA is joined to an AD domain, the domain administrators are also CA administrators.

The local administrator will always have full control of the CA system and cannot be blocked from taking control of the CA. Local administrator privileges are also required for tasks such as CA key and certificate renewal, enabling role separation, and so forth.

Certificate managers are the accounts that have been assigned the Issue and Manage Certificates permission in the security properties of the CA object. Certificate Manager roles can be fine-tuned

from the Certification Authority snap-in: You can restrict which users and which groups' certificates a Certificate Manager group can manage. To do so, open the Properties dialog box of the CA object, and then go to the Certificate Managers Restrictions tab (as illustrated in Figure 6-12).

Figure 6-12
Assigning certificate managers restrictions



The Backup Operator role is based on the “Backup files and directories” and “Restore files and directories” user rights. A CA backup operator has the capability to stop the CA service, but not to start it. The Auditor role is based on the “Manage auditing and security logs” user right. By default, the local system administrator has these user rights.

The Enrollee role is based on the Request Certificates permission on the CA object. The Read role is based on the Read permission on the CA object (see Table 6.7).

Windows Server 2003 PKI also allows for strict role separation between and among the different administrative roles. If role separation is enabled, a user can be assigned only to a single role. If a user is assigned to multiple roles and attempts to perform a CA administrative operation, the operation will be denied. To enable role separation, type the following commands at the command line:

```
certutil -setreg ca\RoleSeparationEnabled 1
net stop certsvc
net start certsvc
```

To disable role separation, type the following:

```
certutil -delreg ca\RoleSeparationEnabled
net stop certsvc
net start certsvc
```

To see whether role separation is enabled, type the following:

```
certutil -getreg ca\RoleSeparationEnabled
```

CA Server Hardening

A CA's private key is the most critical element of PKI security. If a root or intermediate CA's private key is compromised, all or part of your PKI trust infrastructure falls down. The security level provided for a CA's private key also has an important impact on the amount of trust people have in the CA. This is why it is so important to store the CA's private key securely, to keep root and intermediate CAs offline, and to harden your CA server by boosting its levels of physical, logical, communications, and organizational security.

- *Physical security.* Install Certificate Servers on computers that are located in secure areas, with physical access control and adequate protection against fire, power loss, and other disasters.
- *Logical security.* In a Windows Server 2003 environment, logical security depends on the quality of the operating system's authentication, access control, and auditing system.
 - You can provide high-quality authentication by equipping all servers with smart card readers, which provide two-factor authentication.
 - You can implement strict access-control settings on all of the CA server's resources. You must lock down the server wherever possible, and not install any unneeded services or software components. A good resource for Windows Server 2003 hardening is the Windows Server 2003 Security Guide, available from the Microsoft Web site.
 - You can use the built-in Windows and CA auditing system, and the Security Configuration and Analysis tool to audit the security- and PKI-related events on Windows Server 2003 machines.
- *Communications security.* To provide communications security for the CAs (issuing CAs or other online CAs) connected to your production network, you can install them on a separate subnet or behind a dedicated firewall or router that filters all non-PKI related traffic.
- *Organizational security.* Make the CA administrators and operators aware of the important security role of the CA. Convince them that this is not an ordinary file and print server, but instead a server that is used to secure the rest of your corporate IT environment.

CA Fault Tolerance

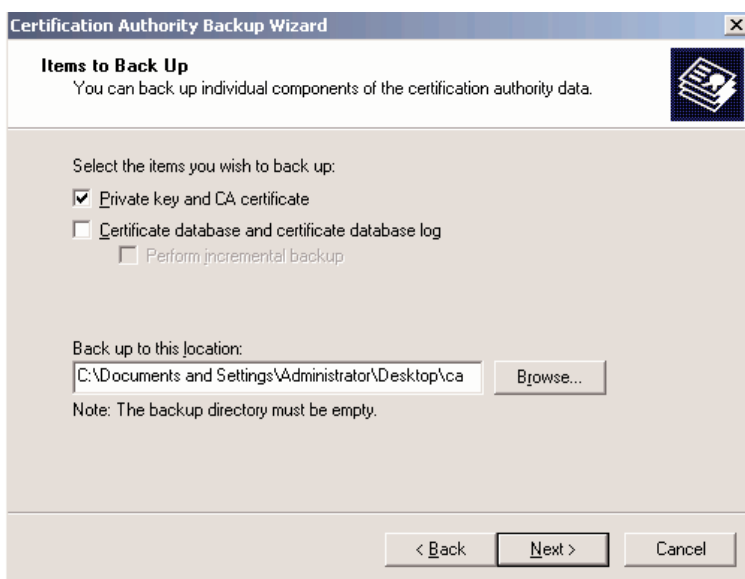
Neither Windows 2000 PKI nor Windows Server 2003 PKI supports CA service or database clustering. Also, a Windows 2000 or Windows Server 2003 machine can host only a single CA instance.

Certificate-enrollment fault tolerance is provided when using multiple enterprise CAs in an AD environment. AD-enabled PKI clients can query the AD to find out about the location of an enterprise CA and can contact another enterprise CA when one is not available.

Revocation-checking fault tolerance can be provided by making sure that you can always—independent of the CA’s availability—issue a new CRL and make it available to your PKA and PKI clients. In Windows PKI, doing this is possible thanks to a feature known as *CRL resigning*.

CRL resigning lets you issue a new CRL using the old CRL, provided you have access to a copy of the CA’s private key. You can export a CA’s private key using the Certification Authority Backup Wizard. You can access this wizard by right-clicking the CA object in the Certification Authority snap-in, and then selecting All Tasks\Back up CA. In the wizard, make sure that you select the “Private key and CA certificate” option (as illustrated in Figure 6-13).

Figure 6-13
Exporting a CA’s private key and certificate



The wizard will save the exported private key and certificate in a PKCS#12-formatted file (*.p12). When you export the CA’s private key using the procedure outlined above, make sure you store the PKCS#12 file in a highly secure place. If the CA’s private key becomes compromised, the trust of your entire PKI is compromised.

You can always retrieve the old CRL from a CDP. To resign an old CRL, use the following certutil command:

```
certutil -sign <old CRL name> <resigned CRL name>
```

Defining Public Key Policy Settings

Windows Server 2003 Group Policy Object (GPO) objects include the following PKI-related entries: Encrypting File System, Automatic Certificate Request Settings, Trusted Root Certification Authorities, Enterprise Trust, and Autoenrollment Settings. These objects are located in the Windows Settings\Security Settings\ Public Key Policies GPO container.

The Encrypting File System GPO container is used to define accounts that have the ability to recover Encrypting File System (EFS) data. The Automatic Certificate Request Settings (ACRS) container is used to define machine certificate autoenrollment settings. The Trusted Root Certification Authorities container is used to distribute trusted root CA certificates to clients. The Enterprise Trust container is used to define CTLs. The Autoenrollment Settings object is used to define both user and machine autoenrollment settings.

There is an important difference between using the ACRS container and using the Autoenrollment Settings option for defining autoenrollment settings in a GPO:

- The ACRS container can be used only to define machine certificate autoenrollment on Windows 2000, Windows XP, and Windows Server 2003 machines that are domain members. You can use this option only to autoenroll a machine for a certificate that's based on a version 1 certificate template. ACRS is basically the machine autoenrollment mechanism that was introduced in Windows 2000—and that is still supported in Windows Server 2003.
- The Autoenrollment Settings object can be used to define both machine and user certificate autoenrollment. Unlike ACRS, this object cannot be used to define autoenrollment on Windows 2000 (Professional or Server) machines. The Autoenrollment Settings object can be used only to autoenroll a user or machine for a certificate that's based on a version 2 certificate template. This is the autoenrollment mechanism Microsoft introduced in Windows Server 2003.

Table 6-8 shows on which GPO level (domain, site, OU, or local) and for which GPO portion (user or computer) the PKI-related GPO settings are available. GPO settings that are defined on the machine level are shared with all users logging on to that machine and all services running on it.

Table 6-8 PKI-related GPO settings

	GPO Level	User GPO Portion	Machine GPO Portion
Encrypting File System	Domain	—	X
	Site	—	X
	OU	—	X
	Local	—	X
Automatic Certificate Request	Domain	—	X
	Site	—	X
	OU	—	X
	Local	—	—
Trusted Root CAs	Domain	—	X
	Site	—	X
	OU	—	X
	Local	—	—
Enterprise Trust	Domain	X	X
	Site	X	X
	OU	X	X
	Local	—	—
Autoenrollment	Domain	X	X
	Site	X	X
	OU	X	X
	Local	—	—

Conclusion

This chapter has provided a step-by-step overview of the planning and design process for a Windows-rooted PKI. As we have mentioned multiple times in this ebook, careful planning and design of a PKI is paramount—independently of whether you're building an internal PKI, whether you're outsourcing your PKI design and operation to an external company, or whether you're buying certificates from a commercial CA. In the next chapter, we explore some of the Windows PKI-enabled applications and how you can configure them to take advantage of the security services the PKI offers.