Results 1 - 10 for: **install certificate**

All Results

View results in another search category by clicking a link in the right column...

Show Me:

All Results

Downloads

Product Information

Support & Troubleshooting

Technical Resources

Training & Books

Partner & Business Resources

Communities & Newsgroups

Microsoft News & Corporate Information

Related Links

- Building an Enterprise Root Certification Authority in Small and Medium Businesses
- Platform SDK: Windows Installer

- Microsoft Learning Home Page

Install certificate after deleting the pending certificate request (IIS 6.0)
Install certificate after deleting the pending certificate request (IIS 6.0)
http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/53dfdb5e-6106-4d99-85bb-da199bc27c7e.mspx

Microsoft BizTalk Server 2002 - Install Certificate Services
Microsoft BizTalk Server 2002 provides tools for developing and executing integrated business processes in the form of XLANG orchestrations within and between companies. Version enhancements include event management and XML Web Services support.
http://msdn.microsoft.com/library/en-us/bts_2002/htm/bts_sdk_samp_encryption_ljxs.asp

How to install a certificate for use with IP Security in Windows Server 2003
When IP Security (IPSec) is configured to use a Certificate Authority (CA) for mutual authentication, you must obtain a local computer certificate. This article describes how to install a local computer certificate for use with IPSec from a...
http://support.microsoft.com/default.aspx?scid=kb;en-us;323342

Planning a Certificate Infrastructure to Support Client Authentication: Virtual Private Network (VPN)
Planning a Certificate Infrastructure to Support Client Authentication
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/DepKit/498313ab-aef4-42b9-b891-7e32bd622fa0.mspx

HOW TO: Install a Certificate for Use with IP Security
When IP Security (IPSec) is configured to use a certification authority (CA) for mutual authentication, you must obtain a local computer certificate. You can obtain this certificate from a third-party CA or you can install Certificate Services in...
http://support.microsoft.com/default.aspx?scid=kb;en-us;253498

Deploying a Certificate Infrastructure: Wireless
Deploying a Certificate Infrastructure
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/DepKit/9383f94c-9e19-4012-9501-7c1ea4e6fc18.mspx

Computer certificates for certificate-based authentication: Internet Authentication Service (IAS)
Computer certificates for certificate-based authentication
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/dac646dd-b8ff-46a4-9129-18584c3a02cb.mspx

How to install a trusted root CA certificate and an intermediate CA certificate on a computer that is running Microsoft Entourage 2004 for Mac on a Mac OS X 10.3 or a Mac OS X 10.2 operating system
Describes how to install a trusted root certification authority (CA) certificate and an intermediate CA certificate on a computer that is running Microsoft Entourage 2004 for Mac on a Mac OS X 10.3 or a Mac OS X 10.2 operating system.
http://support.microsoft.com/default.aspx?scid=kb;en-us;887413

Installing a Digital Certificate
Installing a Digital Certificate Language Filter: All
http://msdn.microsoft.com/library/en-us/xmlsdk/html/a36f3576-14aa-45dd-8b6d-656c507347a6.asp

Microsoft Office Assistance: Installing a root certificate
( ) Client Deployment Server Deployment Related Web Sites Worldwide Feedback Reverse Proxy Configurations for Windows SharePoint Services and Internet Security and Acceleration Server Chapter: Go Installing a root certificate For a client computer to trust the server certificates that you
http://office.microsoft.com/en-us/assistance/HA011923651033.aspx

.001 seconds

Results 1 - 10 **Next** >

Advanced Search|Search Preferences|Search Help

Search Microsoft.com for

install certificate Go

Search Microsoft.com Worldwide

Choose a different location

Didn't find it here?

Search the entire Internet using MSN Search

Install certificate after deleting the pending certificate request (IIS 6.0)

IIS stores the private key for a certificate as the pending request. Deleting the pending request deletes the association of the private key with IIS, but the private key still exists in the certificate store. To install the certificate without having the pending request available, you can use version 5.2.3718.0 of the Certutil. exe command-line tool that is available through the Certificate Services MMC snap-in in Windows Server 2003. For more information about Certutil.exe, see Certutil.exe.

Related Links

- Installing Server Certificates

Procedures

To install a Web server certificate that lacks a pending certificate request

1. Click **Start**, point to **Run**, type **cmd**, and then click **OK**.
2. Navigate to the directory where Certutil.exe is stored; by default, this is %windir%\system32.
3. Type the following command at the command prompt: **certutil -addstore my** *certnew.cer*

   where *certnew.cer* is the name of the certificate you received from the certification authority (CA). You should see the following message: **CertUtil: -addstore command completed successfully**.
4. Navigate to the directory where you stored the certificate you received from the CA. Right-click the certificate and then point to **Properties**.
5. Click the **Details** tab and select <All> in the **Show** drop-down list.
6. In the **Field** list, select **Thumbprint** to display its value in the view pane.
7. Select the **Thumbprint** value in the view pane and then click CTRL+C.
8. Return to the command prompt window and type the following command: **certutil -repairstore my "***thumbprint***"**

where *thumbprint* is the value of the **Thumbprint** field. Be sure to type the double quotes as part of the command. If the command is successful, the following message is displayed: "Encryption test passed CertUtil: = repairstore command completed successfully."

1. Install the server certificate on your Web server.

Important

If the certutil command does not complete successfully, the following error message is displayed: "Certutil: -repairstore command FAILED: 0x80090011 (-2146893807) Certutil: Object was not found." This message indicates that the private key for the certificate does not exist in the certificate store. You cannot install the certificate you obtained from the CA. Instead, you must generate a new certificate request, obtain the new certificate, and install that new certificate on your Web server.

Top of page

Installing Server Certificates (IIS 6.0)

After you have obtained a server certificate, you can install it. When you use the Server Certificate Wizard to install a server certificate, the process is referred to as *assigning* a server certificate.

Important

You must be a member of the Administrators group on the local computer to perform the following procedure or procedures. As a security best practice, log on to your computer by using an account that is not in the Administrators group, and then use the **runas** command to run IIS Manager as an administrator. At a command prompt, type **runas /user:** *Administrative_AccountName* **"mmc %systemroot%\system32\inetsrv\iis.msc"**.

Procedures

To install a server certificate using the Web Server Certificate Wizard

1. In IIS Manager, expand the local computer, and then expand the **Web Sites** folder.
2. Right-click the Web site or file that you want, and then click **Properties**.
3. On the **Directory Security** or **File Security** tab, under **Secure communications**, click **Server Certificate**.
4. In the Web Server Certificate Wizard, click **Assign an existing certificate**.
5. Follow the Web Server Certificate Wizard, which will guide you through the process of installing a server certificate.

Note

When you use the Web Server Certificate Wizard to assign a certificate, you must specify a password before the certificate can be assigned to your Web server.

Related Information

- For information about obtaining server certificates, see Obtaining Server Certificates.
- For general information about certificates, see SSL and Certificates.

Top of page

Certutil

Updated: January 21, 2005

Certutil

Certutil.exe is a command-line program that is installed as part of Certificate Services in the Windows Server 2003 family. You can use Certutil.exe to dump and display certification authority (CA) configuration information, configure Certificate Services, back up and restore CA components, and verify certificates, key pairs, and certificate chains.

For more information about how to use Certutil.exe to perform specific tasks, see the following topics:

Top of page

BizTalk Server 2002 ~ Developer Solutions

Install Certificate Services

1. On the **Start** menu, point to **Settings**, and then click **Control Panel**.
2. Double-click **Add/Remove Programs**.
3. Click **Add/Remove Windows Components**, click **Certificate Services**, and then click **Next**.
4. Complete the steps in the Certificate Services installation wizard.

For this sample, it is sufficient to use the wizard's default selected options.

Did you find this information useful? Please send your suggestions and comments about the documentation to BizTalk Server Documentation Feedback@microsoft.com

How to install a certificate for use with IP Security in Windows Server 2003

View products that this article applies to.

Article ID	:	323342
Last Review	:	July 15, 2004
Revision	:	8.1

This article was previously published under Q323342

On This Page

SUMMARY

When IP Security (IPSec) is configured to use a Certificate Authority (CA) for mutual authentication, you must obtain a local computer certificate. This article describes how to install a local computer certificate for use with IPSec from a stand-alone Windows CA.

To obtain a local computer certificate, do one of the following:

- Obtain this certificate from a third-party CA.
- Install Certificate Services in Windows to create your own CA.

The request for the local computer certificate is requested by using HTTP. Because a local computer certificate must be used with IPSec, you must submit an advanced request to the CA to specify this.

When you are using a **Local** Certificate Authority, the CA must be set up to allow IPSEC certificates. The instructions in this article assume that you have permitted Client Authentication, IPSEC, and IPSEC (Offline Request). If you are missing these during the request, you must correctly set up your CA before you continue.

Back to the top

Install a Local Computer Certificate from a Stand-Alone Windows Certificate Authority

1. The request is a Web address that contains the IP address or name of the Certificate server, with "/certsrv" appended. In your Web browser, type the following Web address

   http://*IP address of CA*/certsrv

   where *IP address of CA* is the IP address or name of the Certificate server.
2. On the initial Welcome page of the Certificate server, click **Request a certificate**, and then click **Next**.
3. On the Choose Request Type page, click **Advanced request**, and then click **Next**.
4. On the Advanced Certificate Requests page, click **Submit a certificate request to this CA using a form**, and then click **Next**.
5. On the Advanced Certificate Request page, type your name and your e-mail name in the appropriate boxes.
6. Under **Intended Purpose**, click **Client Authentication Certificate** or **IPSec Certificate**.

   If you click **IPSec Certificate**, this certificate will only be used for IPSec.
7. Under **Key Options**, click **Microsoft Base Cryptographic Provider v1.0**, click **Signature** for **Key Usage**, and then click **1024** for **Key Size**.
8. Leave the **Create new key set** option selected (you can clear the **Container Name** check box unless you want to specify a specific name), and then click **Use local machine store**.
9. Leave all the other options set to the default value unless you have to make a specific change.
10. Click **Submit**.

    If the Certificate Authority is configured to issue certificates automatically, the Certificate Issued page appears.
11. Click **Install this Certificate**.

    The Certificate Installed page appears with the following message: "Your new certificate has been successfully installed."
12. If the Certificate Authority is not configured to issue certificates automatically, a Certificate Pending page appears and requests that you wait for an administrator to issue the certificate that was requested.

    To retrieve a certificate that an administrator has issued, return to the Web address, and then click **Check on a pending certificate**. Click the requested certificate, and then click **Next**.

    If the certificate is still pending, the Certificate Pending page appears. If the certificate has been issued, the Install This Certificate page appears.

Back to the top

Install a Local Computer Certificate from an Enterprise Windows Certificate Authority

1. The request is a Web address that contains the IP address or name of the Certificate server, with **/certsrv** appended. In your Web browser, type the following Web address

   http://*IP address of CA*/certsrv

   where *IP address of CA* is the IP address or name of the Certificate server.

2. If the computer that you are using is not logged on to the domain already, you are prompted to supply domain credentials.
3. On the initial Welcome page of the Certificate server, click **Request a Certificate**, and then click **Next**.
4. On the Choose Request Type page, click **Advanced Request**, and then click **Next**.
5. On the Advanced Certificate Requests page, click **Submit a certificate request to this CA using a form**, and then click **Next**.
6. On the Advanced Certificate Request page, click **IPSEC (Offline Request)** for the **Certificate Template** option. Restart Certificate services.
7. Open the Certificate Authority snap-in, right-click **Policy Settings**, click **New**, click **Certificate to Issue**, select **IPSec (Offline Request)**, and then click **OK**.

   Note By default, this template is not listed on an Enterprise CA.
8. Under **Key Options**, click **Microsoft Base Cryptographic Provider v1.0**, click **Signature** for **Key Usage** and then click **1024** for **Key Size**.
9. Leave the **Create new key set** option selected (you can clear the **Container Name** check box unless you want to specify a name), and then click **Use local machine store**.
10. Leave all the other options set to the default value unless you have to make a specific change.
11. Click **Submit**.

    The Certificate Issued page appears.
12. Click **Install this Certificate**. The Certificate Installed page appears with the following message:

    Your new certificate has been successfully installed.

Back to the top

Verify That the Local Computer Certificate Has Been Installed

After the certificate is installed, verify the location of the certificate by using the Certificate (Local Computer) snap-in in the Microsoft Management Console (MMC). Your certificate appears under **Personal**.

If the certificate that you have installed does not appear here, the certificate was installed as a user certificate request, or you did not click **Use local machine store** in the advanced request.

Back to the top

APPLIES TO

- Microsoft Windows Server 2003, Datacenter Edition
- Microsoft Windows Server 2003, Enterprise Edition
- Microsoft Windows Server 2003, Standard Edition
- Microsoft Windows Server 2003, Web Edition
- Microsoft Windows Server 2003, Datacenter Edition
- Microsoft Windows Server 2003, Enterprise Edition for Itanium-based Systems
- Microsoft Windows Small Business Server 2003 Standard Edition
- Microsoft Windows Small Business Server 2003 Premium Edition

Back to the top

Keywords: kbhowtomaster kbipsec kbtool kbenv kbsecurityservices KB323342

Back to the top

How to install a trusted root CA certificate and an intermediate CA certificate on a computer that is running Microsoft Entourage 2004 for Mac on a Mac OS X 10.3 or a Mac OS X 10.2 operating system

View products that this article applies to.

Article ID	:	887413
Last Review	:	November 5, 2004
Revision	:	1.0

On This Page

INTRODUCTION

This article describes how to install a trusted root certification authority (CA) certificate and an intermediate CA certificate on a computer that is running Microsoft Entourage 2004 for Mac on one of the following operating systems:

- The Mac OS X 10.3 operating system.
- The Mac OS X 10.2 operating system.

Note You must have administrative permissions on your computer to be able to use the methods that are outlined in this article.

Back to the top

MORE INFORMATION

To complete the certificate installation, you will need access to the certificate file. You can obtain the needed certificate file by using any one of the following methods:

- Copy the file to the local workstation by using removable storage media.
- Copy the file from a network share location.
- Download the file from the Web URL that is assigned in the Authority Information Access extension of the certificate or from the enrollment page for your CA certificate.

Back to the top

Install the certificate

To install the certificate, you must use the Keychain Access program. To start the Keychain Access program, double-click the certificate file. The Keychain Access program will automatically load, and it will display the **Add Certificates** dialog box.

- To install a trusted root CA certificate, follow these steps:
  1. Click the **Keychain** menu, click **X509 Anchors**, and then click **OK**.
  2. You will be prompted to authenticate through **Keychain Access**. Type your password, and then click **OK**.

     Note If **X509 Anchors** is not available in the **Keychain** menu, the certificate that you have opened is not a trusted root CA certificate. The most common file name extensions for this kind of certificate are .cer and .crt.
- To install an intermediate CA certificate, follow these steps:
  1. Click the **Keychain** menu, click **Microsoft_Intermediate_Certificates**, and then click **OK**.

Back to the top

Verify the certificate installation

To verify that the certificate is installed and that it is ready for use by Entourage 2004 for Mac, use the Microsoft Certificate Manager. To do this, follow these steps:

1. On the **Go** menu in **Finder**, click **Applications**.
2. Open the Office 2004 folder, and then open the Office folder.
3. Double-click **Microsoft Cert Manager**.
4. On the **Look for certificates of type** menu, select one of the following options:
   - Click the **Apple Trusted Root Certificate Authorities** option for an installed root CA certificate.
   - Click the **Intermediate Certificate Authorities** option for an installed intermediate CA certificate.

Verify that the newly installed certificate is in the appropriate certificate list. After you verify the location of the certificate, Entourage 2004 for Mac is ready to use the certificate for Secure Sockets Layer (SSL) communications.

Back to the top

Personal certificates for sending digitally signed and encrypted messages

Personal certificates that are obtained from a certification authority are installed into **Microsoft_Entity_Certificates** by using the steps that were discussed earlier in this article.

Back to the top

Setting up digital IDs in Entourage 2004 for Mac

After the certificates are installed, you are ready to set up Entourage 2004 for Mac to use digital IDs. To do this, follow these steps:

1. Start Entourage 2004 for Mac.
2. On the **Entourage** menu, click **Account Settings**.
3. Double-click the account that you want to set up for signing and encrypting mail.
4. Click the **Security** tab.
5. Click **Select** under **Signing Certificate**, click the digital ID or the certificate that you want to use, and then click **Choose**.

   Note The list will contain all the personal certificates that you imported into your personal keychain.
6. Repeat step 5 to select an encryption certificate.
7. Select any options that you want. Typically, the default settings are what you would want to use.
8. Click **OK**.

Back to the top

Mac OS X 10.2

Follow these steps when you import a certificate on a Macintosh computer that is running Mac OS X 10.2.

Note You must have administrative permissions on your computer to be able to follow these steps.

1. Download the certificate to your desktop.
2. Make sure that the certificate is in privacy enhanced mail (PEM) format.

   Note If the certificate is not in PEM format, use the Microsoft Certificate Manager in the Office folder to change formats. Import the certificate and then use the PEM format when you save the certificate.
3. Click **Applications** on the **Go** menu, open the Utilities folder, and then double-click the **Terminal** program.
4. Type the following commands, and press the ENTER key after each line.
   Replace *cert_filename* with the actual file name of your certificate.

   ```
   cd ~/Desktop
   cp /System/Library/Keychains/X509Anchors ~/Library/Keychains
   certtool i cert_filename k=X509Anchors
   sudo cp ~/Library/Keychains/X509Anchors /System/Library/Keychains
   ```

   Note You must enter an administrative password after you press ENTER for the last Terminal command.

Back to the top

APPLIES TO

- Microsoft Entourage 2004 for Mac

Back to the top

Keywords: kbhowto kbcertservices KB887413

Back to the top

Installing a Digital Certificate

[This topic covers a procedure for working with the XML digital signatures support first implemented in MSXML 5.0 for Microsoft Office Applications.]

After you have requested and been issued a digital certificate, you must install it on your machine.

To install a certificate to a machine

1. Verify that the certificate has been issued by visiting the certificate authority (CA) server, such as "http://*myCAServer*/certsrv", to which you have previously submitted the request.

Note:
If your certificate request is still pending, you must wait until an administrator on the CA server machine issues the certificate before you attempt this procedure. See Issuing a Certificate for a Pending Request for more information.

2. Under **Select a task**, select **Retrieve the CA certificate or certificate revocation list** and click **Next**.
3. On the **Retrieve the CA certificate or certificate revocation list** page, select the appropriate certificate (such as "Code Signing Certificate"), and click **Next**.
4. Click the **Install this certificate** link.
5. When prompted, click **Yes** to confirm installation of the certificate.

Next, you need to install the latest version of the Platform SDK and configure Visual C++ to use it.

Client Deployment
Office Resource Kit
Server Deployment
Live Communications Server
Project Server
SharePoint Portal Server
Windows SharePoint Services Technology
Microsoft Content Management Server
Microsoft Exchange Server
Related Web Sites
Product Support
Office Community
Office Developer Center
Worldwide Office Worldwide
Feedback
Comment on this Web page

Installing a root certificate

For a client computer to trust the server certificates that you have installed from a local CA, you must install the root certificate from the CA on the client computer. Follow this procedure on any client computer that requires the root certificate. Note that you can also transfer the root certificate on a medium such as a disk, and then install it on the client computer.

Install a root certificate

1. Open Internet Explorer.
2. On the **Tools** menu, click **Internet Options**.
3. On the **Security** tab, click **Custom Level** to open the **Security Settings** dialog box.
4. Under **Reset custom settings**, in the **Reset to** box, select **Medium**, and then click **OK** to close the **Security Settings** dialog box.
5. Click **OK** to close the **Internet Options** dialog box.

   Note Certificates cannot be installed when the security setting is set to **High**.

6. Browse to: http://*IP_Address*/certsrv, where *IP_Address* is the IP address of your Certification Authority Server.
7. Click **Download a CA Certificate, Certificate Chain, or CRL**.
8. On the next page, click **Download CA Certificate**.

   This is the trusted root certificate that must be installed on the ISA Server computer.

9. In the **File Download** dialog box, click **Open**.
10. On the **Certificate** dialog box, click **Install Certificate** to start the Certificate Import Wizard.
11. On the **Welcome** page, click **Next**.
12. On the **Certificate Store** page, select **Place all certificates in the following store** and click **Browse**.
13. In the **Select Certificate Store** dialog box, select **Show Physical Stores**.
14. Double-click **Trusted Root Certification Authorities**, select **Local Computer**, and then click **OK**.
15. On the **Certificate Store** page, click **Next**.
16. On the summary page, review the details and click **Finish**.

Verify that the server certificate was properly installed

1. Open MMC, and go to the Certificates snap-in.
2. Open **Certificates (local computer),** double-click the **Trusted Root Certification Authorities** node, click **Certificates**, and then verify that the root certificate is in place.

Note You can also install certificates on a computer from the MMC Certificates (Local Computer) snap-in. This provides access only to CAs in the same domain.

Printer-friendly version

Accessibility | Contact Us | Free Newsletter | Office Worldwide

Results 1 - 10 for: **install private key**

All Results

View results in another search category by clicking a link in the right column...

Show Me:

All Results

Downloads

Product Information

Support & Troubleshooting

Technical Resources

Training & Books

Partner & Business Resources

Communities & Newsgroups

Microsoft News & Corporate Information

Related Links

- Platform SDK: Windows Installer
- Microsoft Security Response Center PGP Key

Install Certification Authorities
Install Certification Authorities You must install the CA hierarchies necessary to provide the required certificate services for your organization. Certification hierarchies with Windows 2000 CAs can
http://www.microsoft.com/resources/documentation/windows/2000/server/reskit/en-us/distrib/dscj_mcs_tpuv.asp

You are no longer prompted to enter your private key password every time that the private key is accessed after you upgrade your computer to Windows XP Service Pack 2
Describes how you are not prompted to enter your private key password when strong private key protection functionality is set to high after you upgrade your computer to Windows XP Service Pack 2. You must modify the registry to resolve this issue.
http://support.microsoft.com/default.aspx?scid=kb;en-us;890062

How to back up the recovery agent Encrypting File System (EFS) private key in Windows Server 2003, in Windows 2000, and in Windows XP
Describes how to back up the recovery agent Encrypting File System (EFS) private key in Windows Server 2003, in Windows 2000, and in Windows XP.
http://support.microsoft.com/default.aspx?scid=kb;en-us;241201

Key Archival and Management in Windows Server 2003
This white paper covers best practices for private key archival and management; procedural steps in a key recovery strategy; as well as migration procedures for moving from an Exchange KMS environment to a Windows Server 2003 Certificate Authority.
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/security/kyacws03.mspx

Certificate Templates Troubleshooting: Public Key
Certificate Templates Troubleshooting
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/43881ad5-aa6b-4527-ad59-cd2218bd9934.mspx

Submit an advanced certificate request via the Web: Public Key
Submit an advanced certificate request via the Web
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/f0741bcd-b80d-4ee4-8972-ebb0ba741c0c.mspx

Submit an advanced certificate request via the Web
To submit an advanced certificate request via the Web Open Internet Explorer In Internet Explorer, connect to http://servername/certsrv, where servername is the name of the Windows 2000 Web server where the certification authority you want to access is located. Click Request a
http://www.microsoft.com/windows2000/en/advanced/help/sag_CSWprocs_reqadv.htm

Microsoft Office Assistance: Step 3: Export the Client Certificate Without a Private Key (.Cer File) for Use on Front-End Web Servers
( ) Client Deployment Server Deployment Related Web Sites Worldwide Feedback Managing Search Settings Chapter: Go Step 3: Export the Client Certificate Without a Private Key (.Cer File) for Use on Front-End Web Servers You will export two versions of the client certificate—one version
http://office.microsoft.com/en-us/assistance/HA011647831033.aspx

Microsoft Windows XP - Submit an advanced certificate request via the Web

Open Internet Explorer In Address, type http://servername/certsrv, where servername is the name of the Windows 2000 Web server where the certification authority you want to access is located. Click Request a certificate, and then click Next.
http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_cswprocs_reqadv.mspx

Requesting Certificates with the Certificate Request Wizard

Requesting Certificates with the Certificate Request Wizard You can request certificates for Windows 2000–based computers by using the Certificates console. When you right-click the Personal store for
http://www.microsoft.com/resources/documentation/windows/2000/server/reskit/en-us/distrib/dscj_mcs_hsar.asp

0.734 seconds

Results 1 - 10 **Next** >

Advanced Search|Search Preferences|Search Help

Search Microsoft.com for

renew certificate Go

Search Microsoft.com Worldwide

Choose a different location

Didn't find it here?

Search the entire Internet using MSN Search

Install Certification Authorities

You must install the CA hierarchies necessary to provide the required certificate services for your organization. Certification hierarchies with Windows 2000 CAs can include a mixture of enterprise CAs and stand-alone CAs. You can install the root CA first and then each subordinate CA in the hierarchy. For example, to create a three-level certification hierarchy, you can install CAs on servers in the following order:

1. Root CA
2. Intermediate CAs
3. Issuing CAs

However, to install the CA software on computers, you are not required to install CAs in this order. Root CAs are certified by self-signed certificates, so they do not depend on another CA to complete the installation. However, the complete installation of child CAs requires the parent CA to process the certificate request and issue the subordinate CA certificate. You can install a subordinate CA at any time, save the certificate request to a file, and submit it to the parent CA later, after the parent CA is installed and running. After parent CAs are installed and running, you can submit the certificate request file by using the Advanced Certificate Request Web pages for the parent CA. After the certificate for the child CA is issued, you can install the certificate for the child CA by using the Certification Authority console. A CA must have a valid CA certificate to start.

Although you can install CAs on domain controllers, it is not a recommended practice. To distribute the network load and prevent excessive load conditions on computers, install CAs on Windows 2000 Server–based computers that are dedicated to providing CA services. Also consider installing the Web Enrollment Support pages on separate Windows 2000 Server–based computers.

For information about installing third-party CAs and using them with Windows certification hierarchies, see the documentation for the third-party CA product.

Upgrading from Certificate Server 1.0

If you upgrade a Windows NT 4.0–based server that is running Certificate Server 1.0 to Windows 2000 Server, Certificate Server 1.0 is upgraded automatically to the new version of Certificate Services. If the CA being upgraded is using a policy module other than the default policy module for Certificate Server 1.0, it continues to use its old

policy module, which is referred to as the Legacy policy module. If the CA you are upgrading uses the default policy module that was provided with Certificate Server 1.0, the upgraded CA uses the Certificate Services stand-alone policy.

If you are not upgrading a Certificate Server 1.0 CA and, instead, are installing a separate Windows 2000 CA that is to replace the old CA, you might want to use the older policy module instead of the default policy module that is provided with Certificate Services. If you want to replace the policy module that is provided with Certificate Services with a custom policy module or a policy module developed for Certificate Server 1.0 and Windows NT 4.0, you must first register the policy module DLL file by using the **Regsrv32** command, and then select the policy module by using the Certification Authority console. For more information about using Regsv32 and selecting policy modules, see Windows 2000 Server Help and Certificate Services Help.

Creation of an Issuer Statement for the Certification Authority (Optional)

When you install a CA, you have the option of adding an issuer statement for the CA that appears when users click **Issuer Statement** in the **Certificate General** dialog box. The issuer statement is a policy statement that gives legal and other pertinent information about the CA and its issuing policies, limitations of liability, and so forth.

The issuer statement file must be installed on the server before you install Windows 2000 Certificate Services. This file, named Capolicy.inf, must be placed in the directory in which Windows 2000 Server is installed — the *systemroot* directory. (The default *systemroot* is C:\Winnt.) CAPolicy.inf can contain the text you want to be displayed as the policy statement, or it can contain a URL that points to the policy statement, for example, a Web page. For more information about how to create the Capolicy.inf file, see Certificate Services Help.

Installing Windows 2000 Certificate Services

Before you can install a CA, you must be logged on as either a member of the local Administrator security group for stand-alone computers or a member of the Domain Administrator security group for computers that are connected to the domain.

To install Windows 2000 Certificate Services

1. In Control Panel, click **Add/Remove Programs**.

   The **Add/Remove Programs** dialog box appears.

2. Click **Add/Remove Windows Components**.

   The Windows Component wizard appears.

3. In Windows Components, select the Certificate Services check box.
4. Click **Next**, and use the Windows Component wizard to install the CA.

Tables 16.5 through 16.9 describe the available CA configuration options for each page of the Windows Component wizard.

Note

After the CA is installed, the computer cannot be renamed, joined to a domain, or removed from a domain. Installing an enterprise CA requires Active Directory, so the CA computer must already be joined to the Windows 2000 domain.

Table 16.5 Certification Authority Type Selection Page

Option	Description
Enterprise root CA	Select to install an enterprise root CA.
Enterprise subordinate CA	Select to install an enterprise subordinate CA.
Stand-alone root CA	Select to install a stand-alone root CA.
Stand-alone subordinate CA	Select to install a stand-alone subordinate CA.
Advanced options	Select to configure advanced options in the **Public and Private Key Selection** page of the wizard.

Table 16.6 Public and Private Key Selection Page

Option	Description
Cryptographic service provider	Select the CSP to be used to generate the public key and private key set for the CA certificate. This CSP also manages and stores the private key. The default CSP is the Microsoft Base Cryptographic Provider or the Microsoft Enhanced Cryptographic Provider, depending on whether the server that is running Windows 2000 contains exportable or nonexportable cryptography. If you want to use another CSP, such as a hardware-based CSP to manage and store the CA's private key, you must select the appropriate CA from the list of CSPs.
Hash algorithms	Select the message digest that is to be used for the digital signature of the CA certificates. The default is SHA-1, which provides the strongest cryptographic security.

Key length	Select a key length from the list, or type a key length for the private key and public key. The default key length is 512 bits for the Base Cryptographic Provider and 1,024 bits for the Enhanced Cryptographic Provider. The minimum key length you can specify is 384 bits, and the maximum is 16,384 bits. Use a key of at least 1,024 bits for CAs. In general, the longer the key, the longer the safe lifetime of the private key. Use the longest key that is feasible and that meets both CA performance requirements and CSP key storage limitations.
Use existing keys	Enables the selection of an existing private key from the list. The existing private key is used for the CA. You might need to use this option to restore a failed CA.
Use the associated certificate	Enables the selection of the certificate that is associated with the existing private key which is used for the CA. This option is not available unless you first select **Use the associated certificate**. You might need to use this option to restore a failed CA.
Import	Imports a private key that is not in the **Use existing keys** list. For example, you might import a private key from an archive for a failed CA.
View Certificate	Select this option to view the certificate associated with the private key in the **Use existing keys** list.

Table 16.7 CA Identifying Information Page

Option	Description
CA name Organization Organizational unit Locality State or province Country/region E-mail	Enter information that is to be used to uniquely identify the CA. This information is included in the CA certificate in the Subject field. The **CA name** that you enter here is used by Windows 2000 to identify the CA, so the **CA name** must be unique for each CA you install in your organization. However, all of the other information that is entered here can be the same if appropriate. Others can view the Subject field in the CA certificate to identify the CA or to find out how to contact the CA.
CA description	Enter a description for this CA (optional).

Validity duration	Enter the duration for the certificate lifetime for the root CA certificate, and select **Years**, **Months**, or **Weeks** from the list. The default certificate lifetime for root CAs is 2 years. You must choose a lifetime that supports your planned certificate life cycles. This option is not available for subordinate CAs because the certificate lifetime is determined by the parent CA.
Expires on	Lists the expiration date for the root CA certificate, which corresponds to the certificate lifetime in **Validity duration**.

Table 16.8 Data Storage Location Page

Option	Description
Certificate database Certificate log	By default, the certificate database and the log are installed at <*Drive*:>\WINNT \System32\CertLog, where <*Drive:*> is the letter of the disk drive where the CA is installed. You have the option of storing the database and the log on different drives to manage storage space. If this is something you want to do, type the new path and folder name in the **Certificate database** box or in the **Certificate log** box, or click **Browse** to select the new location.
Store configuration information in a shared folder	Select to store configuration information in a shared folder, and then type the path and folder name in the **Shared folder** box; or click **Browse** to select an existing folder. Members of the local Administrators security group are granted full control for the folder. Members of the Everyone security group are granted read permissions for the folder. The shared folder acts as a location where users can find information about certification authorities. This option is useful only if you are installing a stand-alone CA and do not have Active Directory.
Preserve existing certificate database	Select to preserve an existing certificate database. This option is available only when you are reusing a private key and the associated certificate from an existing CA configuration. You can use this option to restore a failed CA.

Table 16.9 CA Certificate Request Page (Subordinate CAs Only)

Option	Description
Send the request directly to a CA already on the network	Type the name of the parent CA, or click **Browse** to select the parent CA from a list of CAs. The certificate request is submitted to this CA, and the certificate is then processed and issued to the subordinate CA. If you make a request from a stand-alone CA, the CA is not certified automatically. An administrator must approve the certificate request before the CA can issue the certificate. You must later use the Certification Authority console to install the CA's certificate.

Save the request to a file	Select to save the request to a file, and then type the path and file name in the **Request file** box; or click **Browse** to select the file location. This option saves the certificate request to a request file that you can submit to an offline CA for processing. The CA is not certified automatically. You must later use the Certification Authority console to install the CA's certificate.

Key Archival and Management in Windows Server 2003

Updated: December 6, 2004
By David B. Cross and Avi Ben-Menahem

Windows Server 2003, Enterprise Edition introduces several new features in the area of Public Key Infrastructure (PKI) technologies and Certificate Authorities (CAs). One area of new functionality is private key archival, recovery, and management. This white paper covers best practices and procedural steps in a key recovery strategy as well as migration procedures for moving from a Microsoft Exchange Key Management Server (KMS) environment to a Windows Server 2003 Certificate Authority.

On This Page

Introduction

Windows Server 2003, Enterprise Edition introduces significant advancements in the area of data protection and private key recovery. Windows 2000 introduced the capability for data recovery with the implementation of Encrypting File System (EFS). EFS in both Windows 2000 and Windows Server 2003 supports the use of Data Recovery Agents (DRAs) to decrypt files that have been encrypted by other users. With the expanded functionality of the Windows Server 2003 Certificate Services to offer key archival and recovery services, an enterprise may choose to implement different strategies based on the needs for data protection and data recovery.

Microsoft first offered key archival and recovery features in the Exchange Server 4.0 product line through the KMS component of the Exchange Server. The KMS can act as a Registration Authority (RA) to a Microsoft Certificate Server to provide user registration, key archival, key recovery, and certificate publishing capabilities to an Exchange Server e-mail and collaboration system. The KMS allows an administrator to recover the lost or corrupted encryption private keys of a Microsoft Outlook® user and generate a new signing key. The Outlook client as well as many other Secure/Multipurpose Internet Mail Extension (S/MIME)–enabled mail clients support the use of separate signing and encryption key pairs. This method helps to isolate an organization from non-repudiation issues. For more information about Key Management Server and Windows PKI interoperability, see Appendix B: Additional Information.

The Windows Server 2003 key archival solution also provides migration flexibility to organizations that desire to migrate their existing Exchange 2000 KMS solution to a more integrated archival solution in the Windows Server 2003 CA or for those customers who have public key certificates from third-party CAs. The Windows Server 2003 PKI solution allows import and migration of third-party public keys and certificates into the Windows Server 2003 CA as an escrow solution as well.

Windows Server 2003 key archival can be performed in two different ways: manual key archival and automatic key archival. Manual key archival refers to a process where users manually export private keys from their keys store and send them to a CA Administrator who in turn imports them to the protected CA database. Automatic key archival is done as part of the certificate enrollment process. It is possible to mark certificate templates to require key archival. In such a case, the private key will be sent to the CA as part of the certificate request and will be archived automatically by the CA. Manual and automatic key archival are discussed in depth later in this document.

Understanding Key Archival and Recovery

Key recovery implies that the private key portion of a public-private key pair may be archived and recovered. Private key recovery does not recover any data or messages. It merely enables a user to retrieve lost or damaged keys, or for an administrator to assume the role of a user for data access or data recovery purposes. In many applications, data recovery

cannot occur without first performing key recovery. Figure 1 demonstrates a typical key archival and recovery scenario.

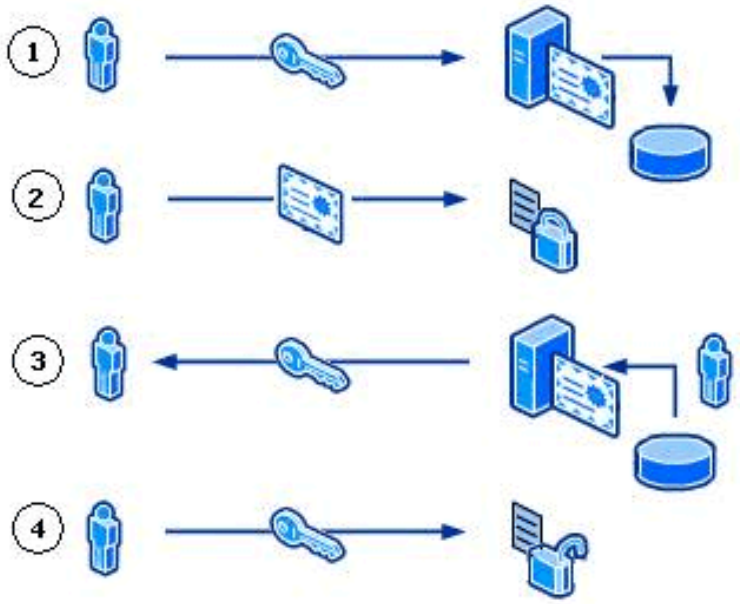

Figure 1: Key Archival and Recovery

1. The user requests a certificate from a CA and provides a copy of the private key as part of the request. The CA which is processing the request archives the encrypted private key in the CA database and issues a certificate to the requesting user.
2. The issued certificate can be used by an application such as EFS to encrypt sensitive files.
3. If at some point the private key is lost or damaged, the user can contact the company's Certificate Manager to recover the private key. The Certificate Manager, with the help of the Key Recovery Agent (KRA), recovers the private key, stores it in a protected file format, and sends it back to the user.
4. Once the user stores the recovered private key in the user's local keys store, it can be used by an application such as EFS to decrypt previously encrypted files or to encrypt new ones.

Understanding EFS Data Recovery

Encrypting a file always includes a risk that it cannot be read again. The owner of the private key, without which a file cannot be decrypted, might leave the organization without decrypting all of his or her files. Worse yet, he or she might intentionally or accidentally encrypt critical shared files so that no one else can use them. A user's profile might be damaged or deleted, meaning that the user no longer has the private key needed to decrypt encrypted files. Data recovery is one of the methods used for recovering files or data when encrypted files cannot be decrypted. By implementing data recovery and

DRAs, every encrypted file's encryption key (FEK) is encrypted by using the DRA's public key in addition to the user's public key. By using the associated private key, any designated DRA can decrypt any encrypted file within the scope of the EFS recovery policy. Figure 2 demonstrates encrypted file format when data recovery is implemented.

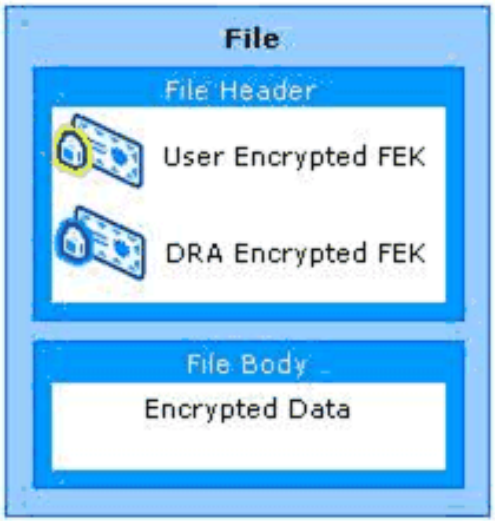

Figure 2: File Encryption Using Data Recovery

For more information about Data Recovery, see Appendix B: Additional Information.

Understanding When to Use Data Recovery vs. Key Recovery

It should be noted first that there is no definitive answer of whether key recovery or data recovery should be used over the other option. Both solutions have technical advantages and disadvantages and the decision to use one or both is subjective. Both are viable solutions separately and in summation. This section is written solely to identify some of the key advantages and disadvantages so that an organization may make an informed decision.

Key recovery should be used when the policy of a given company or enterprise permits the private keys and certificates of a given user(s) to be retrieved. Key archival and recovery imply that a person other than the original user may gain access to the private keys of another user. If a company or enterprise does not permit a person other than the original requestor to ever have access to private keys of other users, key archival and recovery should not be used.

Data recovery should be used when a company or enterprise requires the ability to recover data, but not access to the individual private keys of a user. For example, when private key recovery is provided, new operations can be performed

when using the key, not just data decryption.

Data recovery and key recovery should not be used when a company or enterprise wants to protect data from all parties except the original user. If data should not be accessed or recovered by any person other than the author/owner, neither data recovery nor key recovery processes should be implemented.

Table 1: Key Recovery vs. Data Recovery

	Key Recovery	Data Recovery
Certificates re-enrollment is **NOT** required.	✔	
Existing certificates revocation is **NOT** required.	✔	
Administrators do not have access to the user's private key.		✔
Non-repudiation assurance.		✔
Re-Encryption of all previously encrypted data is **NOT** required.	✔	

Key Recovery Best Practices

Some key best practices should be followed when implementing key archival and recovery in an organization. First, it should be noted that key recovery by an administrator implies impersonation and that administrators who perform key archival should be highly trusted. Implementation of operations of a key archival and recovery solution should be carefully planned and monitored for security purposes.

Other Best Practices

1. If a key is known to be compromised, it should be immediately revoked and never used again.
2. Keys or certificates that are used to secure high-value transactions, or are identified as high-value certificates should not be recovered except under extreme circumstances.
3. Private keys that are used for signing should never be archived or recovered as this introduces non-repudiation issues.

4. When a key has been compromised or lost, it should be revoked before allowing key recovery. Despite key compromise, it may be necessary to recover a key for the purpose of decrypting old data and encrypting with a new key.
5. Issue the least number of certificates and key pairs possible to ease key management and reduce user confusion.
6. Issue encryption certificates with longer lifetimes than signing certificates.
7. Issue only one valid key pair for a given application purpose at one time.
8. Develop a recovery process that combines role separation of Certificate Managers and KRAs.
9. Minimize the number of CAs that archive keys for a certificate purpose, that is, if possible, do not archive keys for users across a large number of CAs as recovery operations become confusing.
10. If performing key archival over the Certificate Server Web enrollment pages, it is important to use Secure Sockets Layer (SSL) on the enrollment Web site to protect the enrollment traffic from tampering or malicious interception.
11. Use roaming user profiles if possible to reduce user confusion, accidental key loss, and the need for manual import and export operations to enable key roaming.

Requirements

Key archival and recovery using a Windows Server 2003 CA has several technical dependencies.

- Enrollment requires the Certificate Management Protocol using CMS (CMC) protocol, which is only available in Windows XP client, Windows Server 2003 clients, and through the xenroll.dll ActiveX control in the CA Web enrollment interface. Through the Web enrollment interface, Windows 2000 and Windows ME may enroll for certificates with key archival.

  Note: Version 2 templates and key archival are not available to Windows 2000 Microsoft Management Console (MMC) enrollment.

  Note: Windows 2000 and Windows ME require that the key be marked for export during key archival enrollment.
- Active Directory® must have the Windows Server 2003 schema extensions.
- Mandating key archival requires a version 2 certificate template.
- Key archival is only available on a CA running Windows Server 2003, Enterprise Edition.
- Only a Microsoft Enterprise Certification Authority may be used.

  Note: Clients running down-level operating systems, such as Windows NT® 4.0 or Windows 98, will not be able to view certificate templates that allow key archival in the Web enrollment interface.

Top of page

Understanding Manual Key Archival

Manual key archival is supported on the Windows Server 2003 CA as a separate operation from enrollment while still offering centralized key archival. Users may export their private keys into *.pfx files [Public Key Certificate Standard (PKCS) #12 format] or through Outlook into the *.epf format. The following section describes the procedure to export private keys manually from a Windows client so that they may be manually archived on the CA. This is especially useful for users who have enrolled with third-party CAs that do not support key archival.

Exporting Keys and Certificates

Keys and certificates may be exported on Windows clients by one of two methods.

•PKCS #12 (*.pfx file) export from the Certificates MMC snap-in on Windows 2000 or Windows Server 2003
•*.epf file format from the Outlook 2000 or Outlook 2002 client

If a user has enrolled for Exchange Advanced Security with version 1 certificates (first offered with Exchange 5.0 Key Management Server), direct export from Outlook into the *.epf file format will be necessary. X.509 version 1 certificates and keys may not be exported into PKCS #12 format on the Windows client.

If only X.509 version 3 certificates have been used (first offered with the Exchange 5.5 Key Management Server SP1 and/or the NT 4.0 Certificate Server 1.0), the PKCS #12 format may be used.

Exporting Keys from the MMC

To export the certificate and private key while logged in as the user

1. Click the **Start** button, and then click **Run**.
2. Type **mmc.exe** and press **Enter**.
   An empty MMC shell appears.
3. Select the **Console** menu, and then select **Add/Remove Snap-in**.
   A dialog box appears with a list of all the snap-ins that have been added to this MMC shell.
4. Click **Add**.
   A list appears with all the registered snap-ins on the current machine.
5. Click the **Certificates** snap-in and click **Add,** choose **My User Account**, and then click **Finish**.
6. In the **Add Standalone Snap-in** dialog box, click **Close**. In the **Add/Remove Snap-in** dialog box, click **OK**.
   The MMC now contains the personal certificate store for the currently logged-on user.

7. Expand the tree view of the certificate store. Click through **Certificates - Current User**, **Personal**, and then **Certificates**. When you click the **Certificates** folder on the left, the right-hand pane will display a list of all the certificates for the currently logged on user.
8. Right-click the certificate intended for export.
9. Choose **All Tasks**, and then **Export** on the **Context** menu.
   A wizard will guide you through the export process.
10. Click **Yes, export the private key**, and then click **Next**.

    When exporting a private key, the *.pfx file format is used. The *.pfx file format is based on the PKCS #12, which is used to specify a portable format for storing or transporting a user's private keys, certificates, and miscellaneous secrets. For more information about the PKCS #12, see Appendix B: Additional Information.
11. Select the appropriate check boxes, and then click **Next.**

    As a best practice, strong private key protection should also be used as an extra level of security on the private key when exporting. The private key should be deleted only if you are performing archival and will no longer use the key on that machine.
12. The *.pfx file format (PKCS #12) allows a password to protect the private key stored in the file. Choose a strong password, and then click **Next**.
13. The last step is to save the actual *.pfx file. The certificate and private key can be exported to any writeable device, including a network drive or disk. After typing or browsing for a file name and path, click **Next**.

Once the *.pfx file and private key have been exported, the file should be secured on a stable media and transferred in a secure manner to the CA on which the key will be imported in accordance with the organization’s security guidelines and practices.

Exporting Keys from Outlook

To export a key from Outlook

1. In Outlook, click the **Tools** menu, and then select **Options**. Click the **Security** tab, and then click **Import/Export** (Figure 3).

Figure 3: Outlook Security Options

2.Click **Export your Digital ID to a file**, and then complete the **Filename**, **Password**, and **Confirm** (password confirmation) text boxes (Figure 4).

Figure 4: Outlook Import/Export Digital ID Options

3.Copy this file to a location accessible by the Certificate Server or manually transport it using a disk.

Importing a Key Manually on a CA

To import a key manually on a CA

•On the Certificate Server, open a command prompt window, and run the following command.

C:\CertUtil.exe –f –importKMS <name of file>

Note: The –f flag is required when the key and certificate pair have not been issued from the CA in question.

The file may be in one of three formats.

•KMS export file
•PKCS #12 format (*.pfx file)
•Outlook export format (*.epf)

Important: The previous command will work only after the CA was configured for key archival. For more information about the actions required for enabling key archival, see Implementing Key Archival Walkthrough..

Top of page

Understanding Automatic Key Archival

The following sections document in detail how the key archival and key recovery processes work in Windows Server 2003 as well as step-by-step guides for implementing the features in production.

Understanding the Archival Process

The general steps in the archival of a user's private key are described in the following steps and in Figure 5.

1. The client finds CAs in Active Directory (enrollment services container in the configuration partition).
2. The client makes an authenticated Distributed Component Object Model (DCOM) connection to the CA and requests the CA's Exchange certificate (encryption certificate).
3. The CA sends the exchange certificate to the client.
4. The client validates that the CA's exchange certificate has been signed by the same key as the CA signing certificate and performs a revocation status check. This ensures that only the intended CA may decrypt the certificate request containing the private key.

5. The client encrypts the private key corresponding to the request with the CA exchange certificate public key, builds a CMC request, and sends a CMC full PKI request to the CA.
6. The CA validates that the encrypted private key cryptographically pairs with the public key in the certificate request.
7. The CA validates the signature on the request with the public key in the request.
8. The CA encrypts the private key from the user request with a random Triple Data Encryption Standard (3DES) symmetric key and then encrypts the symmetric key with one or more KRA public keys.
9. The CA saves the encrypted key binary large object (BLOB) containing the encrypted private key and the symmetric key encrypted with one or more KRA public keys to the CA database.
10. The CA processes the certificate request normally.
11. The CA responds to the client with a CMC full PKI response.

Note: Only RSA Security encryption keys may be archived in the CA database. Signature only keys as well as non-RSA key pairs will not be archived. Denied and resubmitted requests will also not archive private keys.

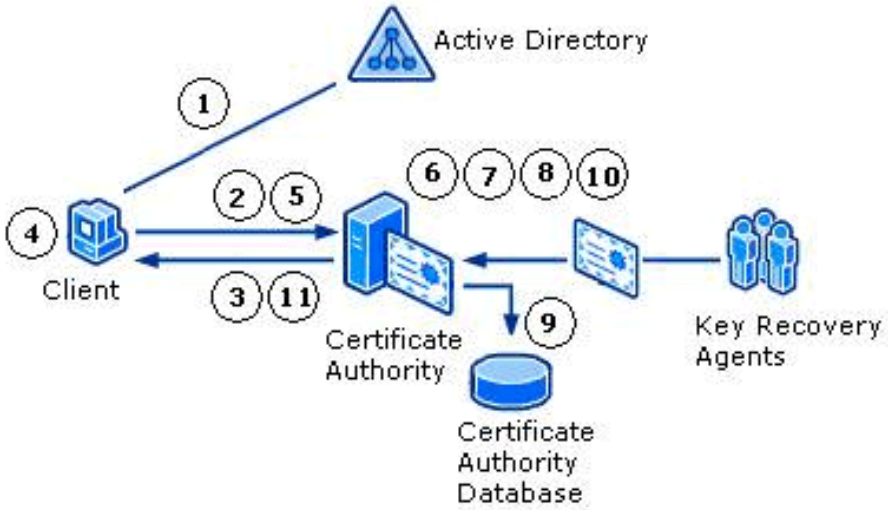

Figure 5: The Protocol for Key Archival

Certificate Request

A certificate request can be driven from at least three different sources in Windows Server 2003. The sources built into the operating system are the Web browser (Internet Explorer), the certificate enrollment wizard, and auto-enrollment. All three methods call xenroll.dll (Microsoft Enrollment Control), which provides various methods to support certificate request

generation and certificate installation. One of the formats the certificate request uses is the CMC format for certificate requests, which supports an optional encrypted data payload. Technically, any client that supports the CMC protocol may enroll with an Enterprise CA and request that the private key be archived by the CA. The CA enforces the archival option through a template flag. For more information about the CMC request format, see Appendix A: Certificate Request Structure.

CA Exchange Certificate Retrieval

Before the private key can be encrypted and delivered to the CA server, the client must first retrieve the CA's exchange certificate. The CA exchange certificate is an encryption certificate for the Certification Authority that can be used by clients to encrypt their private keys as part of their certificate request. The CA exchange certificate is issued by the CA itself where the subject and issuer are almost the same. However, the subject contains a suffix to distinguish the certificate from a root CA. Getting the CA exchange certificate can be accomplished using the existing ICertRequest::GetCACertificate method. The caller of the ActiveX xenroll.dll control will need to retrieve this certificate and verify it prior to calling xenroll.dll to create the request and to encrypt the private key. Xenroll will also perform this action. The client will verify that the exchange certificate is signed by the same key that issued the CA certificate; this step is performed to provide additional security and to prevent man-in-the-middle attacks. For that reason, a CA may not use a certificate issued by another CA for an exchange certificate.

CA Exchange Certificate Generation

A Windows Server 2003 CA will automatically generate a short-lived exchange certificate for use of the key archival mechanisms. By default, if the CA Exchange template is available, it will be used for extensions and the validity and overlap periods. If the validity and overlap periods in the registry differ from the template, the registry values are rewritten, so there will be consistent behavior if the template is unavailable in the future. If the template is not available, hard-coded extensions will be used along with the registry validity and overlap period settings.

If the following registry flag is configured using certutil.exe, the template must be available or the attempt to generate a CA Exchange certificate will fail. In addition, the machine object of the CA (LOCAL SYSTEM) must have Enroll access to the template. This flag is not currently set by setup; it must be manually applied.

certutil –setreg ca\CRLFlags +CRLF_USE_XCHG_CERT_TEMPLATE

The CA Exchange certificate template has a default expiration of one week and a template overlap period of one day. Note that the previous settings apply to both enterprise and stand-alone CAs.

Restricting Key Archival

Windows Server 2003 may be effectively prevented from archiving private keys through the use of qualified subordination and policy constraints. When submitting and processing a request to a parent or root CA for a subordinate CA certificate, the parent CA may exclude the key archival policy in the subordinate CA. For more information about Qualified Subordination, see Appendix B: Additional Information.

Private Key Payload

For Windows Server 2003 clients and servers, the encrypted private key is stored in an unauthenticated attribute as a Cryptographic Message Syntax (CMS) EnvelopedData object. This poses no concern from a security perspective as the private key is later validated to cryptographically match the public key in the certificate request, which is also signed by the same private key. For more information about the format and object structure, see Appendix A: Certificate Request Structure.

Key Verification

The CA must check that the encrypted private key is cryptographically associated with the public key in the certificate request. Because the client encrypted the private key using the public key from the CA's exchange certificate, the CA can decrypt the private key payload contained in the request. For encryption keys, the CA encrypts randomly generated data with the public key in the request, and then decrypts the data with the private key passed in the unauthenticated attributes of the CMC request. The decrypted data is verified against the original random data. If any of the validation steps fail, or if the data does not match, the request is rejected. For signing keys, the CA will reject the request and will not archive a key that is marked for signature only.

To verify the cryptographic association between the public and private keys, the CA loads the key pair into a cryptographic service provider (CSP) that supports the specified algorithms. The CA will use the default-enhanced RSA CSP on the CA by default or a hardware CSP, if available. Both the encrypted private key material payload and the private key loaded into the server-based CSP are securely discarded since the private key will be archived using a different key controlled by a recovery agent and is not available for the CA after verification of the payload. The decrypted private key memory is zeroed out prior to freeing the memory.

Key Encryption

Once the private key has been successfully verified, the CA must then encrypt the private key and archive it for future recovery. The CA will not encrypt the private key using its own key(s) to provide separation between the recovery agent role and the CA administrator role. The certificate request process and certificate issuance are completed after key archival has occurred. For more information about role separation, see Configuring the CA to Allow Key Archival.

The private key will be encrypted by the CA using a dynamically generated symmetric key that is itself encrypted using a public key from a KRA certificate (or certificates) configured on the CA. Each certificate request and private key archival operation will generate a new random 3DES symmetric key using the same CSP that was used to generate the CA signing key. This ensures that a hardware security module (HSM) can be used to generate the long-term keys that protect the individual private keys. The symmetric key algorithm is 3DES by default and cannot be changed.

If multiple recovery agents are associated with the CA, the symmetric key will be encrypted individually with each recovery agent's public key to allow any one recovery agent to perform a recovery operation. The actual KRA(s) used may randomly vary if a round-robin selection of KRAs is chosen. For example, if a CA Administrator configures four KRA certificates on a CA for use, but requires a minimum of two KRAs to be used, when the CA service starts, a random start point in the KRA list is chosen. Two of the four KRAs will be chosen from that start list start point and then for each subsequent archival operation, the list order will be incremented by two. Therefore, a round robin affect is achieved for all key archival operations. These encrypted keys along with the issued certificate will be collectively referred to as the recovery BLOB and are stored as a PKCS #7–encrypted BLOB in the database.

Key Archival

The private key database is the same as the database used to store the certificate requests. The Windows Server 2003 Certification Authority database has been extended to support storing the encrypted private key along with the associated encrypted symmetric key and issued certificate. The recovery BLOB will be stored in the same row as the signed certificate request and any other information the CA persists in its database for each request transaction. The actual encrypted BLOB is stored as an encrypted PKCS #7 BLOB. For more information about the format and object structure, see Appendix A: Certificate Request Structure.

The Microsoft Certification Authority uses the Microsoft Jet database engine upon which various Jet utilities may be used for maintenance purposes.

Top of page

Understanding User Key Recovery

A recovery operation is initiated by an end-entity that has lost access to a private key. The key recovery process is explained in Figure 6.

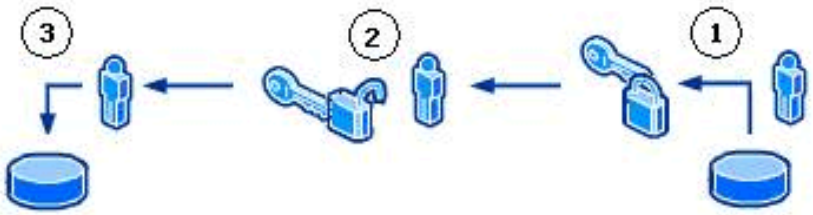

Figure 6: Key Recovery Process

The following are the key recovery steps.

1. The Certificate Manager (CA Officer) locates and retrieves the user's encrypted private key from the CA database. The encrypted BLOB(s) is protected by access control lists (ACLs) to ensure that only a valid Certificate Manager is able to copy the BLOB from the database. Since it is encrypted with the KRA's public key, the Certificate Manager cannot extract the user's private key but it can retrieve it from the CA. The encrypted PKCS #7 BLOBs in the database contain the Issuer name and serial number of each KRA certificate for KRA identification purposes. Once extracted, the Certificate Manager sends it to the appropriate KRA for decryption. For more information about the format and structure of the recovery BLOB, see Appendix A: Certificate Request Structure.
2. The KRA decrypts the user's private key and stores it in a password-protected file format (PKCS #12) and sends it to the user.
3. The user imports the key to the user's local, protected, keys store.

It is at the discretion of organizational policy whether a KRA and a Certificate Manager may be combined into one role or separate roles. For better security, it is recommended that the Certificate Manager and the KRA roles be separated. It should be noted that since the Windows Server 2003 Certification Authority supports role separation, an organization could easily implement a two-step process to recover the private key(s) of a user.

Understanding Role Separation

Role-based administration is used to organize CA administrators into separate, predefined task-based roles. It is recommended that the management roles are distributed among several individuals in your organization to ensure that a single individual cannot compromise PKI services. Role separation enables one person to audit the actions of another person.

The Common Criteria PKI management roles in Windows Server 2003 include the following:

1. **CA Administrator** Configures and maintains the CA, designates other CA administrators and certificate managers, and renews CA certificates.
2. **Certificate Manager** Approves or denies certificate enrollment requests and revokes issued certificates.
3. **Backup Operator** Performs backups of the CA database, the CA configuration, and the CA's private and public key pair (also known as a key pair).
4. **Auditor** Defines what events are audited for Certificate Services and reviews the security log in Windows Server 2003 for success and failure audit events that are related to Certificate Services.

Note: Role-based administration is supported by both Windows Server 2003 Enterprise CAs and stand-alone CAs running Windows Server 2003, Enterprise Edition or Windows Server 2003, Datacenter Edition.

To help determine role separation, you can use the Common Criteria specification, which defines security standards for all forms of network security and includes specifications for managing PKIs.

For more information about Role-Separation and Common Criteria, see Appendix B: Additional Information.

Understanding the Key Recovery Agent Role

KRAs are Information Technology (IT) administrators who can decrypt users' archived private keys. An organization can assign KRAs by issuing KRA certificates to designated administrators and configure them on the CA. The KRA role is not one of the default roles defined by the Common Criteria specifications but a virtual role that can provide separation between Certificate Managers and the KRAs. This allows the separation between the Certificate Manager, who can retrieve the encrypted key from the CA database but not decrypt it, and the KRA, who can decrypt private keys but not retrieve them from the CA database. For more information about how to implement Key Recovery Agents, see Implementing Key Archival Walkthrough.

Key Recovery Agent Certificate

A new certificate template exists in the Windows Server 2003 schema to support the KRA role. This certificate will use a unique Extended Key Usage extension to identify a KRA certificate. A Windows Server 2003 Certification Authority will only use certificates that have been properly formatted as per the following information, although it is not necessary for the certificate to contain the Microsoft-specific extensions.

KRA certificates, when issued by an Enterprise CA, are automatically published in the configuration container of Active Directory. The actual certificates are published to the userCertificate attribute of the KRA object when issued to an IT administrator. The location in the configuration container is as follows:

(**CN=KRA,CN=Public Key Services,CN=Services,CN=Configuration,DC=<domainname>**).

The KRA certificate contains the following X.509v1 fields.

•Version
•Serial Number
•Signature Algorithm
•Valid From
•Valid To
•Subject
•Issuer
•Public Key

The KRA certificate contains the following X.509v3 extensions identified in RFC 3280.

•Authority Key Identifier
•Subject Key Identifier
•Authority Information Access
•Key Usage
•Subject Alternative Name
•CDP (CRL Distribution Point)
•Extended Key Usage (Key Recovery OID = 1.3.6.1.4.1.311.21.6)
•Application Policies (Policy Identifier = Key Recovery Agent)

The KRA certificate will contain the following X.509v3 extensions specific to Microsoft.

•Certificate Template Name
•Certificate Template Information

Note: None of the extensions is marked as critical; however, the Key Recovery Object Identifier (OID) must exist in the Extended Key Usage extension for the certificate to be used.

Understanding the Key Recovery Process

Users may require their private keys to be recovered for a multitude of reasons. Key recovery from a Windows Server 2003

CA may be accomplished using the command-line tool, certutil.exe, or through the Windows Server 2003 Resource Kit tool, krecover.exe.

The recovery of a private key is a manual process that requires the user(s) to contact an administrative authority to perform the necessary processes. It should be a best practice of any organization to

1. Validate the identity of a user requesting key recovery.
2. Separate the roles of CA Officer and KRA as a minimum of two physical persons.
3. Develop a mechanism to securely deliver the private keys and passwords to end users. Examples could include e-mail of the *.pfx file and leaving a voice mail with the password for the *.pfx file for the user. Discussion of these best practices is beyond the scope of this white paper.

Finding Recovery Candidates

A CA Officer (Certificate Manager) can perform recovery of private key(s) using the CN (CommonName), UPN (UserPrincipalName), down-level name (domainname\username), certificate serial number of the certificate, or an SHA1 (Secure Hash algorithm) hash (thumbprint) of the certificate. The process is known as finding candidate certificates for recovery.

Examples:

- User1@nwtraders.com (denotes UPN)
- nwtraders\user1 (denotes the down-level name)
- Users\nuser1 (denotes a user in the default users container of Active Directory)
- User1 (denotes the CN)
- <serial number of the certificate>
- <SHA1 hash (thumbprint) of certificate>

Finding Recovery Candidates by Using Command-Line Tools

To recover a user using their CN, type the following command.

Certutil -getkey <cn of user> outputblob

If only one certificate has been issued to that user, that certificate and private key will be generated into the specified <outputblob> file. If more than one certificate has been issued to that CN, certutil.exe will list the serial number(s) of all the

certificates issued to that CN on a specific Certification Authority.

Example:

```
"user1.nwtraders.com\CA1"

Serial Number: 1464be10000000000007
Subject: CN=user1, CN=Users, DC=nwtraders, DC=com
NotBefore: 1/13/2001 11:51 AM
NotAfter: 1/13/2002 11:51 AM
Template: KeyA, KeyArchival
Cert Hash(sha1): a2 7f 77 bc 2f 7b eb 26 bd 3e ed 43 b8 2a 24 04
2e 55 b8 64


Serial Number: 1464fcbc000000000008

Subject: CN=user1, CN=Users, DC=nwtraders, DC=com
NotBefore: 1/13/2001 11:51 AM
NotAfter: 1/13/2002 11:51 AM
Template: KeyA, KeyArchival
Cert Hash(sha1): 21 bd 88 2c 2a 84 0c e5 57 7b 2a bf f0 43 4b b3
ed bf 02 5a
```

Once the actual serial number of the certificate is known, the command may be run again with the serial number instead of the CN to retrieve the actual recovery BLOB.

Important: If a CA Administrator has configured the server to use, for example, three of a possible five available KRAs, certutil –recoverkey will list the three KRAs that were used to encrypt the selected subject's private key. At the time of archival, the three KRAs were randomly selected (out of five possible) by the CA to encrypt the subject’s private key. To perform the recovery operation, it will be necessary to have one of any of the KRAs listed by certutil -recoverkey.

Finding Recovery Candidates by Using the Certificate Authority MMC Snap-In

A CA Officer may determine the serial number of a certificate that has been archived.

To determine the serial number of an archived certificate

1. Log on to a CA or a Windows XP Professional machine that has the Administration Tools pack installed as a CA Officer who has Certificate Management authority of the user(s) in question.
2. On the **Administrative Tools** menu, open **Certification Authority**.
3. In the console tree, expand **Certification Authority**, and then click **Issued Certificates**.
4. On the **View** menu, click **Add/Remove Columns**.
5. In the **Add/Remove Columns** dialog box, in the **Available Column list**, select **Archived Key,** and then click **Add**. Archived Key should now appear in the Displayed Columns listing.
6. Click **OK**.
7. In the details pane, scroll to the right and ensure that the last issued key has a value in the Archived Key column.
8. Double-click the user certificate.
9. Click the **Details** tab
10. Write down the serial number of the certificate (do not include spacing between digit pairs). This is required for recovery.
11. Click **OK**.
12. Close **Certification Authority**.

Command-Line Key Recovery

Command line key recovery is done by using the certutil.exe command-line tool. This tool is installed by default on the Certificate Server and it can be installed separately on a Windows XP Professional or Windows Server 2003 machine as part of the Windows Server 2003 Administration Tool Pack. The certutil.exe command structure was designed to perform batch and scripted processes.

Recovery Using the Certificate Serial Number

Once the serial number(s) is known, key recovery can occur through the certutil.exe command- line tool.

To recover private key(s)

1. Open a command-prompt window.
2. At the command prompt, type
   Certutil -getkey ########################## outputblob
   where ########################## is the serial number of the certificate that should be recovered and outputblob is the file name of the encrypted BLOB that is extracted from the CA database.
3. If a specific CA is to be targeted instead of all Enterprise CAs in the forest, use the following syntax.
   Certutil -config <machine name\CA name> -getkey ########################## outputblob

4. The outputblob file is a PKCS #7 file containing the KRA certificate(s) and the user certificate with its entire certificate chain. The inner content is an encrypted PKCS #7 structure containing the private key [encrypted to the KRA certificate(s)].
5. To decrypt the PKCS #7 encrypted BLOB, the logged on user must have the private key of one or more of the KRAs for the target encrypted BLOB. If the CA Officer is not a Key Recovery Agent, the CA Officer must transfer the encrypted BLOB file to a user that holds a KRA private key for further processing.
6. At the command prompt, type the following command:

   Certutil -recoverkey outputblob user.pfx –p password
   where:

   •outputblob is the encrypted PKCS #7 file to be decrypted.
   •user.pfx is the output file name for the user private keys in PKCS #12 format.
   •password is the password for the *.pfx file.

   The PKCS #12 format allows for the private key file to be protected with a password. Certutil.exe will prompt the KRA for a password when generating the PKCS #12 file.

   The system will search for a valid private key in the store of the logged in user that corresponds to a valid KRA certificate that was used to encrypt the recovery BLOB. If the user does not have the proper private key in their local store, they will receive an error.
7. Enter and confirm a password for the file that cannot be randomly guessed. The user should be given the password through a secure out-of-band mechanism that is separate from the file itself to ensure strong security in the recovery process.

   Note: An event log message will be generated when a private key is recovered from the database. The event log message will be similar to the following:

```
Event Type: Success Audit
Event Source: Security
Event Category: Object Access
Event ID: 787
Date:  2/19/2001
Time:  5:23:45 PM

User:  NWTRADERS\User1

Computer: SERVER1
```

```
Description: Certificate Services retrieved an archived key.
Request ID 12
```

Recovery Batch File

The certutil.exe –getkey command is an advanced tool that can be used with explicit commands and also build batch file syntax to perform a complete recovery operation. The following is the command-line syntax for certutil.exe –getkey.

Usage:

CertUtil -GetKey [Options] SearchToken [RecoveryBlobOutFile]

Retrieve archived private key recovery blob

SearchToken—Used to select the keys and certificates to be recovered.

RecoveryBlobOutFile—Output file containing a certificate chain and an associated private key, still encrypted to one or more KRA certificates.

If the following command is performed using any of the previous SearchToken criteria, the certutil.exe tool will output the complete syntax that can be used.

Example:

certutil -v -getkey user1@northwindtraders.com >myBatchfile.bat

Outputfile:

```
@goto start
Querying northwind5.nwtraders.com\TestCA9.....................


"northwind5.nwtraders.com\TestCA9"
  Serial Number: 611e23c200030000000e
  Subject: E=user1@nwtraders.com, CN=user1, CN=Users, DC=nwtraders, DC=com
```

```
  UPN:user1@nwtraders.com
  NotBefore: 1/12/2002 1:24 PM
  NotAfter: 1/12/2003 1:24 PM
  Template: EFS2
  Version: 3
  Cert Hash(sha1): d6 41 99 e7 e7 24 73 34 02 41 53 2d 29 11 a8 3b e6 aa 12 2f

  Serial Number: 6131f9c300030000000f
  Subject: E=user1@nwtraders.com, CN=user1, CN=Users, DC=nwtraders, DC=com
  UPN:user1@nwtraders.nttest.microsoft.com
  NotBefore: 1/12/2002 1:45 PM
  NotAfter: 1/12/2003 1:45 PM
  Template: EFS2
  Version: 3
  Cert Hash(sha1): 1a cc 8d 26 7f 9f 70 6c 65 05 a0 84 8c 4c e9 b7 b4 6c 66 a3

:start

@echo Version 3 certificates and keys:
CertUtil -config "northwind5.nwtraders.com\TestCA9" -getkey
611e23c200030000000e "user1-611e23c200030000000e.rec"

CertUtil -config "northwind5.nwtraders.com\TestCA9" -getkey
6131f9c300030000000f "user1-6131f9c300030000000f.rec"

CertUtil -p "UQcYLsJ(57s]FuBl" -recoverkey -user "user1-611e23c200030000000e.rec" "
user1-611e23c200030000000e.p12"

CertUtil -p "UQcYLsJ(57s]FuBl" -recoverkey -user "user1-6131f9c300030000000f.rec" "
user1-6131f9c300030000000f.p12"

CertUtil -p "UQcYLsJ(57s]FuBl,iG1-bOt&tqdvJiu1" -MergePFX -user "user1-611e23c20003
0000000e.p12,user1-6131f9c300030000000f.p12" "user1.p12"

@del user1-611e23c200030000000e.rec
@del user1-611e23c200030000000e.p12
@del user1-6131f9c300030000000f.rec
```

```
@del user1-6131f9c300030000000f.p12
@echo PASSWORD: "iG1-bOt&tqdvJiu1"

@goto exit
CertUtil: -GetKey command FAILED: 0x8002802c (-2147319764)
CertUtil: Ambiguous name.
```

The following is an explanation of the previous syntax, which can be run by Certificate Managers that hold a valid KRA private key.

1. The tool finds all possible CAs that are registered in Active Directory. Then, it will query each CA for keys that have been archived for the specified user.

```
@goto start
Querying northwind5.nwtraders.com\TestCA9.....................
```

2. The tool will show the archived keys found for the user on each CA that is queried. It will list the most important details on each archived key found that will be useful for Certificate Managers and administrators.

```
"northwind5.nwtraders.com\TestCA9"
  Serial Number: 611e23c200030000000e
  Subject: E=user1@nwtraders.com, CN=user1, CN=Users, DC=nwtraders, DC=com
  UPN:user1@nwtraders.com
  NotBefore: 1/12/2002 1:24 PM
  NotAfter: 1/12/2003 1:24 PM
  Template: EFS2
  Version: 3
  Cert Hash(sha1): d6 41 99 e7 e7 24 73 34 02 41 53 2d 29 11 a8 3b e6 aa 12 2f

  Serial Number: 6131f9c300030000000f
  Subject: E=user1@nwtraders.com, CN=user1, CN=Users, DC=nwtraders, DC=com
  UPN:user1@nwtraders.nttest.microsoft.com
  NotBefore: 1/12/2002 1:45 PM
  NotAfter: 1/12/2003 1:45 PM
  Template: EFS2
  Version: 3
  Cert Hash(sha1): 1a cc 8d 26 7f 9f 70 6c 65 05 a0 84 8c 4c e9 b7 b4 6c 66 a3
```

3. The tool will retrieve the encrypted recovery BLOB(s) from the CA. Note that the administrator must be a Certificate Manager for the recovered user(s) on the CA.

```
@echo Version 3 certificates and keys:
CertUtil -config "northwind5.nwtraders.com\TestCA9" -getkey
611e23c200030000000e "user1-611e23c200030000000e.rec"

CertUtil -config "northwind5.nwtraders.com\TestCA9" -getkey
6131f9c300030000000f "user1-6131f9c300030000000f.rec"
```

4. The tool will decrypt the recovery BLOB(s) and generate *.pfx files (PKCS #12 format) for each recovered key and set a random password as denoted after the –p parameter. Note that the administrator must have a valid KRA private key to perform this step.

```
CertUtil -p "UQcYLsJ(57s]FuBl" -recoverkey -user "user1-611e23c200030000000e.rec"
"
user1-611e23c200030000000e.p12"

CertUtil -p "UQcYLsJ(57s]FuBl" -recoverkey -user "user1-6131f9c300030000000f.rec"
"
user1-6131f9c300030000000f.p12"
```

5. The tool will merge the multiple *.pfx files (if applicable) into a single *.pfx file to simplify the process for the user installing recovered keys. The certutil.exe –MergePFX command is automatically used to perform this process.

```
CertUtil -p "UQcYLsJ(57s]FuBl,iG1-bOt&tqdvJiu1" -MergePFX -user "user1-
611e23c200030000000e.p12,user1-6131f9c300030000000f.p12" "user1.p12"
```

6.The tool will clean up any temporary files used during the process.

```
@del user1-611e23c200030000000e.rec
@del user1-611e23c200030000000e.p12
@del user1-6131f9c300030000000f.rec
@del user1-6131f9c300030000000f.p12
@echo PASSWORD: "iG1-bOt&tqdvJiu1"

@goto exit
```

Note that the following output is expected when the user has multiple archived keys.

```
CertUtil: -GetKey command FAILED: 0x8002802c (-2147319764)
CertUtil: Ambiguous name.
```

This output implies that a full key recovery could not be performed because the command line was not specific enough to retrieve a specific recovery BLOB.

CA Officer Does Not Have a KRA Certificate

As mentioned previously, a CA Officer need not also hold a KRA certificate. In the case where the CA Officer does not hold a valid KRA certificate and only has permissions to retrieve a recovery BLOB, running certutil.exe –getkey <serial number> with no output file name will list the certificate hashes (thumbprints) of all of the KRA(s) that may be used to decrypt (recover) the recovery BLOB in question. The CA Officer can then determine which KRA(s) may be used and send the recovery BLOB to one of the valid KRA(s).

The previous procedure is extremely useful in an organization that has a round-robin selection of KRA(s) when archiving private keys. In a round-robin archival, it may be difficult to always predict which KRA(s) were used to encrypt any specific private key of a user. It is probably a good best practice to keep the list of all KRA certificates hashes accessible so that they may be readily identified in the previous scenario.

The following certutil command may be used to list all the information from Active Directory, including KRA certificate hashes.

Certutil.exe –DS – v > c:\output.txt

Note: This command will dump all PKI information and certificates from Active Directory and may be more usable when placing the output in a text file as shown previously.

Setting the Timeout Option

When keys are recovered from a CA using the command line or the key recovery tool, the recovery tool(s) will build a complete chain for the end-entity certificate, if possible. Sometimes a chain may not be able to be built after a long time due to infrastructure changes, CA certificate availability, and so on; the tool may also timeout when trying to build these chains. The certutil –recoverkey command and other commands that verify chains or construct *.pfx files accept a –t MilliscondCount option. This allows key recovery to avoid a 15-second timeout for each user key when building a chain. Fetching the certificate specified in the AIA extension Uniform Resource Locators (URLs) can time out when the recovered keys are associated with expired encryption certificates and the CAs have been decommissioned. For more information about certificate chains and status, see Appendix B: Additional Information.

Using the Key Recovery Tool

The key recovery tool is a Windows Server 2003 Resource Kit tool for recovering keys from a Windows Server 2003 CA instead of using the certutil.exe command-line tool.

The tool may be launched from the command line by running krecover.exe. The tool is dependent on the existence of the certutil.exe and other binaries, which are available by default in Windows Server 2003 and in the Administration tool pack of Windows XP Professional. The tool is also dependent on one dynamically linked library (crutredir.dll), which is installed by the resource kit.

When launched, the tool will look similar to the following:

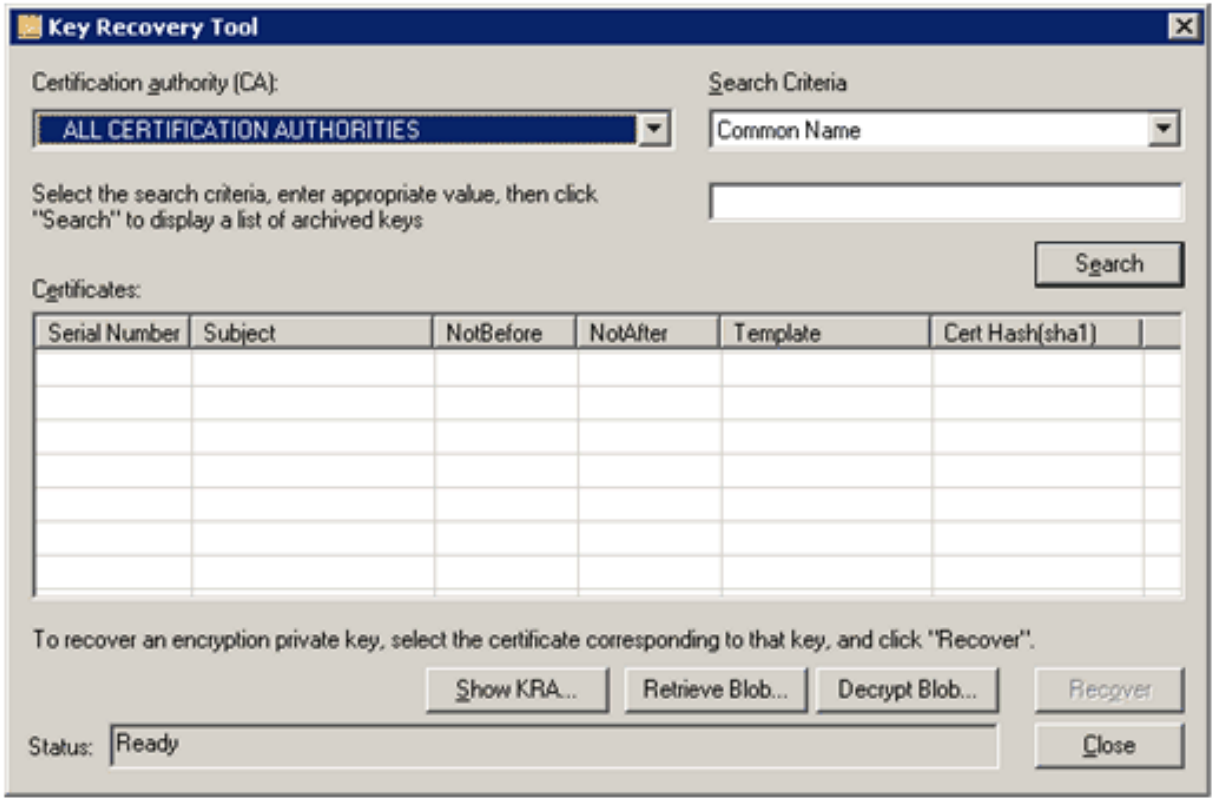

Figure 7: Key Recovery Tool Interface
See full-sized image

The tool will automatically enumerate the forest for the currently logged on user and identify Enterprise CAs that may have archived private keys stored. The tool will filter out the following from the list of available CAs.

•CAs that are not Enterprise CAs
•CAs that are not running Windows Server 2003, Enterprise Edition
•CAs that are running Windows 2000
•CAs that the current user does not have CA Administrator or CA Manager permissions for

The following are several options enabled by the tool.

•**Search**—Allows a Certificate Manager to search for archived keys for a user based on several search criteria options.
•**Recover**—Allows a Certificate Manager who holds a KRA private key to recover a user from a CA database.

- **Show KRA**—Displays the valid KRA certificates that may decrypt the user private key.
- **Retrieve BLOB**—Allows a Certificate Manager to retrieve the encrypted user's private key from the database for the purpose of transferring to a KRA.
- **Decrypt BLOB**—Allows a KRA to decrypt and create a *.pfx file for an encrypted BLOB retrieved from the database by a Certificate Manager.

If the user does not have adequate permissions on the CA to recover a user or is not a Certificate Manager, an error similar to the following will be displayed.

Figure 8: Failed to enumerate the KRA certificates Error Message

Recovering a Specific User's Keys

To recover the private key(s) of a specific user, enter a user name and click **List** to list all the certificates for that user with an archived key. In the **User Name** field, enter the user name (as described previously), or domain\user_name, or user_name@northwindtraders.com—any input that is valid for the certutil -getkey command.

Important: For the key recovery tool to recover a key for a user, the same permissions apply as in the certutil.exe command-line as described in the previous section.

If no key(s) is found for the selected user on the selected CA(s), the following dialog box will appear.

Figure 9: No archived keys were found Message

– or –

Figure 10: No archived keys were found Message

Here, you can select one or more certificates to be recovered, and then click **Recover**. If you select more than one certificate, you will be prompted if you want all the keys to be recovered in a single *.pfx file, or if you want separate *.pfx files for every certificate.

Figure 11: Single or Separate *.pfx Prompt Message

In either case, you will be prompted for a destination for the *.pfx file. The dialog box looks like the following:

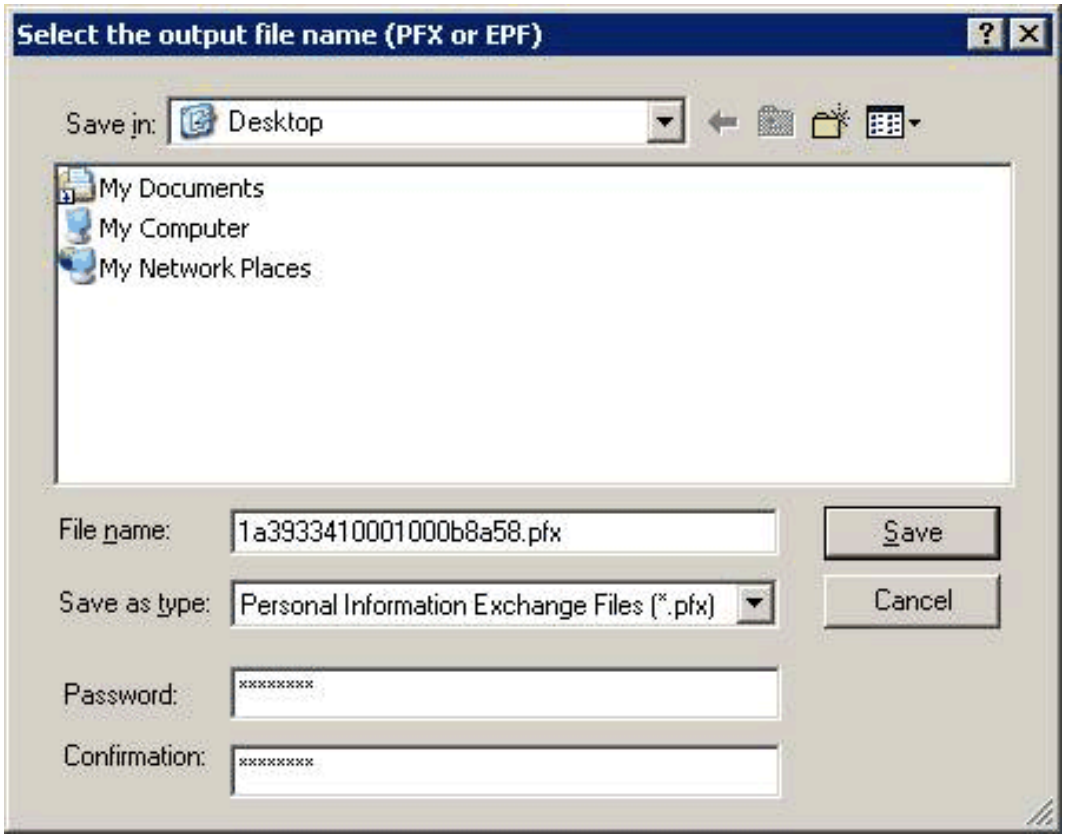

Figure 12: Output File Dialog Box

When saving a *.pfx file, the tool will prompt for a location to save the *.pfx file and require a password to be set on the PKCS #12 file. When recovering a single private key, the default name for the *.pfx file will be the certificate serial number that corresponds to the private key. The output file name may be changed.

If the tool cannot recover a key for a user, the following dialog box will be displayed.

Figure 13: Recovery Failure Message

If you do not have a valid certificate and private key for a KRA that was used to encrypt and archive the private key of the user on the CA, the following error will be displayed.

Figure 14: Invalid Certificate and Private Key Message
See full-sized image

Recovering Version 1 and Version 3 Certificates

A Windows Server 2003 CA may be used to archive both X.509, and version 1 and version 3 certificates as detailed previously. The key recovery tool allows recovery of both types, although x.509 version 1 certificates must be exported separately as they may only be exported in the *.epf format for Outlook. The Windows Server 2003 CA supports the following version 1 certificates and keys to be imported and exported from the database.

•Profile Version=2, CAST=3, Protection=40, EPFALG_CAST_MD5
•Profile Version=2, CAST=3, Protection=64, EPFALG_CAST_MD5
•Profile Version=3, HashCount=0, Protection=128, EPFALG_RC2_SHA
•Profile Version=3, Protection=128, EPFALG_3DES

If no version 1 certificates are in the recovery selection, the user will not be prompted any differently. If one x.509 version 1 certificate is in the selection, the user will be prompted for an *.epf file name for export. If more than one x.509 version1 certificate is selected, the user will be prompted whether the user would like one *.epf file or separate *.epf files for each key recovered. X.509 version 1 certificates will be recovered first and then the x.509 version 3 certificates will be recovered. If both version 1 and version 3 certificates have been selected, the user will be prompted twice whether the certificates should be recovered in a single file or separate files. X.509 version 1 and version 3 certificates may not be recovered in a single file together.

Importing Recovered Keys

The following steps would be performed by an end user who has received the recovered key file (*.pfx file) and associated password from an administrator. It is assumed that the password and associated file have been transmitted to the user in a method consistent with the organization's security policy.

To import a recovered key for an individual user

1. Log on as the user who needs to recover their private key(s).
2. Click the **Start** button, and then click **Run**.
3. Type **mmc.exe**, and then press **Enter.**
   An empty MMC shell starts up.
4. Select the **Console** menu, and then click **Add/Remove Snap-in**.
   A dialog box appears with a list of all the snap-ins that have been added to this MMC shell.
5. Click **Add**.
   A list appears with all the registered snap-ins on the current machine.
6. Double-click the **Certificates** snap-in, choose **My User Account**, and then click **Finish**.
7. In the **Add Standalone Snap-in** dialog box, click **Close**, and then click **OK** in the **Add/Remove Snap-in** dialog box.
   The MMC now contains the personal certificate store for the user.
8. Expand the tree view of the certificate store. Click **Certificates**, **Current User**, **Personal**, and then **Certificates**.
9. In the console tree, right-click **Personal**, click **All Tasks**, and then click **Import**.
10. In the **Certificate Import Wizard** dialog box, click **Next**.
11. On the **Files to Import** page, in the **File name** box, type the name and path of the recovered key file (*.pfx), and then click **Next**.
12. On the **Password** page, in the **Password** box, type the password for the key file, and then click **Next**.
13. On the **Certificate Store** page, click **Automatically select the certificate store based on the type of certificate**, and then click **Next**.

14.On the **Completing the Certificate Import Wizard** page, click **Finish**.
The wizard will report that the import was successful.

Note: A *.pfx file can also be selected (double-click) to invoke the certificate import wizard.

Top of page

Implementing Key Archival Walkthrough

The first step in enabling key archival on a CA is enrolling for one or more KRA certificates.

Enrolling a Key Recovery Agent

The first step in configuring Certificate Server for key archival is to enroll KRAs for KRA Certificates. The following section explains the steps for enrolling a KRA.

Configuring the Certificate Templates

A certificate template suitable for creating KRA certificates is installed in Active Directory. By default, only an Enterprise Administrator or a Domain Administrator may request a KRA certificate as defined by the default ACLs on the KRA certificate template. The certificate template ACLs can be viewed in the Certificate Templates MMC snap-in; in addition, using the Certificate Templates MMC snap-In, certificate templates can be cloned or edited.

Note: Only a domain with the Windows Server 2003 schema will support version 2 templates and only a Windows Server 2003, Enterprise Edition may issue a version 2 template certificate.

To configure a certificate template

1.Log on as an Enterprise or Domain Administrator to the CA machine.
2.Click the **Start** button, click **Run**, and then type **certtmpl.msc**.

3.Click **OK**.

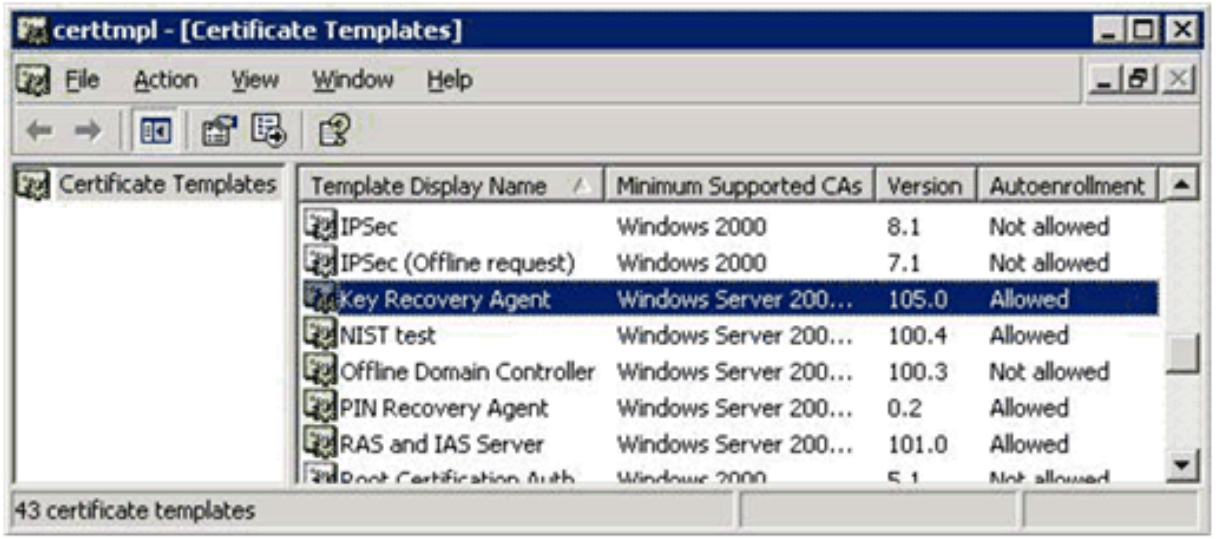

Figure 15: Certificate Templates MMC Snap-In
See full-sized image

4.In the details pane, double-click **Key Recovery Agent**.
5.In the **Key Recovery Agent Properties** dialog box, click the **Security** tab.

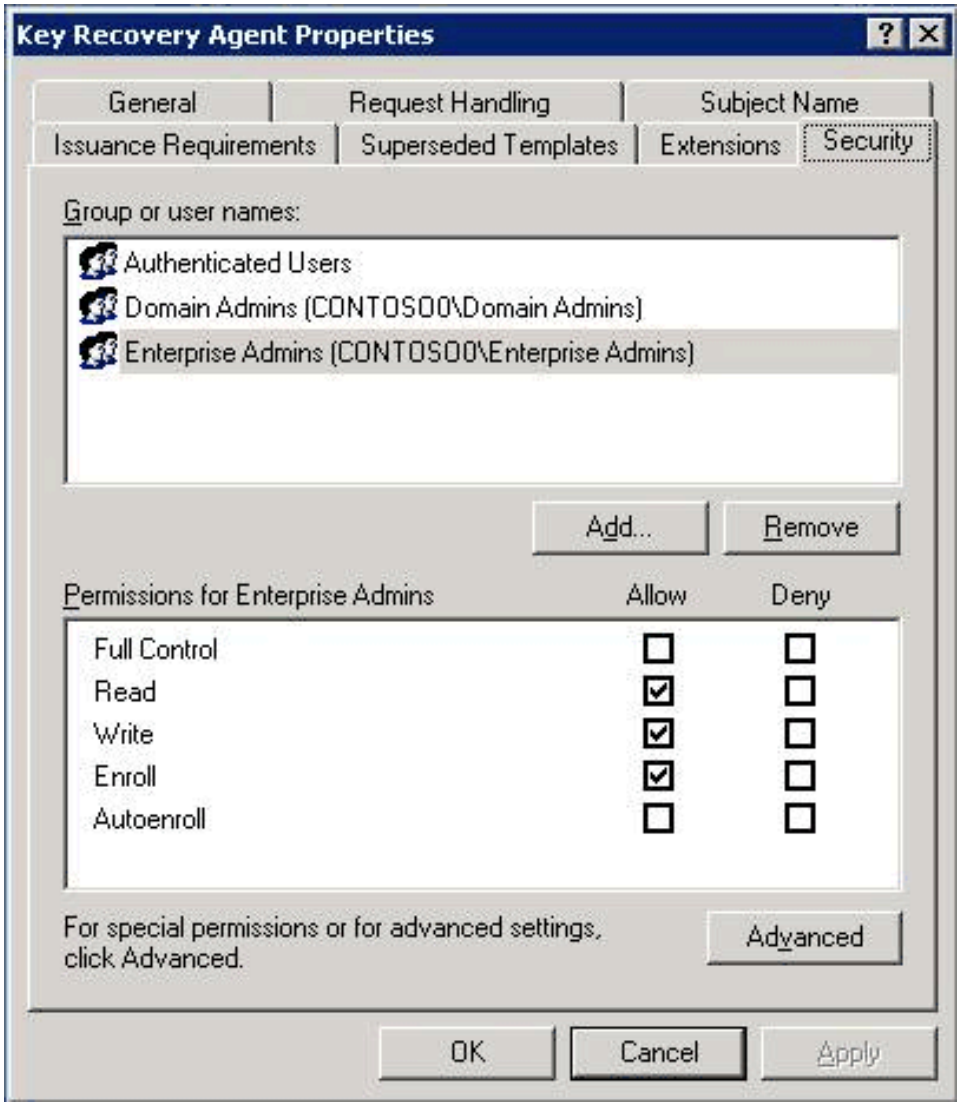

Figure 16: Key Recovery Agent Template Properties

6.Add the appropriate user(s) or group(s) with both Read and Enroll permission.
7.Click **OK** to close the dialog box.

Next, the Certification Authority must be configured to issue this type of certificate.

Certificate Template Permissions

For a user or a computer to enroll for a certificate template, it must have appropriate permissions [access control entries (ACEs)] set on the template in Active Directory. A user or computer must have both Enroll and Read permissions to enroll for a selected certificate template. The Read permission allows the template to be discovered by the user and the Enroll permission is enforced by the enterprise CA when a user requests a certificate for a selected template. The enterprise CA must also have Read permissions on a template to enumerate the template in the directory and issue certificates based on that template. Normally, the enterprise CA is included in the Authenticated Users group, which has Read permissions by default on a template.

The Full Control permission is given to Enterprise Administrators by default on installation of a fresh Windows Server 2003 domain. If a domain has been upgraded from Windows 2000, Enterprise Administrators will not have this permission by default and the Full Control permission allows a user to set or modify the permissions on a selected template.

The Autoenroll permission is set on a template when a user or computer wants to automatically enroll for a selected certificate template. The Autoenroll permission is needed in addition to the Enroll permission for a user to enroll for a given certificate template.

The Write permission allows a user to modify the contents of a certificate template. Note that only a version 2 certificate with a Windows Server 2003 schema may be modified and version 1 certificate templates may only have the ACLs modified.

Smart Card Support

Smart cards are supported for use in conjunction with KRA certificates. It may be necessary to use a smart card and CSP that supports at least an 8-KB smart card to enroll for a KRA certificate on a smart card. If a smart card does not have adequate memory to support a KRA certificate, the following error will be generated on enrollment.

Error: 0x80100028

An attempt was made to write more data than would fit in the target object

All recovery operations are supported using a Smartcard. The system will prompt the recovery agent to insert an appropriate Smartcard when the key is needed to decrypt the recovery BLOB.

Configuring an Enterprise CA to Issue KRA Certificates

An Enterprise CA must be configured to issue a KRA certificate.

To configure an EnterpriseCA to issue a KRA certificate

1. On the **Administrative Tools** menu, open the **Certification Authority** snap-in.
2. In the console tree, expand **Certification Authority**, and then click **Certificate Templates**.
3. Right-click the **Certificate Templates** node, click **New**, and then click **Certificate Template to Issue**.
4. In the **Select Certificate Template** dialog box, click **Key Recovery Agent**, and then click **OK**.
5. Close the **Certification Authority** MMC snap-in.
6. The last manual step is to add the Certificate Authority machine account to the Pre-Win2K Compatible Access group in every domain in which users will be using key archival. If this mandatory step is not performed, a CA Officer may not be able to manage groups of users. The Pre-Win2K Compatible Access group allows the CA to enumerate a user account and determine the group membership for CA Manager's capability.

Enrolling a User with a KRA Certificate

A user may enroll for a certificate with a CA by using the Certificates MMC snap-in or through the CA Web pages. In the case of the KRA template, the template is marked to be "pended" by the CA, which requires that the certificate request be approved first by a CA Administrator or a Certificate Manager before it is issued. Pended certificate requests may only be retrieved through the Web enrollment interface or through the auto-enrollment process. For more information, see Appendix B: Additional Information.

Important: It is strongly recommended not to automatically enroll KRA certificates as this may cause confusion for CA Administrators when automatic enrollment is unintentionally initiated resulting in additional KRA certificates in Active Directory.

KRA Web Enrollment

To enroll through a Web page

1. Connect to the CA using a Web URL, for example:
   http://<CA Machine Name>/Certsrv

   Note: The Microsoft Certification Authority uses an ActiveX control known as xenroll.dll, which can be downloaded to client browsers that support ActiveX controls. Windows XP and Windows Server 2003 clients both ship with the ActiveX control pre-installed.

   A Web site will open allowing you to request a certificate.

2. Select **Request a Certificate**.
3. On the next Web page, select **Advanced Certificate Request**.
4. Select **Create and submit a request to this CA**.
5. The next page will allow the user to select various configuration options, including the type of certificate to request. Choose **Key Recovery Agent** as the Certificate Template.

   Warning: A key size of 8192 or larger may take several hours to generate on the client. (Key pairs are always generated by the client CSP, not the CA.) This may slow public key operations on the CA when keys are archived. Key sizes of 2048 are much more reasonable for standard security needs; a key size of 8192 is used only as an example.

   The keys should be marked as exportable. This will enable an administrator (KRA) to export the private keys from the local store of the workstation to a disk or other medium for safe storage. Strong private key protection is also recommended, as this will require a password to be used when accessing the private key.
6. When finished, click **Submit**.

   The Web page will show that the request is being generated. When the key generation and request is complete, if the request had included Enable strong private key protection, a dialog box will appear showing that a protected item is being created (KRA private key).

Figure 17: Private Key Security Level Dialog Box

The dialog box will present the opportunity to set the security level on the private key. If the workstation will be used for KRA operations and the private key is to be kept in the protected store, it is recommended that the security level be set to High. This will ensure that access to the private key will require a separate password.

Figure 18: Private Key Security Level Dialog Box

7.Click **Next**.

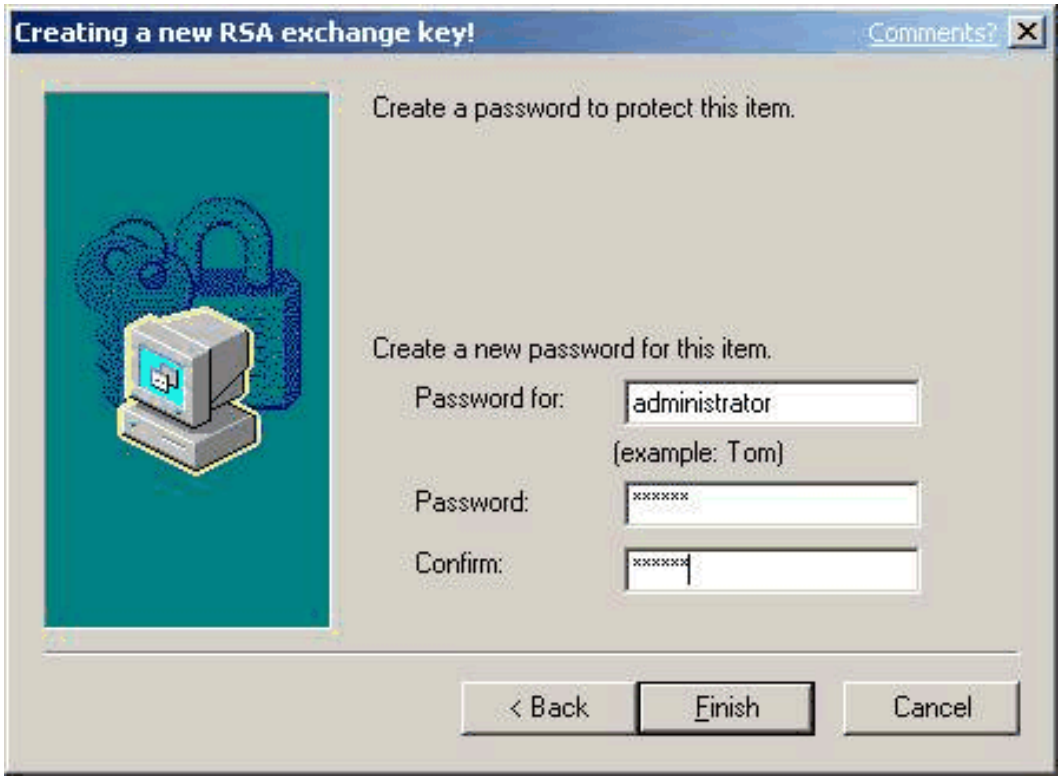

Figure 19: Private Key Security Level Dialog Box

8.Choose a password, confirm it, and then click **Finish**.

Figure 20: Private Key Security Level Dialog Box

9. Click **OK** to confirm. The Web page will appear and offer a link to install the certificate. If the Certification Authority is configured to set all certificate requests to pending, the user will have to return to the Web link to install the certificate after the CA administrator or a CA Officer has approved the request.

   Important: The user may have to return to the Web link using the same machine that was used to generate the request. This is due to the fact that certificate pending information is stored as a browser cookie. If auto-enrollment is enabled in Active Directory for the user, the user will not be required to actually return to the Web page. The auto-enrollment process will automatically retrieve the certificate for the user when available. For more information about the auto-enrollment process, see the Certificate Auto-Enrollment in Windows XP white paper in Appendix B: Additional Information.

   Important: If the issuing CA is in a hierarchy, a second option will appear on the Web page to download the entire path (certificate chain). You should choose to download the entire chain.

The default KRA certificate template requires that the certificate request be pended and not issued automatically. When a certificate request is pended, a CA Officer (Certificate Manager) must manually issue the certificate, assuming that the request was valid. Pending certificate requests can be issued through the Certification Authority MMC and by selecting the pending request node.

The Certificate Manager (or administrator of the server, if role separation is not enabled) can right-click the certificate request and choose to issue or deny the request. For more information about Certificate Managers, see Configuring Certificate Managers.

After the certificate has been issued, the user (KRA) can return to the Web pages to retrieve the pending request.

The user should select **View the status of a pending certificate request**.

The Web page will display all the pending requests for that user that have been requested from that machine. As mentioned previously, this is managed through Web browser cookies. The user should select the appropriate certificate.

The last page allows the user to install the selected certificate and private key into the local **MY** store of the user.

Configuring the CA to Allow Key Archival

This section details the steps required to configure the CA to allow key archival.

Note: If the Certification Authority is enabled to enforce role separation, only a CA Administrator may configure KRAs on a CA. Role separation is enabled through the registry and only a local server administrator may configure the registry. The easiest way to enable the CA for role separation is to use the certutil.exe command-line tool:

certutil -setreg ca\RoleSeparationEnabled 1

It is necessary to stop and start certificate services for the setting to take affect.

Note: Certutil.exe and other tools may be installed on a Windows XP Professional machine by installing the Administrative Tools (adminpak.msi) that are found in the \i386 directory on all Windows Server 2003 CD-ROM media.

Enabling a Key Recovery Agent

To enable a KRA

1.Log on as Administrator of the server or CA Administrator, if role separation is enabled.
2.On the **Administrative Tools** menu, open **Certification Authority**.
3.In the console tree, select the CA.

4.Right-click the CA name, and then click **Properties**.
5.Click the **Recovery Agents** tab.

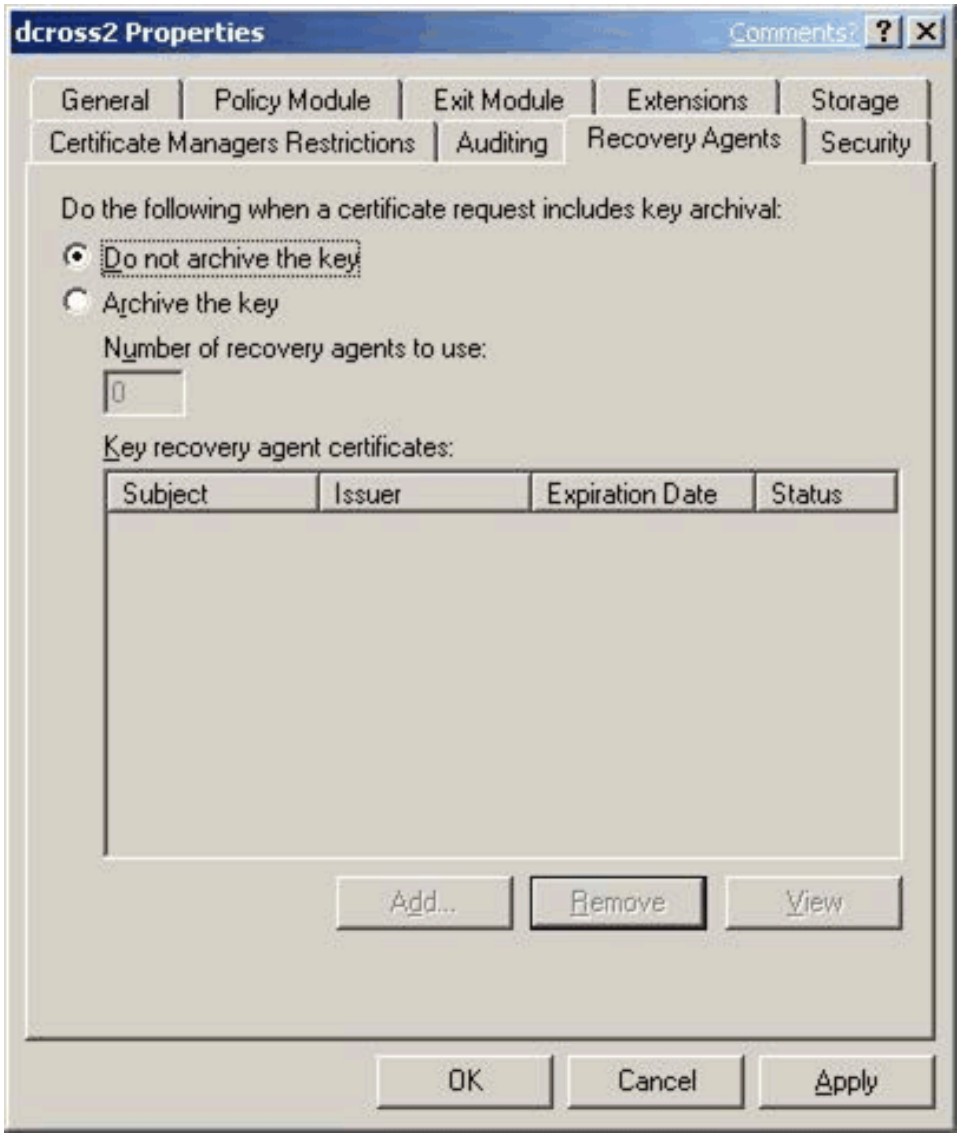

Figure 21: Certificate Server Properties Dialog Box Recovery Agents Tab

6.To enable key archival, click **Archive the key**.

7. By default, the CA will only use one KRA. However, a KRA certificate must first be selected for the CA to begin archival. To select a KRA certificate, click **Add**.

   The system will find valid KRA certificates and display the available KRA certificates.

   KRA certificates are normally published to Active Directory by an Enterprise CA when enrollment occurs. KRA certificates are stored under the KRA container in the Public Key Services branch of the configuration partition in Active Directory. Since a CA may issue multiple KRA certificates, each KRA certificate will be added to the multi-valued userAttribute attribute of the CA object.
8. Select one certificate and click **OK**. You may view the highlighted certificate to ensure that you have selected the intended certificate.

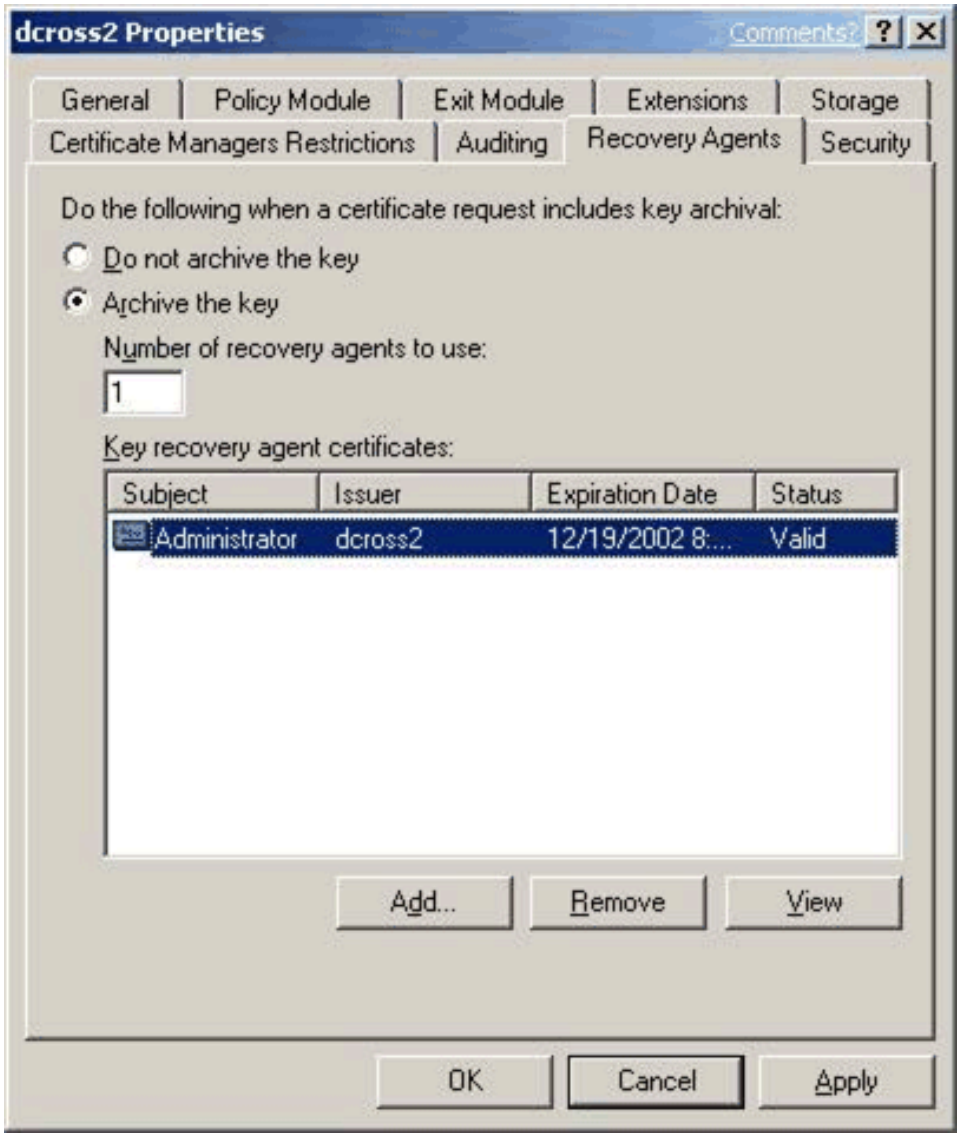

Figure 22: Certificate Server Properties Dialog Box Recovery Agents Tab

9. After one or more KRA certificates have been added, click OK to enable key archival on the CA. However, Certificate Services must be stopped and started to enable the use of the selected KRAs. KRA certificates are only processed at service start.

Configuring Certificate Managers

To recover the private keys of a user, the CA enforces that a person be a Certificate Manager (if defined) and also holds a private key for a valid KRA certificate. As a best practice, most organizations separate these two roles. By default, the CA Administrator is a Certificate Manager for all users unless otherwise explicitly defined. A KRA is not necessarily a CA Officer (Certificate Manager). They may be segmented as separate roles. A KRA is defined as a person who holds a private key for a valid KRA certificate.

A CA Officer is defined as a Certificate Manager who has the security permission on the CA to Issue and Manage Certificates. The security permissions are configured on a CA using the Security tab on the CA Properties in the Certification Authority MMC snap-in.

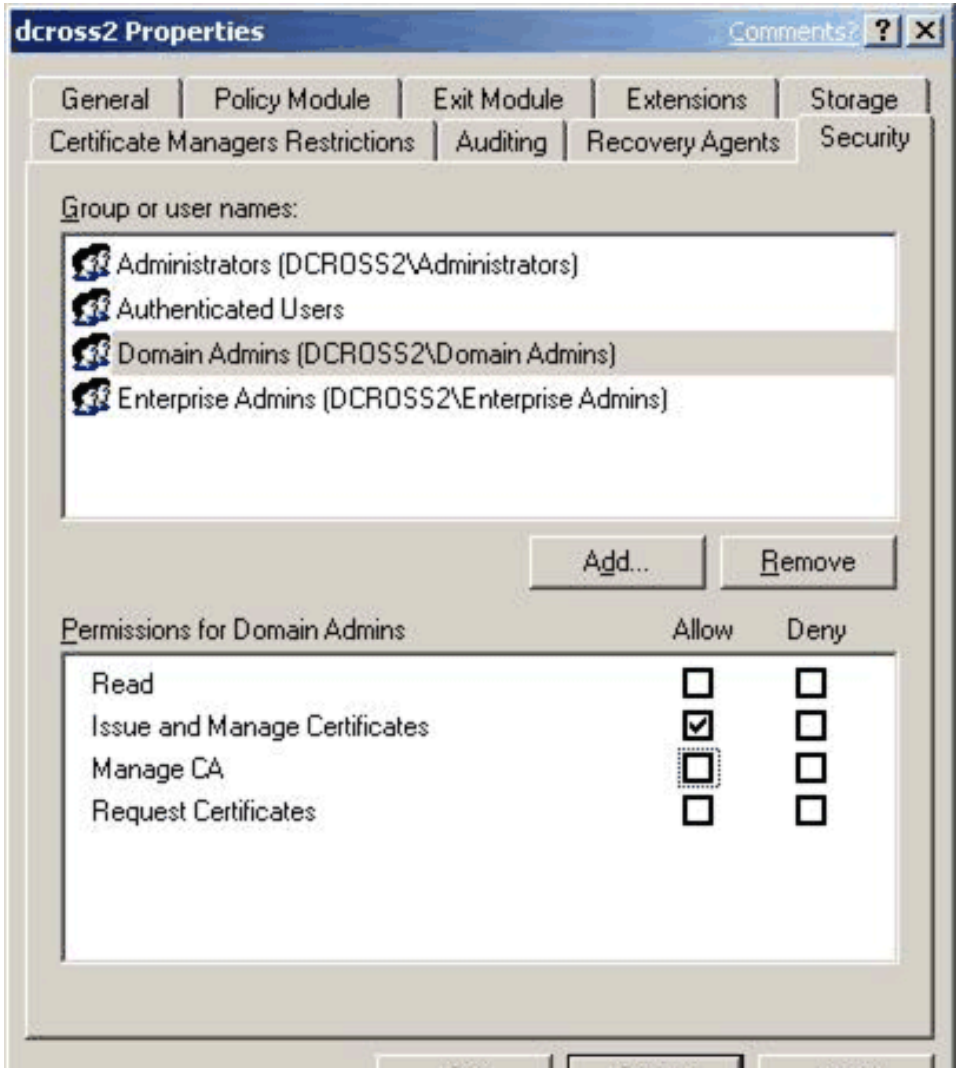

Figure 23: Certificate Server Properties Dialog Box Security Tab

A CA Administrator can define more granular CA Officer Roles on a Certification Authority by using the Certificate Managers Restrictions tab. By default, any user or group that has the permission to Issue and Manage Certificates is also a CA Officer and can issue, revoke, or export a recovery BLOB for any other user who has been issued a certificate from that CA.

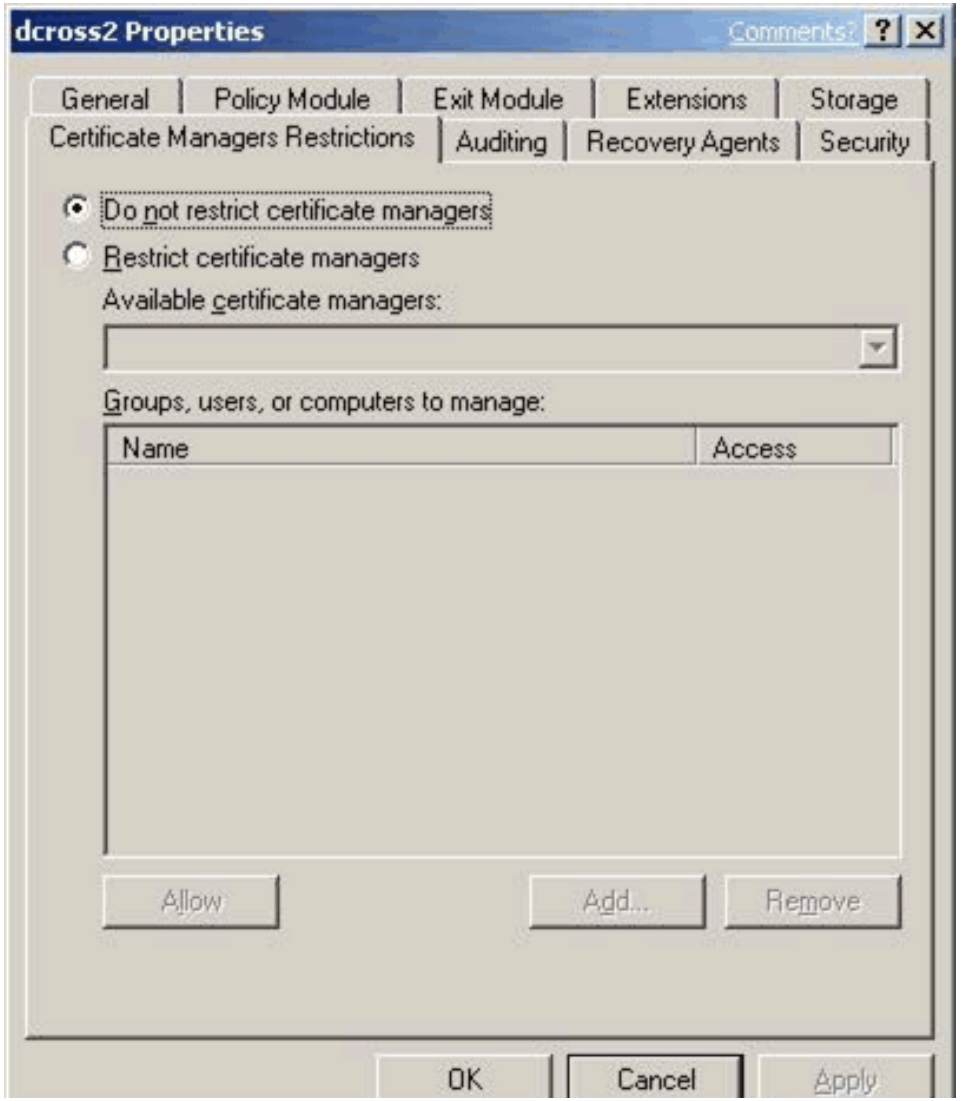

Figure 24: Certificate Server Properties Dialog Box Certificate Managers Restrictions Tab

A CA Administrator can define that individual CA Officers have the ability to issue, revoke, and recover certificates for defined sets of user(s) and group(s) by selecting the Restrict certificate managers option. Once the option has been selected, a CA Administrator may define CA Officers' roles as required.

Important: Once this is complete, it is also necessary to add the machine account of the CA to the *Pre W2K Compatible Access Group* of every domain in which users will be archived and recovered. This is necessary to provide proper group membership enumeration for role restriction enforcement when performing recovery functions. Certificate Managers will not be able to recover users unless this task is performed first in every user domain.

Configuring User Templates

To enable a user template for key archival, select the template that should enforce key archival and select the Archive subject's encryption private key check box. Once the check box has been selected, the CA will always enforce key archival for that template.

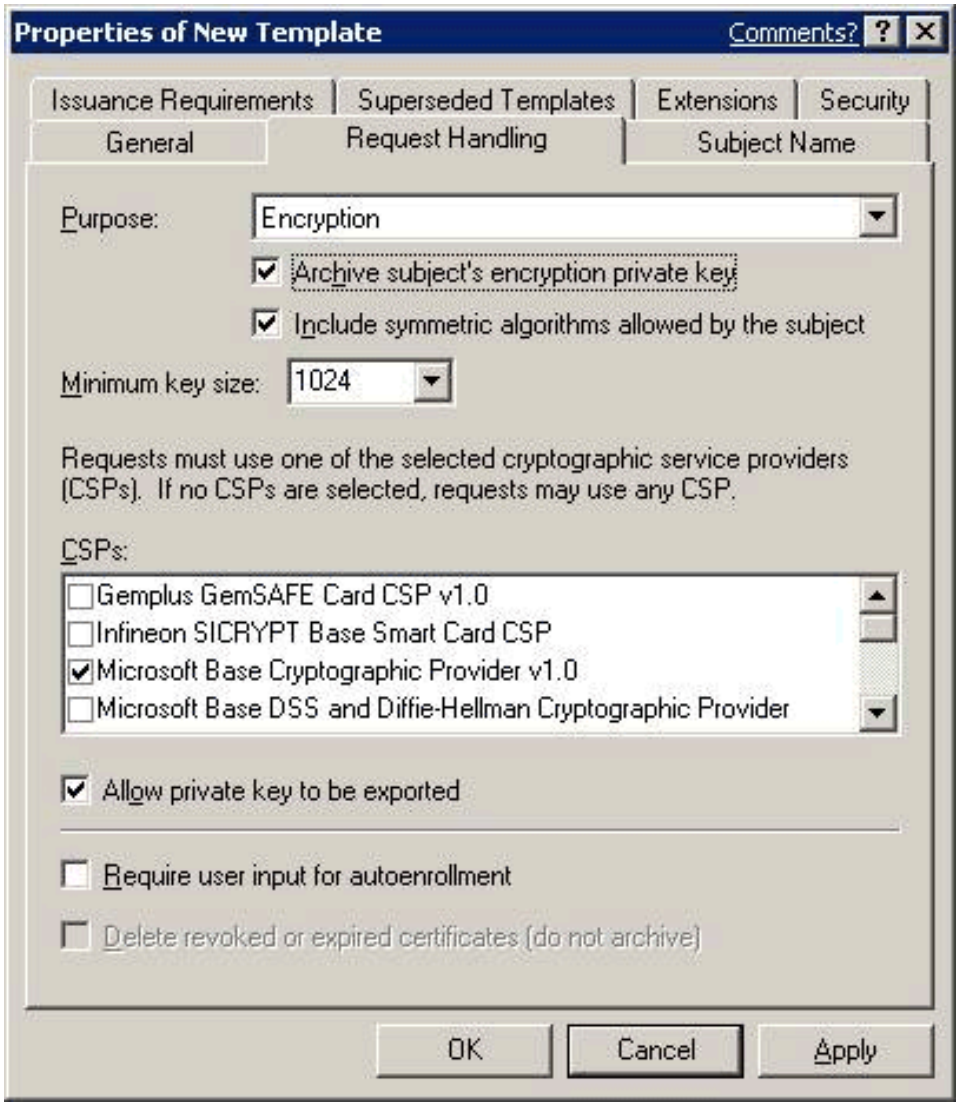

Figure 25: Certificate Template Properties Dialog Box Request Handling Tab

Note: A Windows Server 2003 CA will not allow archival of a key marked for signature only and will reject the request, even if sent programmatically to the CA.

Top of page

Migrating Exchange KMS to Windows Server 2003 CA

The following are the summary steps for migrating Exchange 2000 Server KMS to a Windows Server 2003 Certificate Authority.

1.If running Exchange 5.5 KMS, upgrade to Exchange 2000.
2.Configure Windows Server 2003 CA for key archival.
3.Ensure that the certificate is available for database migration.
4.Enable the foreign certificate import option on the CA, if necessary.
5.Export the Exchange KMS database.
6.Import the Exchange database into the CA.

Important: Before migrating KMS to a Windows Server 2003 CA, it is important to consider the version 1 CRL that is published by the Exchange KMS for Outlook clients in the Exchange Global Address List (GAL). If KMS is migrated to a Windows Server 2003 CA, the v1 certificates can no longer be revoked and it is recommended that a KMS migration is only performed when all V1 certificates are expired and/or are no longer being issued by KMS. If x.509 version 3 certificates are being issued by KMS with a Windows 2000 CA, the existing CA will need to be maintained to publish CRL (s) until all the original certificates issued by KMS have expired.

Creating an Export Certificate on the Certificate Server

When a KMS migration to a Windows Server 2003 CA is performed, the export file from the KMS must be encrypted with a public key certificate and then subsequently decrypted by the Windows Server 2003 CA. The CA may or may not have an encryption certificate available to be used for this process. It is absolutely critical that an encryption certificate and private key be installed in the machine store (local machine) of the CA to facilitate KMS migration. Since the Certificate Server process runs as SYSTEM, any encryption certificate and private key available in the machine store may be used.

To view the certificates installed in the local machine store, open the Certificates MMC console for the local machine and view the certificates under the Personal store. A Secure Sockets Layer (SSL) or machine authentication certificate will suffice for use in this scenario. The certificate corresponding to the private key that will be used should be manually exported and made available during the KMS migration process. For more information about certificate enrollment and exporting certificates, see the Windows Server 2003 help files. If importing a certificate and key to be used by the CA (*.pfx file), ensure that the certificate is marked for export when importing on the CA. Otherwise, the CA may not be able to use the key and certificate for key import purposes.

Important: The export certificate used by the KMS should not have a key size greater than 1024 bits as this may cause errors on import to the Windows Server 2003 CA.

Important: A Windows Server 2003 always has an Exchange certificate (encryption certificate) available for the purpose of key archival. Do not attempt to use this certificate for the purpose of migrating the KMS database as it will not be usable by the CA for this purpose.

Enabling Foreign Certificates Import

If the KMS contains x.509 version 1 certificates and private keys, and/or if the KMS was not configured to use the same CA with Windows 2000, the foreign certificate import option must be enabled on the Windows Server 2003 CA.

Foreign Certificate Import

By default, a CA does not allow certificates (or keys) to be imported on the CA that were issued by another CA. A CA must be enabled to accept certificates and keys into the database that were issued by a foreign CA. (An Exchange 5.5 KMS issuing version 1 certificates is also considered a foreign CA.)

To import a foreign CA

1.Run the following command in a command-prompt window on the CA.

certutil –setreg ca\KRAFlags +KRAF_ENABLEFOREIGN

2.Once that has competed, restart the certificate server service.

Important: When foreign certificates are being imported on a CA, the –f switch must be used with certutil to inform the CA that the keys and certificates will be foreign. The command line would be as follows:

Certutil.exe –f –importKMS [name of import file]

Exporting Users' Keys from Exchange 2000 KMS

Warning: Before an export of data from the KMS occurs, a full backup of the KMS should be performed and validated before continuing. An export of data from a KMS is destructive and will remove the keys from the KMS database.

Important: If the KMS or the CA is online when the export occurs, the KMS will attempt to revoke all version 3 certificates that are exported. If this occurs, it is important to re-enroll all users immediately with the Windows Server 2003 CA to allow continued S/MIME encryption operations. Otherwise, take the CA offline, so the KMS export operation will not revoke the

existing certificates.

To perform the export operation on the KMS

1. Start the Exchange System Manager.

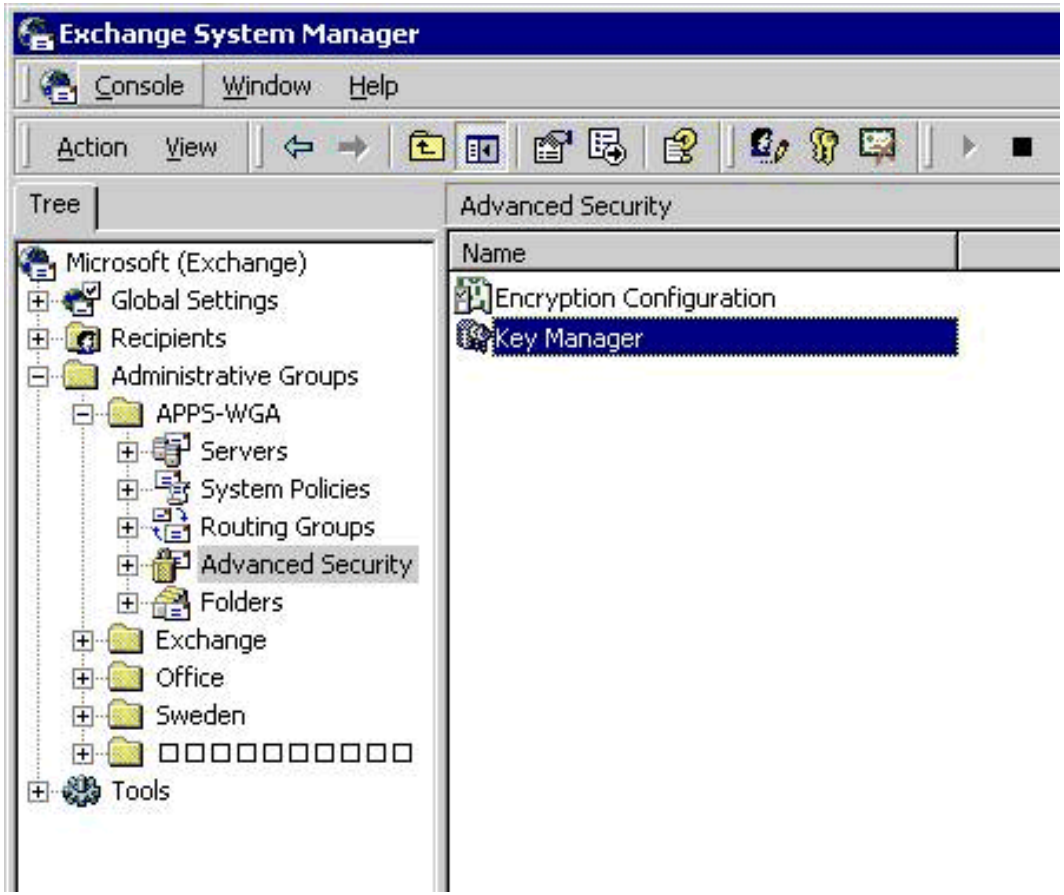

Figure 26: Exchange System Manager MMC Snap-In

2.Point to the **Advanced Security** node, right-click **Key Manager**, click **All Tasks**, and then click **Export Users**.

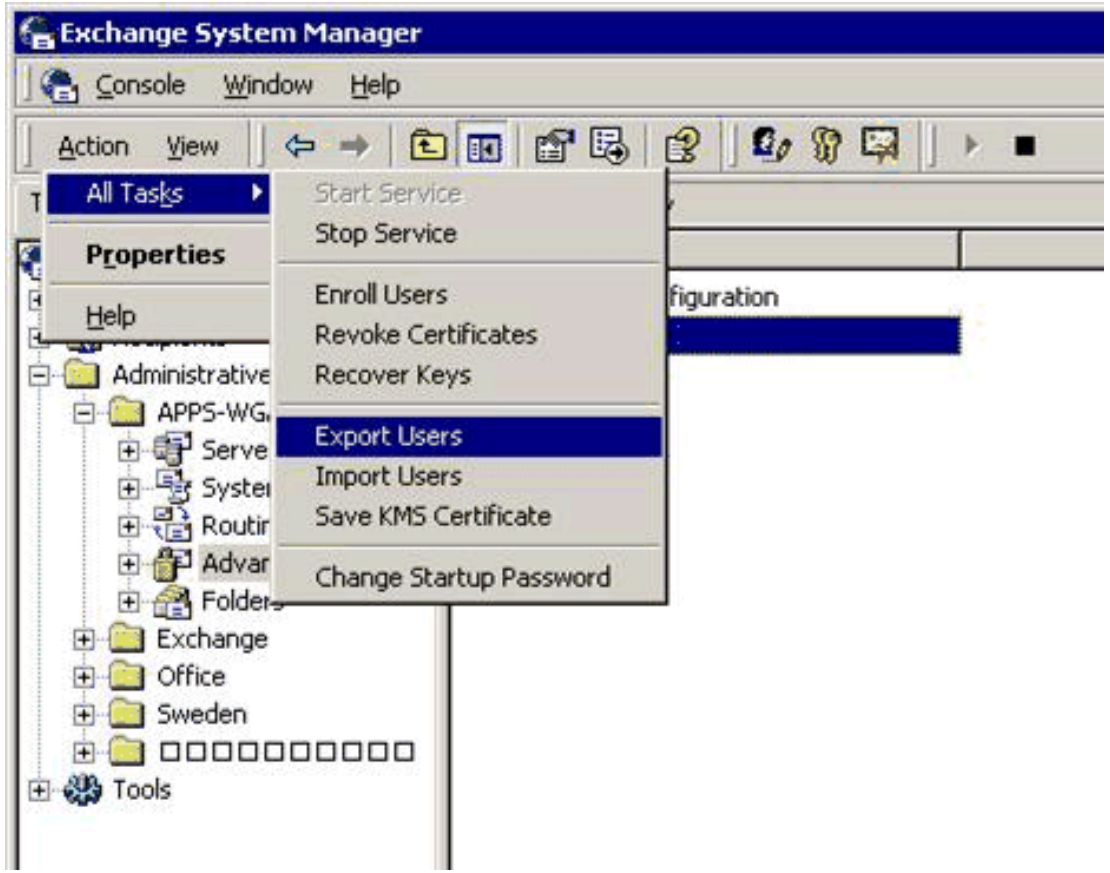

Figure 27: Exchange System Manager Snap-In

3. In the **Key Management Service password** box, type the password (the default password for KMS is “password”), and then click **OK**.

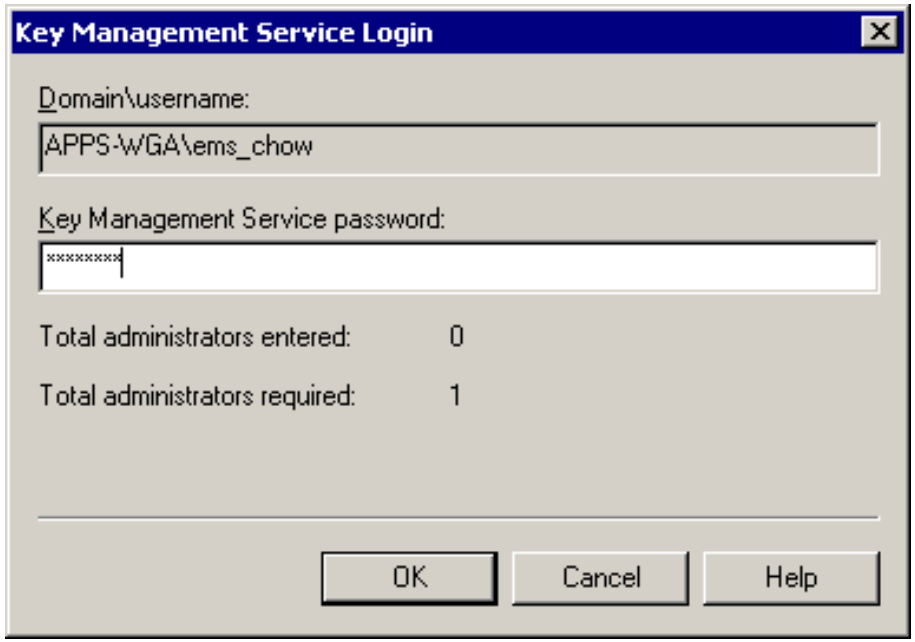

Figure 28: Exchange Management Service Login Dialog Box

The Exchange KMS Key Export Wizard will start.

4. Click **Next**.

Figure 29: Exchange KMS Key Export Wizard Welcome Page

5. Click **Browse** to select the Certificate that will be used to encrypt the export file. This is the certificate file created in the previous section.

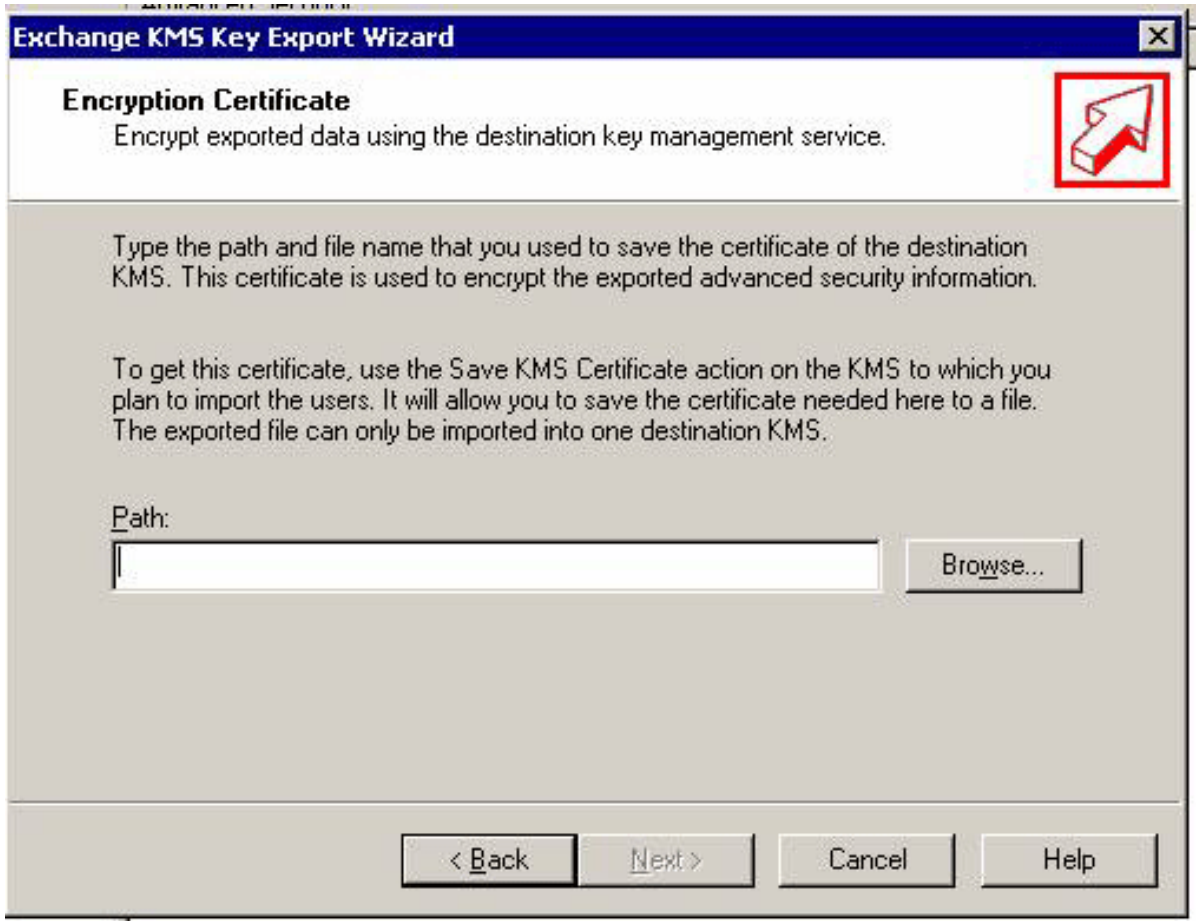

Figure 30: Exchange KMS Key Export Wizard to Select the Encryption Certificate

6. Browse for the certificate that will be used to encrypt the export file to the CA. This is the certificate created in the previous section. Click **Open**.

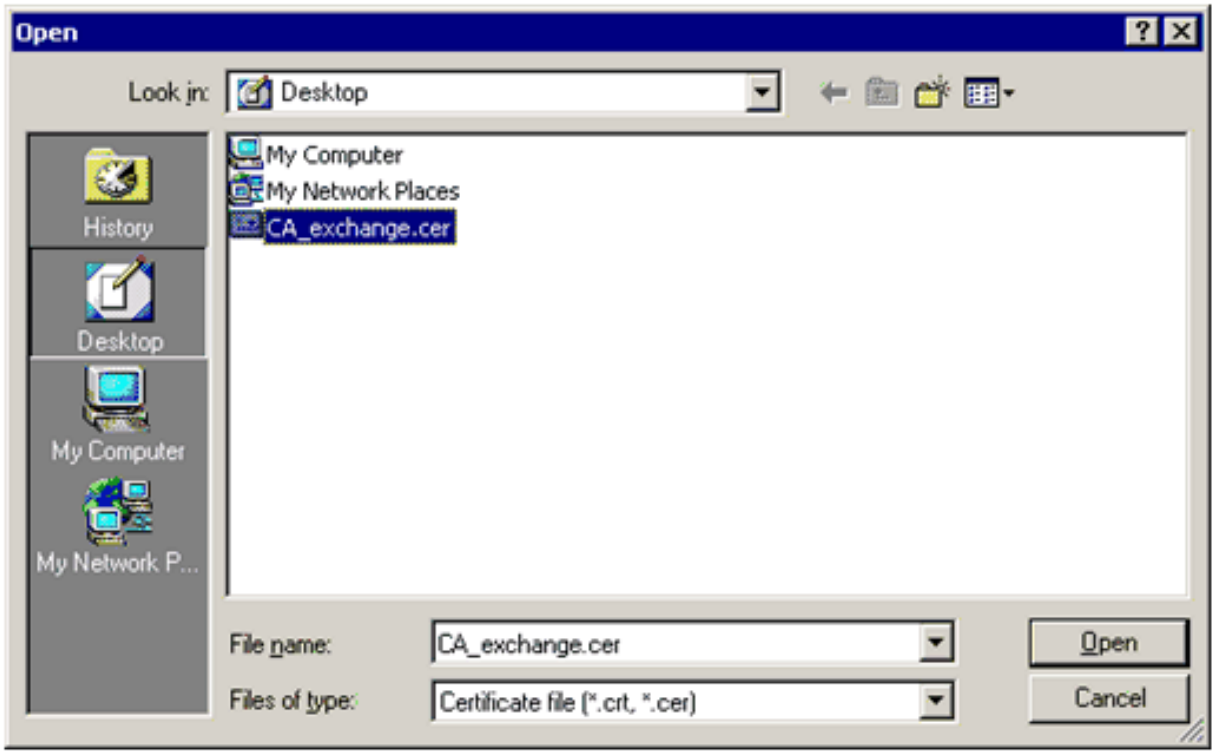

Figure 31: Exchange KMS Key Export Wizard to Open the Certificate File
See full-sized image

7.On the Encryption Certificate screen, click Next.

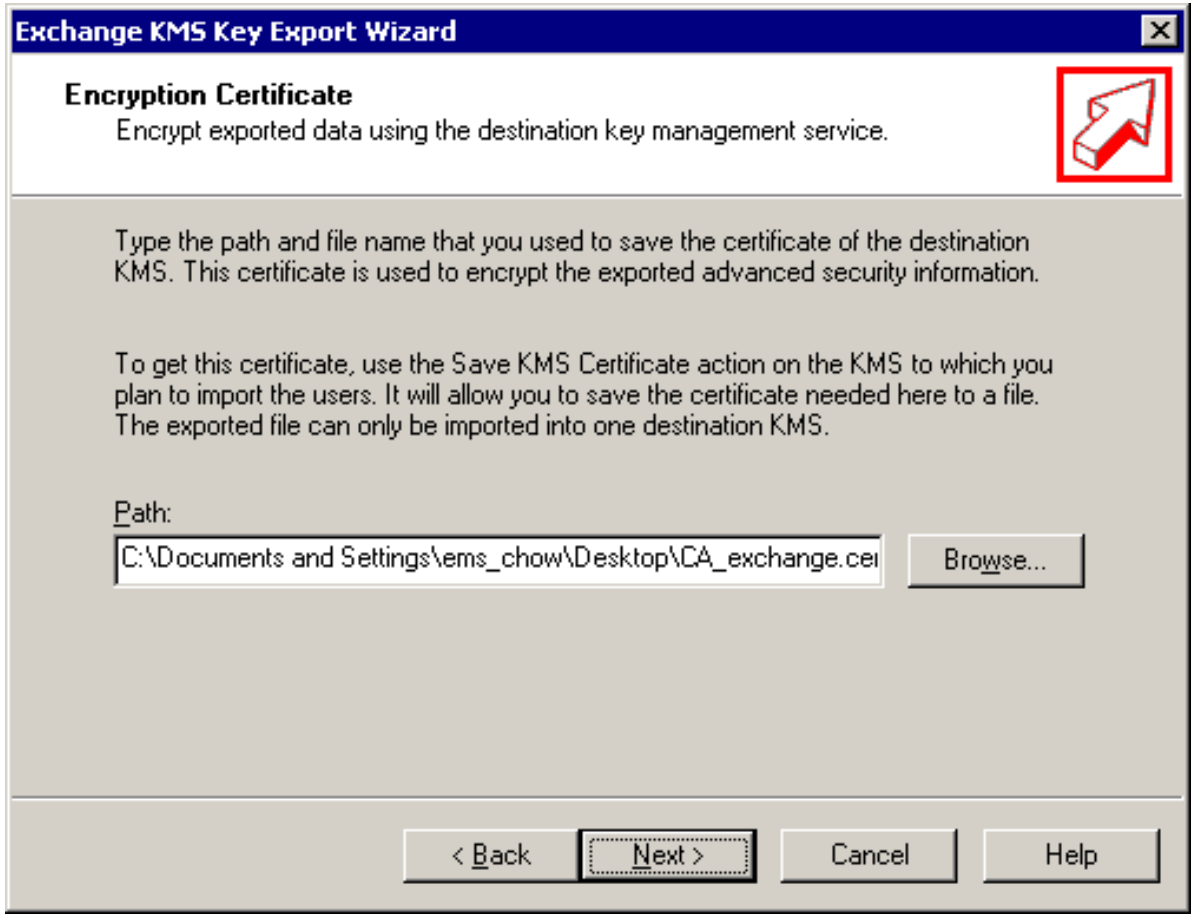

Figure 32: Exchange KMS Key Export Wizard to Select the Encryption Certificate

8.When this screen appears, use **Windows Explorer** to find and open the certificate that you chose from the screen in step 6. You will need to validate this certificate with the **Exchange KMS Key Export Wizard**.

9. Copy the first eight characters from the Certificate thumbprint field in the certificate chosen to encrypt the KMS export file (Figure 33).

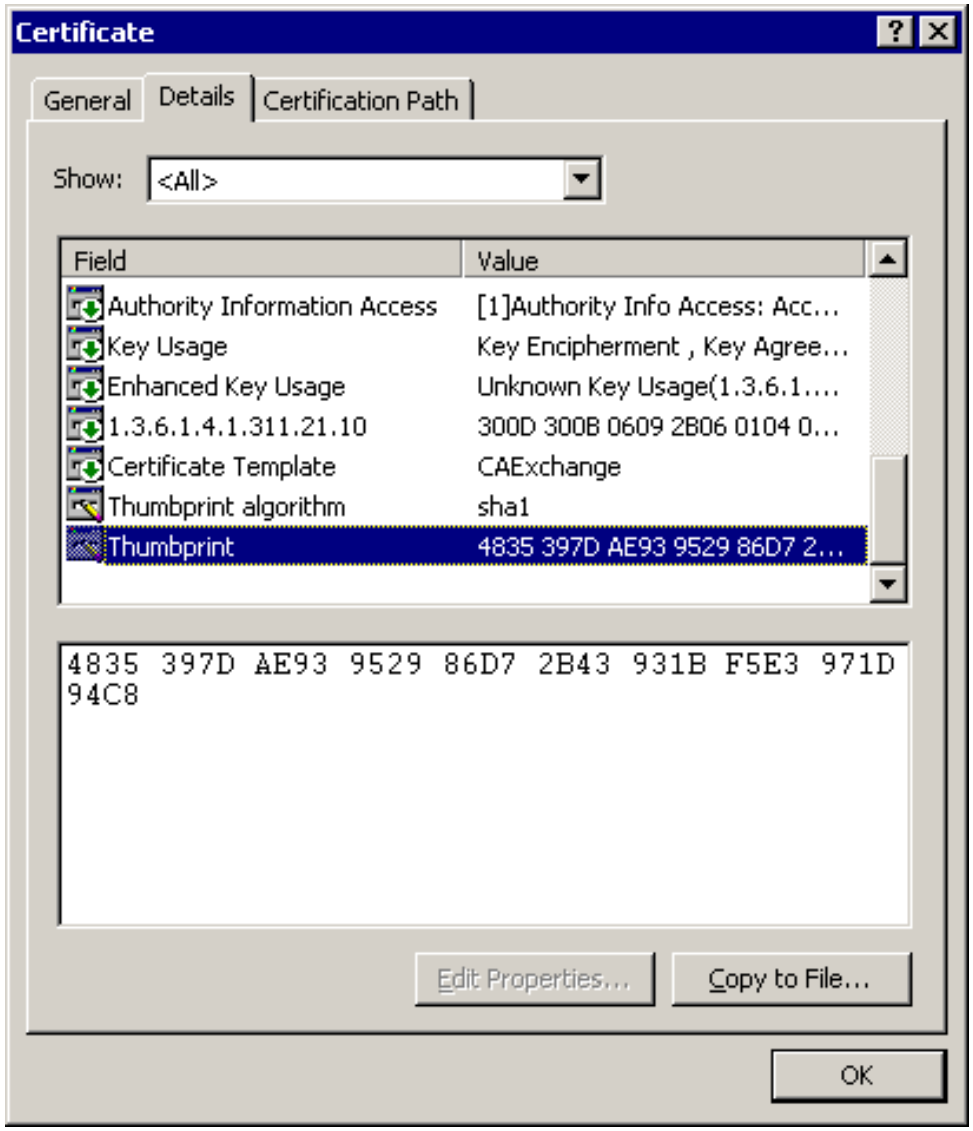

Figure 33: Certificate Properties Dialog Box

10. Type the first eight characters of the certificate thumbprint in the **Thumbprint** field (Figure 34), and then click **Next**.

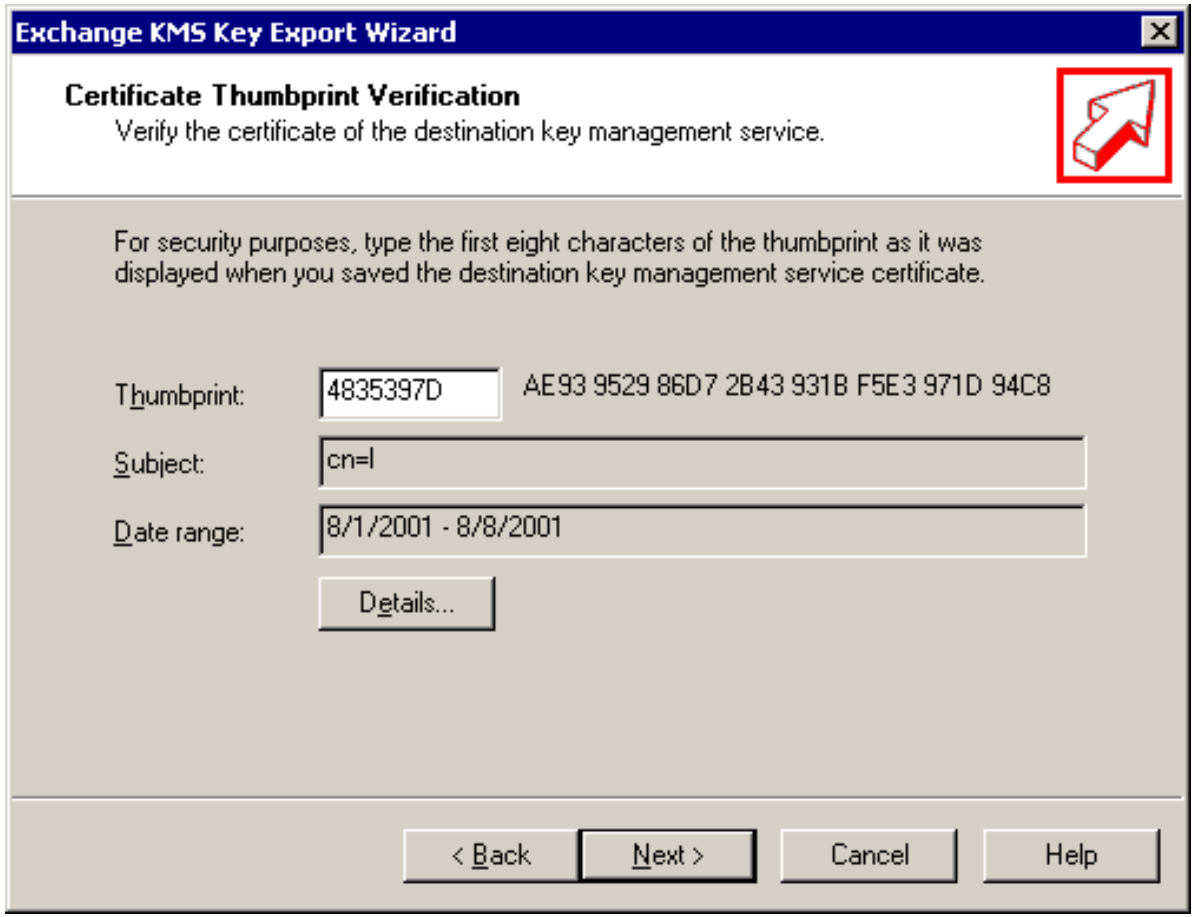

Figure 34: Exchange KMS Key Export Wizard to Verify the Certificate Thumbprint

11. Type the name of the export file (Figure 35). Do not type in a path, only the file name. It will be saved in the following location by default. This is based on the default installation for Exchange.

    C:\program files\exchsrvr\KMSDATA

    This file will not have an extension.

12.Click **Next**.

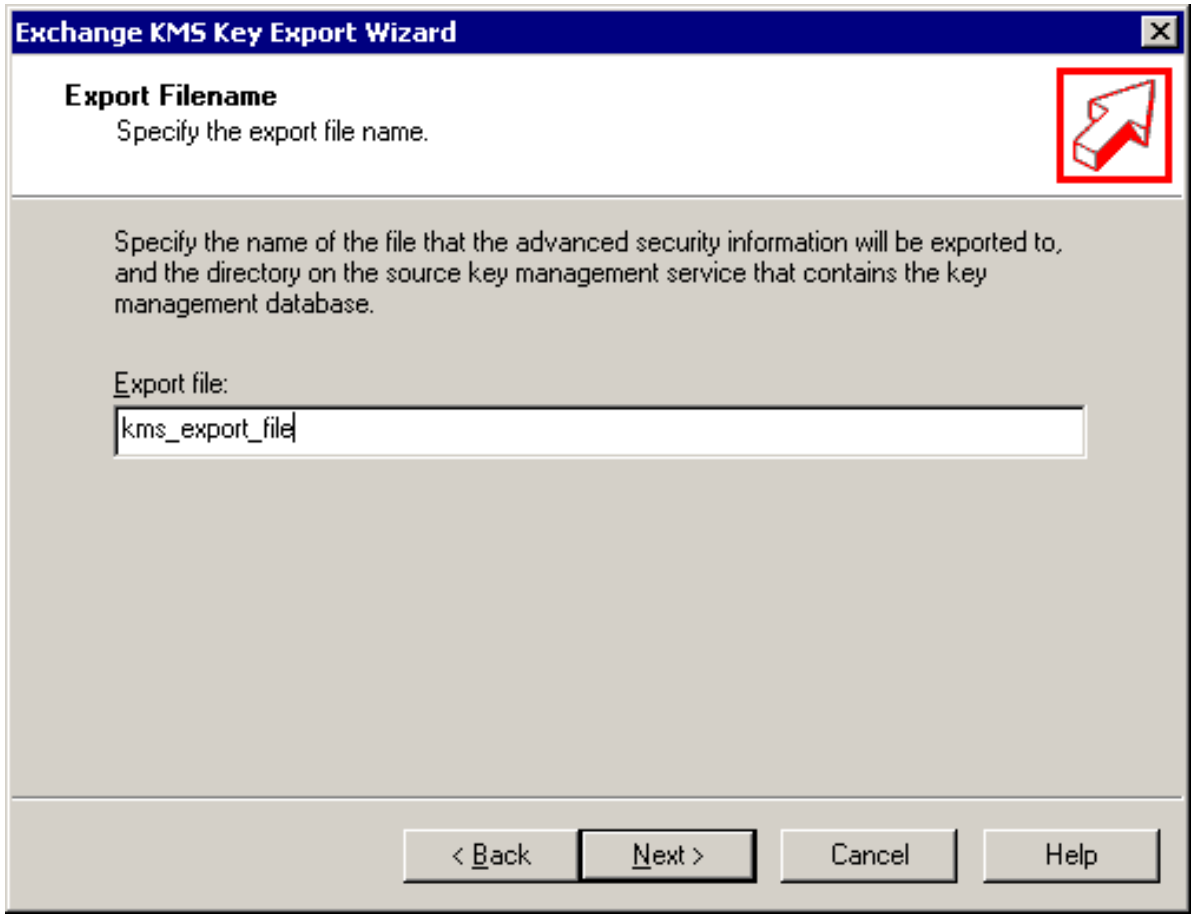

Figure 35: Exchange KMS Key Export Wizard Export Filename Page

13.You may select an alphabetic list of users or select by mailbox store, server, or administrative group.

Figure 36: Exchange KMS Key Export Wizard User View Selection Page

14.In this case, select all of the administrative groups, and then click **Next**.

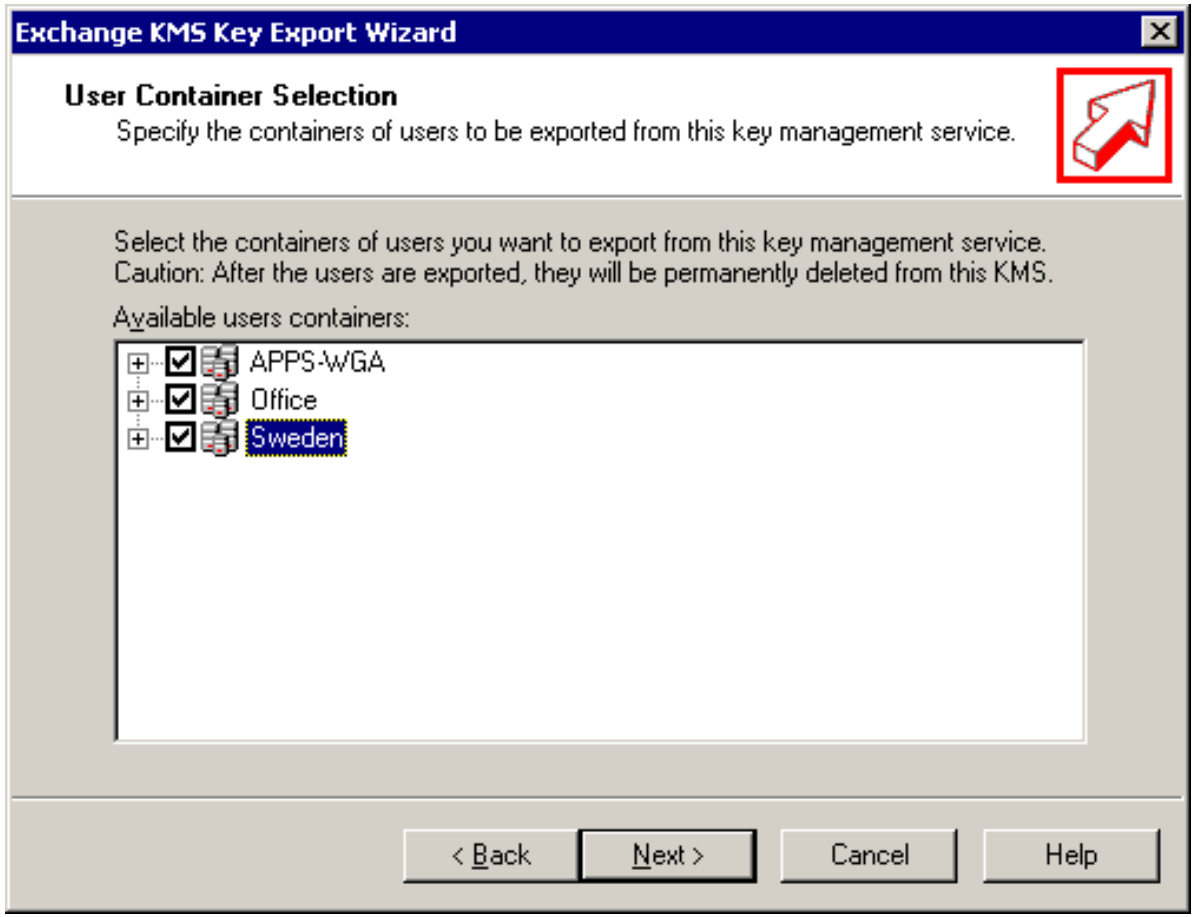

Figure 37: User Container Selection Page

15.To start the Export process after selecting the users or administrative group(s), click **Next**.

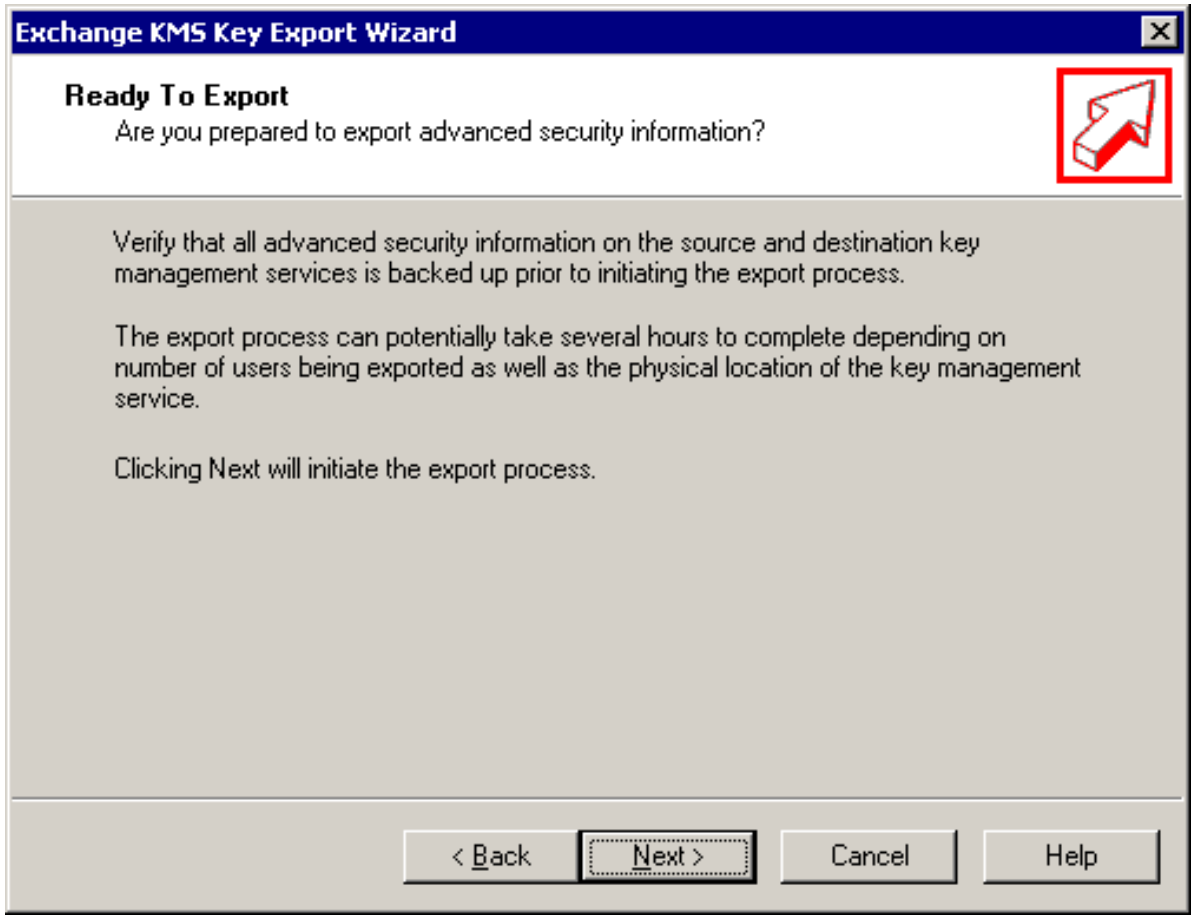

Figure 38: Ready to Export Page

The records will be exported. On average, approximately 100 records will be exported a minute. The actual performance will vary depending on the hardware configuration.

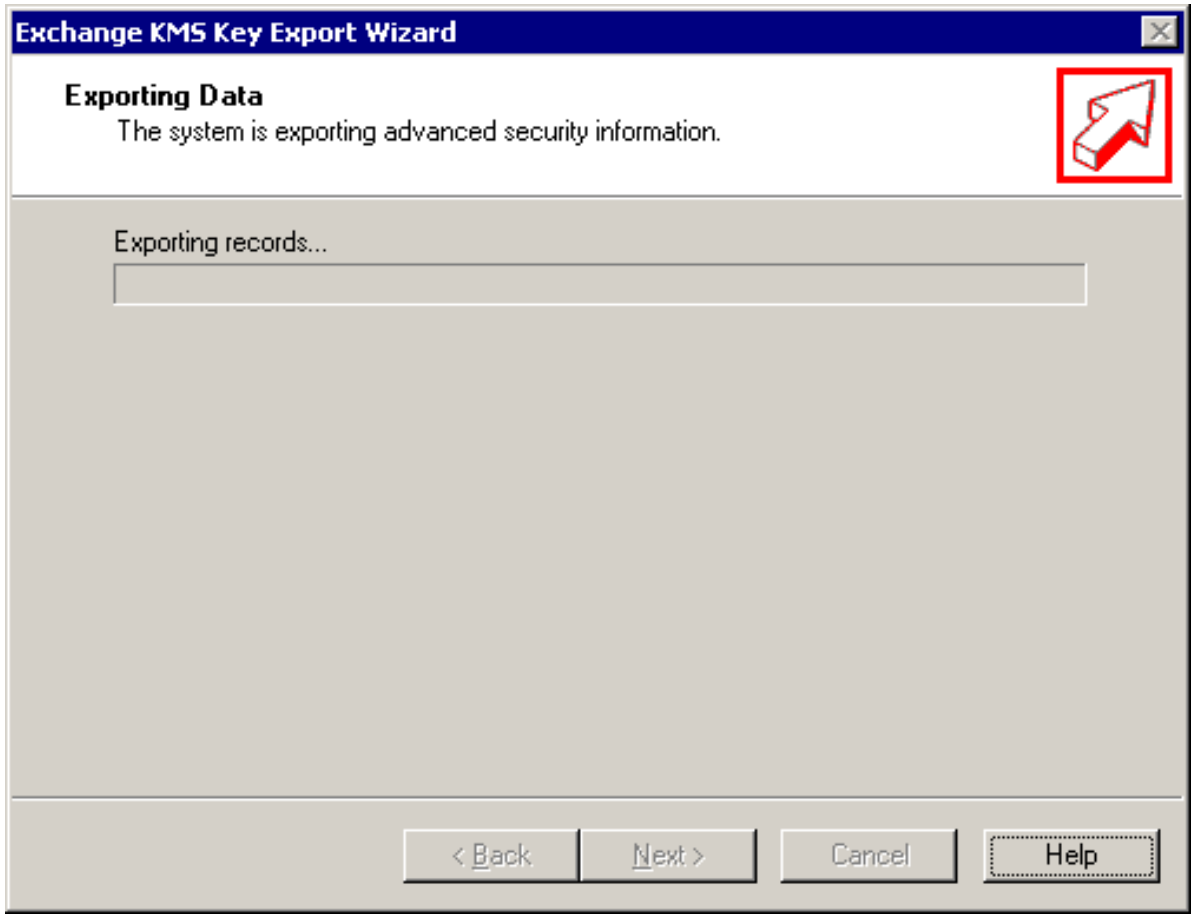

Figure 39: Export Progress Page

16. When complete, click **Next**.

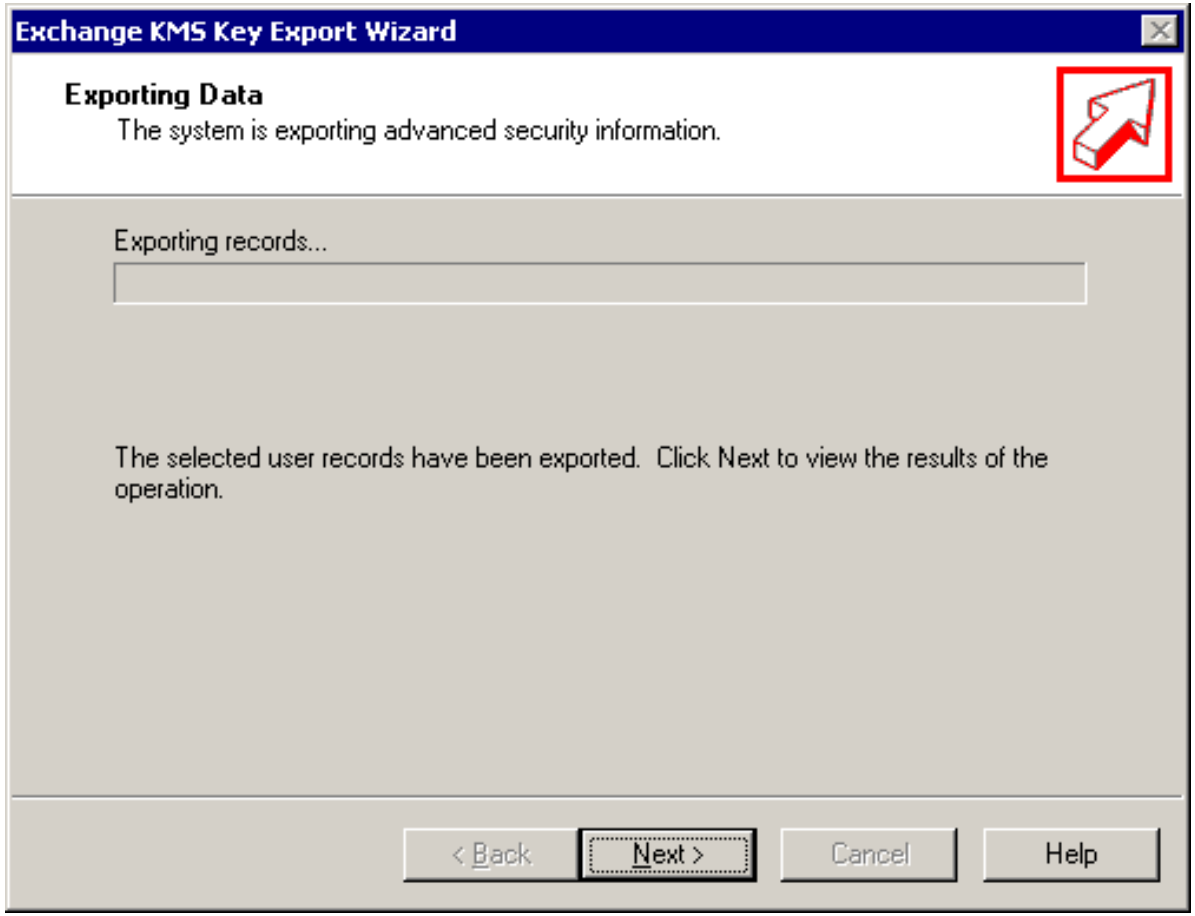

Figure 40: Export Progress Page

17. The results will be displayed. Click **Finish**.

Note: If large numbers of users are exported, KMS may generate multiple export files and split the exported keys across the multiple files. In this case, all export files should be re-imported to the new CA.

Figure 41: Completing the Export Page

The export file will be located in the following folder.

C:\program files\exchsrvr\KMSDATA

18. Copy the KMS export file to the Certificate server that will accept the import file.

Importing Users' Keys to the Certificate Server

The Windows Server 2003 CA allows not only key archival, but also certificate and key importation to the CA database. Certificate and key importation is important in providing migration services for Exchange KMS as well as for providing migration and escrow operations for certificates that were enrolled using a third-party CA. The Windows Server 2003 CA supports both certificate import as well as key import. Certificate import does not require that key archival be enabled on the CA, but key import does.

To import users' keys to the Certificate Server

1.Log in to the Certificate Server as the CA Administrator.
2.Open a command prompt window.
3.Change to a directory containing the KMS import file.
4.Run the following command.

CertUtil.exe –f –importkms <name of export file>

The output will indicate the status of the import process and the number of user keys imported and archived to the CA. The number of imported user keys should match the output from the KMS. The following is a sample successful output.

```
Processing KMS exports from:
    O=microsoft,C=US

KMS export file signature verifies
Lock box opened, symmetric key successfully decrypted
.................................

Users: 6

Ignored signature certificates: 25
Certificates with keys: 17
Foreign certificates imported: 17
Certificates imported: 17

Keys: 17
Keys archived: 17
```

```
CertUtil: -ImportKMS command completed successfully.
```

Top of page

Troubleshooting

Troubleshooting KRA Configuration

This section identifies a number of common mistakes in configuring the KRAs on a CA. The most common error in archiving user private keys on a CA is that the CA is either not configured for key archival or does not have any valid KRA certificate(s) added.

1. The number of recovery agents required by the CA must be less than or equal to the number of available KRA certificates. If an invalid number of recovery agents is entered, the following error message will appear.

Figure 42: Invalid Number of Recovery Agents Error Message

2.If an error occurs in trying to validate the KRA certificate when Certificate Services is started, the Recovery Agents tab on the Certification Authority will show that the selected KRA certificate is invalid. This can occur due to a corrupted certificate, corrupted registry entry, deleted certificate, revoked certificate, and so on. Figure 43 shows an example of a corrupted certificate or registry entry on the Recovery Agents tab as shown in the Status column of the selected KRA.

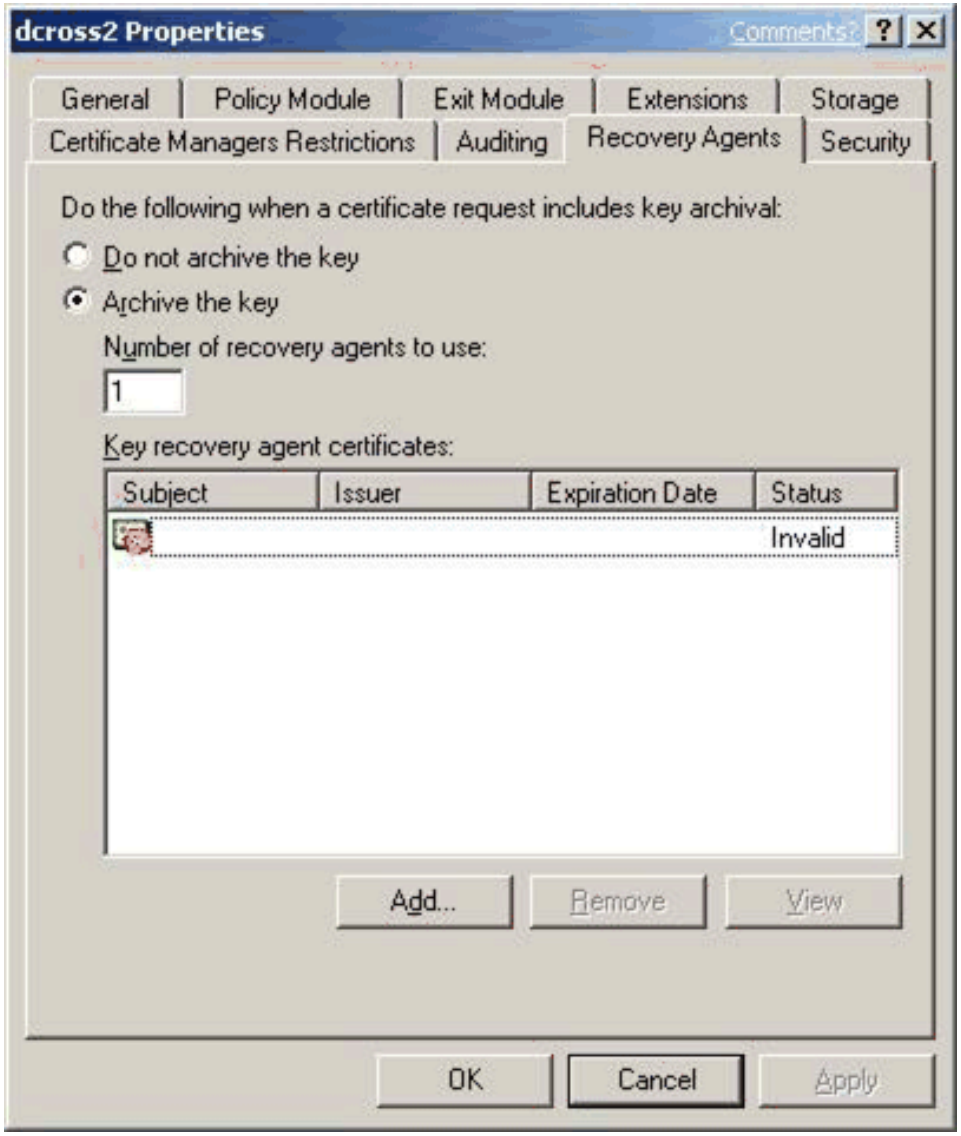

Figure 43: Invalid KRA Certificate on the Recovery Agents Tab

3. Similarly, a revoked KRA certificate will also show an error on the Recovery Agents tab when Certificate Services is stopped and started. The error will be displayed in the status column of the KRAs certificates listing.

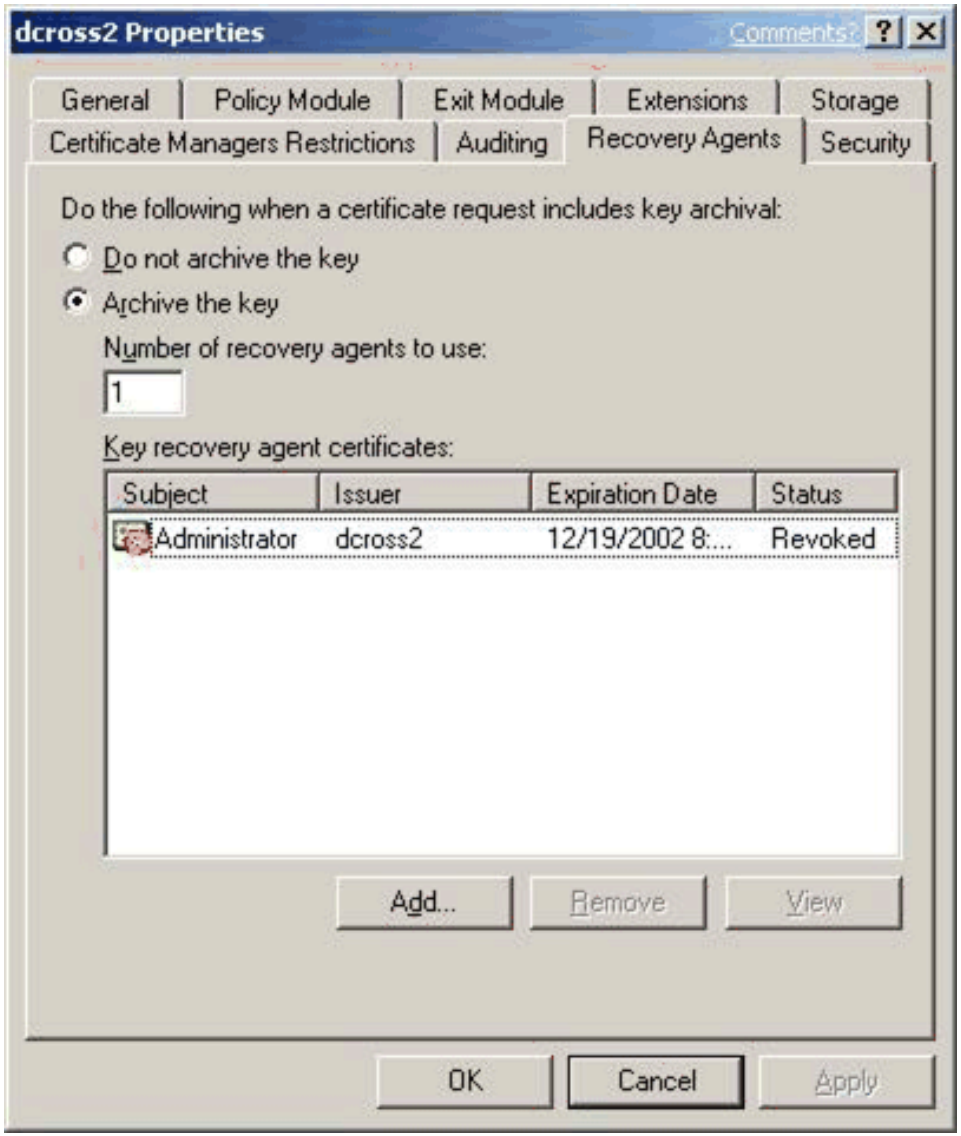

Figure 44: Revoked KRA Certificate on the Recovery Agents Tab

Loading KRA Certificates

When certificate services starts on a Certification Authority, the CA attempts to load the KRA(s) defined by the CA Administrator. If the CA is unable to load one or more KRA(s), event log messages will be generated; however, certificate services will continue to start. If the CA is unable to load a KRA(s) successfully as defined by a CA Administrator, the CA will deny all requests for key archival and not issue any certificates that have key archival defined in the certificate template. The following event log messages may appear in the Certification Authority's Application Log when an error occurs in loading KRA certificates. The event log messages indicate that action is required by a CA Administrator to properly configure or reconfigure KRAs.

Event Type: Error

Event Source: CertSvc

Event Category: None

Event ID: 83

Date: 12/20/2000

Time: 8:24:24 AM

User: N/A

Computer: SERVER1

Description:

Certificate Services encountered an error loading key recovery certificates. Requests to archive private keys will not be accepted. The system cannot find the file specified.
0x80070002 (WIN32: 2)

This is a global error that can be caused by one of several conditions.

•The Certification Authority cannot open the KRA store on the local machine.
•The Certification Authority cannot find a corresponding certificate in the KRA store on the local machine that matches the hash of a certificate set in the registry as a KRA.

•The registry has been edited incorrectly or is corrupted.
•The count of KRA certificate hashes in the registry equals zero.
•A certificate hash in the registry corresponds to a certificate in the KRA store that is not a KRA certificate type.
•The KRA certificates are revoked, expired, or invalid.

Event Type: Error

Event Source: CertSvc

Event Category: None

Event ID: 82

Date: 12/27/2000

Time: 9:05:25 AM

User: N/A

Computer: SERVER1

Description:

Certificate Services could not load any valid key recovery certificates. Requests to archive private keys will not be accepted.

This error is usually caused when none of the certificates specified in the user interface (UI) (which corresponds to the registry) is a valid KRA certificate. This event log message is usually accompanied by the previous global event log message.

Event Type: Error

Event Source: CertSvc

Event Category: None

Event ID: 84

Date: 1/24/2003

Time: 08:49:27

User: N/A

Computer: SERVER1

Description:

Certificate Services will not use key recovery certificate 6 because it could not be verified for use as a Key Recovery Agent. CN=User1, OU=Users, DC=nwtraders, DC=com The revocation function was unable to check revocation because the revocation server was offline. 0x80092013 (-2146885613)

This error usually occurs when the CA receives an error when retrieving the CRL to check the status of the KRA certificate.

Event Type: Error

Event Source: CertSvc

Event Category: None

Event ID: 98

Date: 1/24/2003

Time: 08:49:28

User: N/A

Computer: SERVER1

Description:

Certificate Services encountered errors validating configured key recovery certificates. Requests to archive private keys will

no longer be accepted.

Event Type: Error

Event Source: CertSvc

Event Category: None

Event ID: 85

Date: 12/27/2000

Time: 9:05:25 AM

User: N/A

Computer: SERVER1

Description:

Certificate Services ignored key recovery certificate 0 because it could not be loaded. Cannot find object or property. 0x80092004 (-2146885628)

This error usually occurs when a specific KRA certificate cannot be found in the KRA store on the local machine of the Certification Authority. Specifically, a KRA certificate has been specified in the UI or registry, and the certificate has been deleted or corrupted in the KRA store. This event log message is usually accompanied by a more global event log message.

KRA Certificate Status

When certificate services starts on a Certification Authority, the CA attempts to load the configured KRA(s). The CA must validate the status of each KRA certificate. If the CA is unable to retrieve a current CRL for each KRA certificate, the CA will not be able to load and use that KRA.

The following event log message will be logged in the application event log of the CA.

Event Type: Error
Event Source: CertSvc
Event Category: None
Event ID: 84
Date: 1/12/2001
Time: 11:47:23 AM
User: N/A
Computer: SERVER1
Description:
Certificate Services ignored key recovery certificate 1 because it could not be verified for use as a Key Recovery Agent. CN=User1, OU=Users, DC=nwtraders, DC=com The revocation function was unable to check revocation because the revocation server was offline.
0x80092013 (-2146885613)

Importing Exchange KMS Export File

The Windows Server 2003 CA may fail during the importation of the KMS data file if the key size used for the export certificate is greater than 1024 bits in size. The Windows Server 2003 CA may fail with the following message when a key size of greater than 1024 bits is used.

Processing KMS exports from:
   CN=Certificate Authority, OU=Test, O=Contoso, C=US

KMS export file signature verifies
CertUtil: -ImportKMS command FAILED: 0x80070057 (WIN32: 87)
CertUtil: The parameter is incorrect.

User Enrollment Errors

A user certificate request for a template that requires key archival will be denied if one of the following conditions exists.

•No KRA has been defined on the CA.
•No KRA can be successfully loaded. (KRA certificates are revoked, expired, and so on.)
•The minimum number of KRA certificates defined by the CA Administrator cannot be loaded.

If the user enrolls through a Web page, the following text will display on the Web page.

Your request failed. An error occurred while the server was processing your request.

Contact your administrator for further assistance.

Request Mode:

newreq - New Request **Disposition:**

(never set) **Disposition message:**

(none) **Result:**

Cannot archive private key. The certification authority is not configured for key archival. 0x8009400a (-2146877430) **COM Error Info:**

CCertRequest::Submit Cannot archive private key. The certification authority is not configured for key archival. 0x8009400a (-2146877430) **LastStatus:**

Cannot archive private key. The certification authority is not configured for key archival. 0x8009400a (-2146877430) **Suggested Cause:**

No suggestions.

If enrolling through the MMC, the following error will be displayed.

Figure 45: Incorrect Certificate Request Error Message
See full-sized image

The CA will also log the following error to the application event log of the CA.

Event Type: Error

Event Source: CertSvc

Event Category: None

Event ID: 21

Date: 1/12/2001

Time: 4:23:39 PM

User: N/A

Computer: SERVER1

Description:

Certificate Services could not process request 16094 due to an error: Cannot archive private key. No valid key recovery agent is defined for this certification authority. 0x8009400b (-2146877429). The request was for NWTRADERS\User1.

If the CA is unable to retrieve a current CRL for the CA itself or any of its parent CA(s), it will be unable to issue a certificate when a user submits a request. If the CA does not have a valid CRL for itself, the following error message will be displayed in the application event log of the CA.

Event Type: Warning

Event Source: CertSvc

Event Category: None

Event ID: 53

Date: 1/6/2001

Time: 11:24:05 AM

User: N/A

Computer: SERVER1

Description:

Certificate Services denied request 1471 because the revocation function was unable to check revocation because the revocation server was offline. 0x80092013 (-2146885613). The request was for CN=user1, OU="Test", O="NWTraders", L=Redmond, S=WA, C=US, E=user1@nwtraders.com. Additional information: Denied by Policy Module

Certificate Template Not Supported by the CA

If a user tries to enroll with a CA for a template that is not supported by that CA, the following event log message will be entered in the CA application event log.

Event Type: Warning

Event Source: CertSvc

Event Category: None

Event ID: 53

Date: 1/16/2001

Time: 2:07:02 PM

User: N/A

Computer: SERVER1

Description:

Certificate Services denied request 8 because the requested certificate template is not supported by this CA. 0x80094800 (-2146875392). The request was for NWTRADERS\Administrator. Additional information: Denied by Policy Module The request was for certificate template (1.3.6.1.4.1.311.21.8.4144336743.1329436349.2065260953.3989445610.1.27) that is not supported by the Certificate Services policy.

Client CSP Does Not Permit Key Export

For the client enrollment process to generate and send a private key to the CA, the key must be marked as exportable when the key is generated. If the certificate template is not set to allow key exportable or if the third-party CSP (if applicable) does not support exportable keys, enrollment will fail and the enrollment wizard will return an error that the key is not exportable. Third-party CSPs may report varying errors, such as “catastrophic failure”, when this occurs. If a Windows 2000 or Windows ME client performs enrollment with key archival, the following error may appear if the key is not marked for export.

0x80090009 - NTE_BAD_FLAGS

Note: If the CSP supports the one-time flag for key archival, known as (CRYPT_ARCHIVABLE), the key export flag is not required. The Microsoft default software CSPs support this flag. However, Windows 2000 and Windows ME clients do not support this flag and must allow the key to be exported for enrollment to work with key archival.

Certificate Authority CSP Not Supported for Key Archival Functions

If a software or hardware CSP is not capable of performing the symmetric and public operations for encrypting the private key(s) of users in the CA database, the CA will log an event in the application event log:

Event Type: Warning

Event Source: CertSvc

Event Category: None

Event ID: 86

Date: 12/27/2001

Time: 8:13:54 AM

User: N/A

Computer: NORTHWIND5

Description:

Certificate Services could not use the provider specified in the registry for encryption keys. The keyset is not defined. 0x80090019 (-2146893799)

For more information, see the Help and Support Center at *http://go.microsoft.com/fwlink/events.asp*

Event Type: Warning

Event Source: CertSvc

Event Category: None

Event ID: 88

Date: 12/27/2001

Time: 8:13:54 AM

User: N/A

Computer: NORTHWIND5

Description:

Certificate Services switched to the default provider for encryption keys.

For more information, see the Help and Support Center at http://go.microsoft.com/fwlink/events.asp

To verify which CSP the CA is using for encryption operations associated with key archival, run the following command from the CA.

```
Certutil –getreg ca\EncryptionCSP\Provider
```

Certificate and Key Import Issues

If the CA has not been configured for key archival or if the CA has not been configured for foreign key import, the following error will occur when attempting to import a KMS export file or a foreign key import. Note that the keys were not archived in the following message.

```
Processing KMS exports from:
    O=microsoft, C=US
KMS export file signature verifies
Lock box opened, symmetric key successfully decrypted
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.
CertUtil: Invalid Signature.


Users: 6
```

```
Ignored signature certificates: 25
Certificates with keys: 17
Certificates not imported: 17

Keys: 17
Keys not archived: 17
CertUtil: -ImportKMS command completed successfully.
```

Troubleshooting Key Recovery Issues

Unable to Recover User

If a CA performing key archival is also enabled for role separation with specific Certificate Manager restrictions, a Certificate Manager may not be able to recover a user certificate until the machine account of the CA has been added to the Pre W2K Compatible Access Group of the domain in which the recover user belongs. This is a necessary requirement for the CA to enumerate the group memberships of Certificate Managers and recovered users to ensure that proper restrictions are enforced.

Missing KRA Certificate in the CA Registry

If one of the recipient KRA certificates from the HKEY_LOCAL_MACHINE KRA certificate store on the Certification Authority is deleted, key recovery tools, such as certutil –getkey, will fail because the server cannot find the KRA certificate to include in the recovery BLOB. The following error message will be displayed when this error occurs.

```
certutil -getkey "1b 4a b7 1e 00 00 00 00 00 1d"
Querying server1.nwtraders.com\CA1.....

"server1.nwtraders.com\CA1"
   1b4ab71e00000000001d  CN="Users

Administrator"
CertUtil: -GetKey command FAILED: 0x80092004 (-2146885628)
CertUtil: Cannot find object or property
```

Note that the KRA certificate must be available in the registry on the CA, not the machine where the recovery tool(s) are used.

Top of page

Appendix A: Certificate Request Structure

This appendix provides additional detailed information about the key archival process regarding the certificate request structure.

ASN.1 Structure

A certificate request for key archival to the CA is a CMC Full PKIRequest message as specified in RFC 2797. The ASN.1 structure used by the Windows Server 2003 CA is demonstrated in Figure 46.

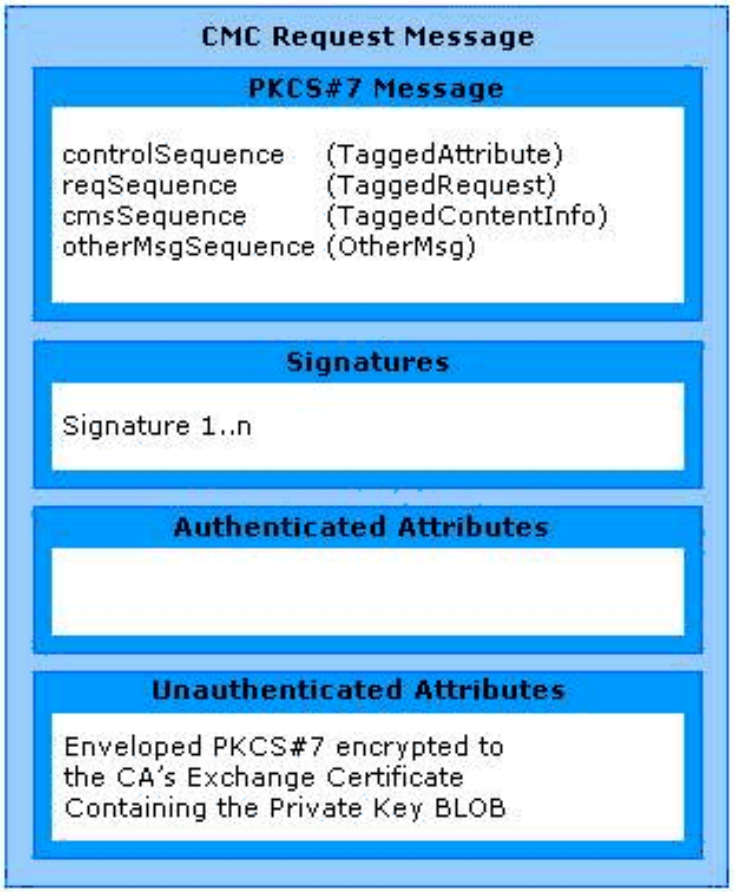

Figure 46: CMC Request Message

Understanding the PKCS #7 Message Content Structure

The first section of the CMC message contains a PKCS #7 message that has the relevant elements for generating a certificate request.

Understanding the controlSequence TaggedAttribute Element

The TaggedAttribute element in the message contains the following information.

•Extensions—The Extensions section of the TaggedAttribute element contains the following extensions.

 •Application Policies
 •Template Information
 •Key Usage
 •Enhanced Key Usage
•Attributes—The Attributes section of the TaggedAttribute element contains the following data.

 •Common Name
 •Template Name to be used
 •Hash of the encrypted private key BLOB
•Other request information

Understanding the reqSequence TaggedRequest Element

The reqSequence TaggedRequest element contains a nested PKCS #10 message. This message contains the user's public key in addition to other information relevant for generating the certificate.

Understanding the cmsSequence TaggedContentInfo Element

The cmsSequence TaggedContentInfo element can contain nested PKCS #7 and CMC messages. In a standard archival request, this element is not used.

Understanding the otherMsgSequence OtherMsg Element

Not Used

Understanding the Signatures Structure

The signatures section of the CMC message contains one or more signatures used to sign the request. The following is an example of the signatures section.

```
Signer Count: 1

Signer Info[0]:
Signature matches request Public Key
CMSG_SIGNER_INFO_CMS_VERSION(3)
CERT_ID_KEY_IDENTIFIER(2)
    0000  81 92 56 3a c4 31 f8 82  0c 54 c9 d0 98 4f d8 c5
    0010  34 63 9e cc
Hash Algorithm:
    Algorithm ObjectId: 1.3.14.3.2.26 sha1
    Algorithm Parameters: NULL
Encrypted Hash Algorithm:
    Algorithm ObjectId: 1.2.840.113549.1.1.1 RSA
    Algorithm Parameters: NULL
Encrypted Hash:
    0000  c1 ae 90 a7 a3 0b 52 66  ea c4 d0 04 17 2e 94 95
    0010  14 20 06 ...
```

Understanding the Authenticated Attributes Structure

The authenticated attributes section contains additional authenticated attributes, such as Content Type, Message Digest, and Client Information. The following is an example of the authenticated attributes section.

```
Authenticated Attributes[0]:
  3 attributes:
  Attribute[0]: 1.2.840.113549.1.9.3 (Content Type)
    Value[0][0]:
    Unknown Attribute type
    1.3.6.1.5.5.7.12.2 CMC Data
  Attribute[1]: 1.2.840.113549.1.9.4 (Message Digest)
```

```
    Value[1][0]:
    Unknown Attribute type
    Message Digest:
        5e 1f 0f f0 28 a4 fe 91 0d c2 2f 1a 18 78 7e 2e 10 7f 17 39
  Attribute[2]: 1.3.6.1.4.1.311.21.20 (Client Information)
    Value[2][0]:
    Unknown Attribute type
    Client Id: = 1
    XECI_XENROLL -- 1
    User: CONTOSO0\avibm
    Machine: dcross-stress.contoso.com
    Process: certreq
```

Understanding the Unauthenticated Attributes Structure

The unauthenticated attributes section contains the encrypted private key. The private key is contained in an enveloped PKCS #7 message that is encrypted to the CA's exchange key. Since this is an unauthenticated attribute, the SHA1 hash of the PKCS #7 message is included as one of the attributes of the controlSequence TaggedAttribute attributes.

The following is an example of the unauthenticated attributes section.

```
Unauthenticated Attributes[0]:
  1 attributes:

  Attribute[0]: 1.3.6.1.4.1.311.21.13 (Encrypted Private Key)
    Value[0][0]:
    Unknown Attribute type
================ Begin Nesting Level 1 ================
PKCS7 Message:
  CMSG_ENVELOPED(3)
  CMSG_ENVELOPED_DATA_PKCS_1_5_VERSION(0)
  Content Type: 1.2.840.113549.1.7.1 PKCS 7 Data

PKCS7 Message Content:
0000    d4 a6 31 b6 5a ee 62 90  cc 17 b1 7a 6a 0d 40 9a
..1.Z.b....zj.@.
0010    33 fd 11 14 0b ae 12 bd  3b 32 b8 73 af cc 1b 76
```

```
3.......;2.s...v ...
```

Performing Binary Export for a Request

To view and decode a CMS key archival request from a Windows Server 2003 CA, it is necessary to do a binary export directly from the CA database. A binary export can be easily achieved through the Certificate Authority MMC snap-in or by using the certutil.exe command-line tool.

Binary Request Export Using the Certification Authority MMC Snap-In Walkthrough

To export a binary request using the Certification Authority MMC Snap-in

1. Log on to the CA machine using a CA Administrator account.
2. Open the **Certification Authority** MMC snap-in.
3. Click the **Issued Certificates** folder.
4. If the binary request column has not been previously added to the database view, it must be added to support a binary request export. To add a column to the view, click **View** on the menu bar, and then select the **Add/Remove Columns** menu option.
5. In the **Add/Remove Columns** dialog box, select the **Binary Request** field in the **Available Columns** list box on the left.
6. Click **Add**, and then click **OK**.

Next, a binary request can be exported.

1. Select a request from the issued certificates view, and then click the **Action** menu.
2. Select **Export Binary Data** on the **All Tasks** menu.
3. In the **Export Binary Data** dialog box, choose **Binary Request** as the column you want to export.
4. Click **OK**.

The data will be exported into ASCII format that can be opened in Notepad using notepad.exe.

Note: Following the previous steps will generate a dump of the certificate archival request only; it does not include the private key material. To dump a full certificate archival request including the private key material, follow the command-line option.

Binary Request Export Using the CertUtil.exe Command-Line Tool Walkthrough

To use the certutil.exe to view the certificate request including private key material, a request file has to be generated first.

To generate a request file

1.Run Notepad.exe.
2.Paste the following certificate request information into Notepad.

```
[Version]
Signature= "$Windows NT$"

[NewRequest]
Subject = "CN=Test Subject"
KeySpec = 1
Exportable = FALSE
PrivateKeyArchive = TRUE

[RequestAttributes]
CertificateTemplate = EFS
```

Note: Make sure that the CA is configured for key archival before starting this process. In this example, the EFS template is used; this should be changed to an existing certificate template that allows private key archival.

3.Save the file as **CertificateRequest.inf**, and then close **Notepad**.
4.Open the command-line window.
5.Type the following command.

Certreq –new CertificateRequest.inf CertificateRequest.req

Notes:

- This command will prompt you to select the CA to fetch the CA exchange certificate from, and to encrypt the private key to.
- This command will write the request to a file named by the last argument on the command line: CertificateRequest.req.
- To avoid using the CA selection dialog, you can specify the CA via -config **CAMachineDNSName \CACertCommonName** before or after the –new option.

6.Type the following command.

certreq -submit CertificateRequest.req KeyArchival.cer KeyArchival. p7b KeyArchival.rsp

This command will prompt you to select the CA to submit the request to.

Notes:

- This command will write the newly issued certificate, a PKCS7 containing only the issued certificate and chain, and the full CMC response to files named by the last three arguments on the command line: KeyArchival.cer, KeyArchival.p7b, and KeyArchival.rsp, respectively.
- To avoid the U/I, you can specify the CA via -config **CAMachineDNSName\CACertCommonName** before or after the **–submit**.

7.Type the following command.

certreq -accept KeyArchival.rsp

This command verifies the response, installs the certificate, and associates it with the private key.

8.Type the following command.

Certutil –privatekey –dump CertificateRequest.req >CertificateRequest.txt

This command will generate a dump of the certificate archival request into the CertificateRequest.txt file.

9.Type the following command.

Certutil –privatekey –dump KeyArchival.rsp >CertificateResponse.txt

This command will generate a dump of the certificate archival response into the CertificateResponse.txt file.

For non-Windows Server 2003 clients or servers enrolling to a Windows Server 2003 CA, the format of the request may be different. The reason is that non-Windows Server 2003 platforms may not support CMC data structures and, therefore, may not be able to encode the request information inside a PKIData object. Instead, the request information may be inside the Data body but not encoded as a PKIData object.

Note: certreq.exe and other tools may be installed on a Windows Server 2003 Professional machine by installing the Administrative Tools (adminpak.msi) that are located in the \i386 directory on all Windows Server 2003 CD-ROM media.

CMC Request and Response Examples

Request:

```
SEQUENCE :
 OBJECT IDENTIFIER :  signedData [1.2.840.113549.1.7.2]
 CONTEXT SPECIFIC (0) :
    SEQUENCE :
       INTEGER : 3
       SET :
          SEQUENCE :
             OBJECT IDENTIFIER :  sha1 [1.3.14.3.2.26]
             NULL :
       SEQUENCE :
          OBJECT IDENTIFIER :  [1.3.6.1.5.5.7.12.2]
          CONTEXT SPECIFIC (0) :
             OCTET STRING :
                SEQUENCE :
                   SEQUENCE :
                      SEQUENCE :
                         INTEGER : 2
                         OBJECT IDENTIFIER :  [1.3.6.1.5.5.7.7.8]
                            SET :
                               SEQUENCE :
                                  INTEGER : 0
                               SEQUENCE :
                                  INTEGER : 1
                               SEQUENCE :
                                  SEQUENCE :
                                     OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.21.10]
                                     OCTET STRING :
                                        SEQUENCE :
                                           SEQUENCE :
                                              OBJECT
IDENTIFIER :  encryptedFileSystem [1.3.6.1.4.1.311.10.3.4]
                                     SEQUENCE :
```

```
                                    OBJECT IDENTIFIER :  keyUsage [2.5.29.15]
                                    OCTET STRING :
                                    BIT STRING UnusedBits:5 :
                                          20
                                 SEQUENCE :
                                    OBJECT IDENTIFIER :  extKeyUsage [2.5.29.37]
                                       OCTET STRING :
                                          SEQUENCE :
                                             OBJECT
IDENTIFIER :  encryptedFileSystem [1.3.6.1.4.1.311.10.3.4]
                                    SEQUENCE :
                                       OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.21.7]
                                       OCTET STRING :
                                          SEQUENCE :
                                             OBJECT IDENTIFIER :
 [1.3.6.1.4.1.311.21.8.4014942.3497959.5914804.3829722.12246394.103.3066650.1537810]
                                             INTEGER : 100
                                             INTEGER : 2
                              SEQUENCE :
                                 INTEGER : 3
                                 OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.10.10.1]
                                 SET :
                                    SEQUENCE :
                                       INTEGER : 0
                                       SEQUENCE :
                                          INTEGER : 1
                                       SET :
                                          SEQUENCE :
                                             OBJECT
IDENTIFIER :  [1.3.6.1.4.1.311.21.21]
                                             SET :
                                                OCTET STRING :
                                                   9231E6C0B87445190EA2CA934B2807FF799
                                                   3C59F
                              SEQUENCE :
                                 INTEGER : 4
                                 OBJECT IDENTIFIER :  [1.3.6.1.5.5.7.7.18]
```

```
                                    SET :
                                       OCTET STRING :
                                          436572746966696361746554656D706C6174653D4172636
                                          869766554657374426173696345465326
                                 SEQUENCE :
                                    CONTEXT SPECIFIC (0) :
                                       INTEGER : 1
                                       SEQUENCE :
                                          SEQUENCE :
                                             INTEGER : 0
                                             SEQUENCE :
                                                SET :
                                                   SEQUENCE :
                                                      OBJECT IDENTIFIER :  commonName
[2.5.4.3]
                                                      PRINTABLE STRING :
'Test Subject'
                                             SEQUENCE :
                                                SEQUENCE :
                                                   OBJECT IDENTIFIER :  rsaEncryption
[1.2.840.113549.1.1.1]
                                                   NULL :
                                                BIT STRING UnusedBits:0 :
                                                   SEQUENCE :
                                                      INTEGER :
                                                         00DAFF7C6859557C698CDA4598222E8E90E
                                                         EB481889531E9F67F10C081F2545B060BE7
                                                         714E755325AC710774764DCA8120C6BEB7B
                                                         6EF74B0260EDD56DD299B242A94EE83C420
                                                         AC7FF0E694122E26EF67670782223C4E8D8
                                                         12C98047F24E10CF6A26FEBEEB826638924
                                                         F36B697CEA02EFC4CA0D108CB85047266AD
                                                         27DE582D181A1
                                                      INTEGER : 65537
                                             CONTEXT SPECIFIC (0) :
                                                SEQUENCE :
                                                   OBJECT
```

```
                IDENTIFIER :  [1.3.6.1.4.1.311.13.2.3]
                                        SET :
                                           IA5 STRING :
                                              '5.2.3790.2'
                                     SEQUENCE :
                                        OBJECT
                IDENTIFIER :  [1.3.6.1.4.1.311.21.20]
                                        SET :
                                           SEQUENCE :
                                              INTEGER : 1
                                              UTF8 STRING :
                                                 'dcross-stress.contoso.com'
                                              UTF8 STRING :
                                                 'CONTOSO0\avibm'
                                              UTF8 STRING :
                                                 'certreq'
                                     SEQUENCE :
                                        OBJECT
                IDENTIFIER :  [1.3.6.1.4.1.311.13.2.2]
                                        SET :
                                           SEQUENCE :
                                              INTEGER : 1
                                              BMP STRING :
                                                 'Microsoft Strong Cryptographic P'
                                                 'rovider'
                                              BIT STRING UnusedBits:0 :
                                                 00000000000000000000000000000000
                                                 00000000000000000000000000000000
                                                 00000000000000000000000000000000
                                                 00000000000000000000000000000000
                                                 00000000000000000000000000000000
                                                 00000000000000000000000000000000
                                                 00000000000000000000000000000000
                                                 00000000000000000000000000000000
                                                 0000000000000000
                                     SEQUENCE :
                                        OBJECT IDENTIFIER :  extensionReq
```

```
[1.2.840.113549.1.9.14]
                                        SET :
                                           SEQUENCE :
                                              SEQUENCE :
                                                 OBJECT
IDENTIFIER :  sMIMECapabilities [1.2.840.113549.1.9.15]
                                                 OCTET STRING :
                                                    SEQUENCE :
                                                       SEQUENCE :
                                                          OBJECT
IDENTIFIER :  rc2CBC [1.2.840.113549.3.2]
                                                          INTEGER : 128
                                                       SEQUENCE :
                                                          OBJECT IDENTIFIER :  rc4
[1.2.840.113549.3.4]
                                                          INTEGER : 128
                                                       SEQUENCE :
                                                          OBJECT
IDENTIFIER :  desCBC [1.3.14.3.2.7]
                                                       SEQUENCE :
                                                          OBJECT IDENTIFIER :  DES-
EDE3-CBC [1.2.840.113549.3.7]
                                              SEQUENCE :
                                                 OBJECT
IDENTIFIER :  subjectKeyIdentifier [2.5.29.14]
                                                 OCTET STRING :
                                                    OCTET STRING :
                                                       8192563AC431F8820C54C9D098
                                                       4FD8C534639ECC
                                              SEQUENCE :
                                                 OBJECT
IDENTIFIER :  [1.3.6.1.4.1.311.21.10]
                                                 OCTET STRING :
                                                    SEQUENCE :
                                                       SEQUENCE :
                                                          OBJECT
IDENTIFIER :  encryptedFileSystem [1.3.6.1.4.1.311.10.3.4]
```

```
                                                SEQUENCE :
                                                   OBJECT IDENTIFIER :  keyUsage
[2.5.29.15]
                                                   OCTET STRING :
                                                      BIT STRING UnusedBits:5 :
                                                         20
                                                SEQUENCE :
                                                   OBJECT IDENTIFIER :  extKeyUsage
[2.5.29.37]
                                                   OCTET STRING :
                                                      SEQUENCE :
                                                         OBJECT
IDENTIFIER :  encryptedFileSystem [1.3.6.1.4.1.311.10.3.4]
                                                SEQUENCE :
                                                   OBJECT
IDENTIFIER :  [1.3.6.1.4.1.311.21.7]
                                                   OCTET STRING :
                                                      SEQUENCE :
                                                         OBJECT IDENTIFIER :
 [1.3.6.1.4.1.311.21.8.4014942.3497959.5914804.3829722.12246394.103.3066650.1537810]
                                                         INTEGER : 100
                                                         INTEGER : 2
                                    SEQUENCE :
                                       OBJECT IDENTIFIER :  sha1withRSAEncryption
[1.2.840.113549.1.1.5]
                                       NULL :
                                    BIT STRING UnusedBits:0 :
                                       31E945A575155D8F91E972DB26A52C8FAE16D7F5074365D
                                       C2E585C8718AB09A4FBB67D8A78A63C76B14482A1DEDCAA
                                       5B234035F3CFFABCAF3DEC24C5944ACE46A1BAFE857F310
                                       7C21105C817FA88C0CCB23B88D2684327B40CB99E9A059F
                                       3B95BAC6423740CA1B46B4DC58664863325004DCA2857C2
                                       2B4117942CC7D39E86900
                              SEQUENCE :
                              SEQUENCE :
                     SET :
                        SEQUENCE :
```

```
INTEGER : 3
CONTEXT SPECIFIC (0) :
   8192563AC431F8820C54C9D0984FD8C534639ECC
SEQUENCE :
   OBJECT IDENTIFIER :  sha1 [1.3.14.3.2.26]
   NULL :
CONTEXT SPECIFIC (0) :
   SEQUENCE :
      OBJECT IDENTIFIER :  contentType [1.2.840.113549.1.9.3]
      SET :
         OBJECT IDENTIFIER :  [1.3.6.1.5.5.7.12.2]
   SEQUENCE :
      OBJECT IDENTIFIER :  messageDigest [1.2.840.113549.1.9.4]
      SET :
         OCTET STRING :
            5E1F0FF028A4FE910DC22F1A18787E2E107F1739
   SEQUENCE :
      OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.21.20]
      SET :
         SEQUENCE :
            INTEGER : 1
            UTF8 STRING :
               'dcross-stress.contoso.com'
            UTF8 STRING : 'CONTOSO0\avibm'
            UTF8 STRING : 'certreq'
SEQUENCE :
   OBJECT IDENTIFIER :  rsaEncryption [1.2.840.113549.1.1.1]
   NULL :
OCTET STRING :
   C1AE90A7A30B5266EAC4D004172E949514200653AA5EA3C2BF17C7731DA8EB
   1A635CE1DC4F5AD9FB44EF2D9E8C9F961800DBEBC1ADE14E0459A8B46880DF
   01A177FC9B02B89113638F3A6A3B3ED0765BD16B905D6BCB404F65E79AAB12
   97F2F9F52D68D13373D41D510D97A954800368F8DEDEE13D8635EBF4364512
   17407F1A
CONTEXT SPECIFIC (1) :
   SEQUENCE :
      OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.21.13]
```

```
                                SET :
                                   SEQUENCE :
                                      OBJECT IDENTIFIER :  envelopedData
[1.2.840.113549.1.7.3]
                                      CONTEXT SPECIFIC (0) :
                                         SEQUENCE :
                                            INTEGER : 0
                                            SET :
                                               SEQUENCE :
                                                  INTEGER : 0
                                                  SEQUENCE :
                                                     SEQUENCE :
                                                        SET :
                                                           SEQUENCE :
                                                              OBJECT
IDENTIFIER :  domainComponent [0.9.2342.19200300.100.1.25]
                                                              IA5 STRING :
                                                                 'com'
                                                        SET :
                                                           SEQUENCE :
                                                              OBJECT
IDENTIFIER :  domainComponent [0.9.2342.19200300.100.1.25]
                                                              IA5 STRING :
                                                                 'contoso'
                                                        SET :
                                                           SEQUENCE :
                                                              OBJECT IDENTIFIER :  commonName
[2.5.4.3]
                                                              PRINTABLE STRING :
                                                                 'TestEnrollment'
                                                     INTEGER :
                                                        18D0100D00000000005B
                                                  SEQUENCE :
                                                     OBJECT IDENTIFIER :  rsaEncryption
[1.2.840.113549.1.1.1]
                                                     NULL :
                                                  OCTET STRING :
```

```
                                        A41AAE9CDA66F283D6D4BC829D2F58BCECFD3F
                                        5A57EC8AE14021179AE5F93F03AE90747FD300
                                        4573ED78F802E02AB3C6ADEDEAA367069DA399
                                        8E1D2D34ABEEFF0F8DE2CB76078C56D883BD94
                                        D7CE9C5CD75F5E3F442A467F74E07C5A434E4A
                                        F1BDD6EC493F3A870764B6CC6446FA5D674255
                                        D93F248DE23E0D96902C79901800
                                  SEQUENCE :
                                     OBJECT IDENTIFIER :  data
[1.2.840.113549.1.7.1]
                                     SEQUENCE :
                                        OBJECT IDENTIFIER :  DES-EDE3-CBC
[1.2.840.113549.3.7]
                                        OCTET STRING :
                                           06003B8D3EB4B44C
                                     CONTEXT SPECIFIC (0) :
                                        D4A631B65AEE6290CC17B17A6A0D409A33FD11140
                                        BAE12BD3B32B873AFCC1B76A4022D0FB2B50E431A
                                        1E48C8D45865EC5B730D7357D61C9495235143381
                                        19CDBF34C5455B73C9FF38AEFBC4E32DD8145647B
                                        46B0B4A60D29D062051F116C6BA49253D4590944A
                                        7CCB70F43E7E850B34DE55074B3C5FF5AE1C5A18C
                                        6BC271D1F2BC3FBBE19558252C894110CC801292D
                                        63DA1485BDB957270E6C1A38FE33D672EA3E8D031
                                        CD7BCFEF5C738818DCC43A6F76F3EC81701C561DF
                                        9FA6032C47236D9A16973BDF6A033F4925CC5B491
                                        C00C635C65F744C8FEBE19B1EDD2172AE3A7CFB70
                                        87A6BCAE7BB52BCEEF412889C4A45ACAE0ACC0E43
                                        A14C7AA34FB4B4C49360ADD0C65D1494B792E04D7
                                        8D43C2EDB79974B5C08C87E0C72767C26A2EBF6F0
                                        E273269D139F2D6F451301944B76218D9BD4C5931
                                        50C79FA5DA1AF1383E5342EC2F5318E2404774345
                                        B82A0CB4EE26FC0D59A1D18EDBBEFF6135675D014
                                        293470B301CC59387C4E627E1F6B038A158A927B9
                                        160387104BFC5466B7FB4107DF02D136E076F2CAD
                                        94718ADD9F93C0D376A80A6E3796C6236888E6517
                                        1D36A0F3BFAA8B8E44FC8DA426F3F19128A910D83
```

```
71A7D68CDFEFCA0BBF32888D8AC679975AE43BB6D
209D61F82EEA2463616E905177E929CFD3D85C8ED
8ED1EDCECA01CA1580960E87D57591817C863FE33
757F527DC7C6457ED5CEDC3BE1597A05BFB10A145
522C98AF266A992CC607434D3421D57A80195D052
557AE89652193B840FC27CB343C2C242445453E78
9E6E397DFC84363B4EAE801DF1BE2993D1AF13256
A1390C4B7D51127CC55FF0B1184D4E87967961E86
B722E1048C0
```

Response:

```
SEQUENCE :
    OBJECT IDENTIFIER :  signedData [1.2.840.113549.1.7.2]
    CONTEXT SPECIFIC (0) :
       SEQUENCE :
          INTEGER : 3
          SET :
             SEQUENCE :
                OBJECT IDENTIFIER :  sha1 [1.3.14.3.2.26]
                NULL :
          SEQUENCE :
             OBJECT IDENTIFIER :  [1.3.6.1.5.5.7.12.3]
             CONTEXT SPECIFIC (0) :
                OCTET STRING :
                   SEQUENCE :
                      SEQUENCE :
                         SEQUENCE :
                            INTEGER : 1
                            OBJECT IDENTIFIER :  [1.3.6.1.5.5.7.7.1]
                            SET :
                               SEQUENCE :
                                  INTEGER : 0
                                  SEQUENCE :
                                     INTEGER : 1
                                  UTF8 STRING : 'Issued'
                         SEQUENCE :
```

```
                                    INTEGER : 2
                                    OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.10.10.1]
                                    SET :
                                       SEQUENCE :
                                          INTEGER : 0
                                          SEQUENCE :
                                             INTEGER : 1
                                          SET :
                                             SEQUENCE :
                                                OBJECT
IDENTIFIER :  [1.3.6.1.4.1.311.21.17]
                                                SET :
                                                   OCTET STRING :
                                                      DE73D68A50323310A01EEDDF66188213DC9
                                                      CD490
                                             SEQUENCE :
                                                OBJECT
IDENTIFIER :  [1.3.6.1.4.1.311.21.21]
                                                SET :
                                                   OCTET STRING :
                                                      9231E6C0B87445190EA2CA934B2807FF799
                                                      3C59F
                              SEQUENCE :
                              SEQUENCE :
                     CONTEXT SPECIFIC (0) :
                        SEQUENCE :
                           SEQUENCE :
                              CONTEXT SPECIFIC (0) :
                                 INTEGER : 2
                              INTEGER :
                                 172B1FB96BBBF2BA49A64EBEA41833EF
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  sha1withRSAEncryption
[1.2.840.113549.1.1.5]
                                 NULL :
                              SEQUENCE :
                                 SET :
```

```
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  domainComponent
[0.9.2342.19200300.100.1.25]
                                 IA5 STRING : 'com'
                           SET :
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  domainComponent
[0.9.2342.19200300.100.1.25]
                                 IA5 STRING : 'contoso'
                           SET :
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  commonName [2.5.4.3]
                                 PRINTABLE STRING :
                                    'TestEnrollment'
                        SEQUENCE :
                           UTC TIME : '040210162354Z'
                           UTC TIME : '090210162738Z'
                        SEQUENCE :
                           SET :
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  domainComponent
[0.9.2342.19200300.100.1.25]
                                 IA5 STRING : 'com'
                           SET :
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  domainComponent
[0.9.2342.19200300.100.1.25]
                                 IA5 STRING : 'contoso'
                           SET :
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  commonName [2.5.4.3]
                                 PRINTABLE STRING :
                                    'TestEnrollment'
                        SEQUENCE :
                           SEQUENCE :
                              OBJECT IDENTIFIER :  rsaEncryption [1.2.840.113549.1.1.1]
                              NULL :
```

```
            BIT STRING UnusedBits:0 :
               SEQUENCE :
                  INTEGER :
                     00E23136361B94412ABD67C376C6AC882B50F45D9AD28719C1
                     5B0F3125CB352E19F5A381A33FF2971CC4702747BD94C3EE93
                     75493C1A48F5174BE1F8135CCFB641F3EE6042C4771E8E176A
                     7B65E49E407903072C28E2CC92153454664630FDA3CC70A805
                     086B586592AF45BFFE5CC82DCF1ED622DD9BE4ECF64D635600
                     9338C96F7D2EF77447F3ACD2AFC9C76EBC7A77DAAA9245A0EE
                     0398D041B37DD78BD77C46D84A808AECDB88EC4319B1E6ADB9
                     19053A84D3403163003EE696F65E0A55F5EA7A4955870D451E
                     E4A0AB684EE6ED503437A3F4388DC96A00A9F7D26E3527B3D0
                     F657EFB8E431B24A97ADBD1475DAF545B9754856200E640E42
                     CA8BF78614A953
                  INTEGER : 65537
         CONTEXT SPECIFIC (3) :
            SEQUENCE :
               SEQUENCE :
                  OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.20.2]
                  OCTET STRING :
                     BMP STRING : 'CA'
               SEQUENCE :
                  OBJECT IDENTIFIER :  keyUsage [2.5.29.15]
                  OCTET STRING :
                     BIT STRING UnusedBits:1 :
                        86
               SEQUENCE :
                  OBJECT IDENTIFIER :  basicConstraints [2.5.29.19]
                  BOOLEAN : 'FF'
                  OCTET STRING :
                     SEQUENCE :
                        BOOLEAN : 'FF'
               SEQUENCE :
                  OBJECT IDENTIFIER :  subjectKeyIdentifier [2.5.29.14]
                  OCTET STRING :
                     OCTET STRING :
                        10C8E49879236E65350924C24EFB074EFB5F4AA0
```

```
                                    SEQUENCE :
                                       OBJECT IDENTIFIER :  cRLDistributionPoints [2.5.29.31]
                                       OCTET STRING :
                                          SEQUENCE :
                                             SEQUENCE :
                                                CONTEXT SPECIFIC (0) :
                                                   CONTEXT SPECIFIC (0) :
                                                      CONTEXT SPECIFIC (6) :
                                                         'ldap:///CN=TestEnrollment,CN=dcross'
                                                         '-stress,CN=CDP,CN=Public%20Key%20Se'
                                                         'rvices,CN=Services,CN=Configuration'
                                                         ',DC=contoso,DC=com?certificateRevoc'
                                                         'ationList?base?objectClass=cRLDistr'
                                                         'ibutionPoint'
                                                      CONTEXT SPECIFIC (6) :
                                                         'http://dcross-stress.contoso.com/Ce'
                                                         'rtEnroll/TestEnrollment.crl'
                                    SEQUENCE :
                                       OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.21.1]
                                       OCTET STRING :
                                          INTEGER : 0
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  sha1withRSAEncryption
                        [1.2.840.113549.1.1.5]
                                 NULL :
                                 BIT STRING UnusedBits:0 :
                                    CA9E6760A8DFB0D213E90D7450B5C7A7C5C920760D01EB45E4F46A23780841
                                    40EDE1A37BA123934C06A39F9638F86C9A50258E43E71DE44239A20DFD6EAE
                                    C636F6B50C964EF23A72B349F35530A96CC99AF8937F22F684AF5E39E64C90
                                    F49C0D87621BBB13DE9FAF84609C26C5ECEB37F479CAEF826D36C19FD5C80D
                                    B865D0C6FF287DE8FF0CD3FE0476E514ED82D9A23DCB684D28E3B93A229A7B
                                    D4DAF89E9A2F2D62599B91E8746830BCF88947611A82E9893137ABBA74B489
                                    6C9C1492DCA2A7FA75F46451C7838EC0E9FB5D9222D3895C116C2C13E3995F
                                    6D56ACB5F62FD7B764FAAB5AF0B5EA73AF3211B40AE44697DCB6E0D28E88E9
                                    00037A506832C0BA
                           SEQUENCE :
                              SEQUENCE :
```

```
                        CONTEXT SPECIFIC (0) :
                           INTEGER : 2
                        INTEGER : '18E922D0000000000060'
                        SEQUENCE :
                           OBJECT IDENTIFIER :  sha1withRSAEncryption
[1.2.840.113549.1.1.5]
                           NULL :
                        SEQUENCE :
                           SET :
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  domainComponent
[0.9.2342.19200300.100.1.25]
                                 IA5 STRING : 'com'
                           SET :
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  domainComponent
[0.9.2342.19200300.100.1.25]
                                 IA5 STRING : 'contoso'
                           SET :
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  commonName [2.5.4.3]
                                 PRINTABLE STRING :
                                    'TestEnrollment'
                        SEQUENCE :
                           UTC TIME : '040812185455Z'
                           UTC TIME : '050812185455Z'
                        SEQUENCE :
                           SET :
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  domainComponent
[0.9.2342.19200300.100.1.25]
                                 IA5 STRING : 'com'
                           SET :
                              SEQUENCE :
                                   OBJECT IDENTIFIER :  domainComponent
[0.9.2342.19200300.100.1.25]
                                    IA5 STRING : 'contoso'
```

```
                                    SET :
                                       SEQUENCE :
                                          OBJECT IDENTIFIER :  commonName [2.5.4.3]
                                          PRINTABLE STRING : 'Users'
                                    SET :
                                       SEQUENCE :
                                          OBJECT IDENTIFIER :  commonName [2.5.4.3]
                                          PRINTABLE STRING :
                                             'Avi Ben-Menahem'
                                 SEQUENCE :
                                    SEQUENCE :
                                       OBJECT IDENTIFIER :  rsaEncryption
[1.2.840.113549.1.1.1]
                                       NULL :
                                    BIT STRING UnusedBits:0 :
                                       SEQUENCE :
                                          INTEGER :
                                             00DAFF7C6859557C698CDA4598222E8E90EEB481889531E9F6
                                             7F10C081F2545B060BE7714E755325AC710774764DCA8120C6
                                             BEB7B6EF74B0260EDD56DD299B242A94EE83C420AC7FF0E694
                                             122E26EF67670782223C4E8D812C98047F24E10CF6A26FEBEE
                                             B826638924F36B697CEA02EFC4CA0D108CB85047266AD27DE5
                                             82D181A1
                                          INTEGER : 65537
                                 CONTEXT SPECIFIC (3) :
                                    SEQUENCE :
                                       SEQUENCE :
                                          OBJECT IDENTIFIER :  sMIMECapabilities
[1.2.840.113549.1.9.15]
                                          OCTET STRING :
                                             SEQUENCE :
                                                SEQUENCE :
                                                   OBJECT IDENTIFIER :  rc2CBC
[1.2.840.113549.3.2]
                                                   INTEGER : 128
                                                SEQUENCE :
                                                   OBJECT IDENTIFIER :  rc4
```

```
                [1.2.840.113549.3.4]
                                        INTEGER : 128
                                    SEQUENCE :
                                        OBJECT IDENTIFIER :  desCBC [1.3.14.3.2.7]
                                    SEQUENCE :
                                        OBJECT IDENTIFIER :  DES-EDE3-CBC
                [1.2.840.113549.3.7]
                                SEQUENCE :
                                    OBJECT IDENTIFIER :  subjectKeyIdentifier [2.5.29.14]
                                    OCTET STRING :
                                        OCTET STRING :
                                            8192563AC431F8820C54C9D0984FD8C534639ECC
                                SEQUENCE :
                                    OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.21.10]
                                    OCTET STRING :
                                        SEQUENCE :
                                            SEQUENCE :
                                                OBJECT IDENTIFIER :  encryptedFileSystem
                [1.3.6.1.4.1.311.10.3.4]
                                SEQUENCE :
                                    OBJECT IDENTIFIER :  keyUsage [2.5.29.15]
                                    OCTET STRING :
                                        BIT STRING UnusedBits:5 :
                                            20
                                SEQUENCE :
                                    OBJECT IDENTIFIER :  extKeyUsage [2.5.29.37]
                                    OCTET STRING :
                                        SEQUENCE :
                                            OBJECT IDENTIFIER :  encryptedFileSystem
                [1.3.6.1.4.1.311.10.3.4]
                                SEQUENCE :
                                    OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.21.7]
                                    OCTET STRING :
                                        SEQUENCE :
                                            OBJECT IDENTIFIER :

                [1.3.6.1.4.1.311.21.8.4014942.3497959.5914804.3829722.12246394.103.3066650.1537810]
```

```
                  INTEGER : 100
                  INTEGER : 2
            SEQUENCE :
               OBJECT IDENTIFIER :  authorityKeyIdentifier [2.5.29.35]
               OCTET STRING :
                  SEQUENCE :
                     CONTEXT SPECIFIC (0) :
                        10C8E49879236E65350924C24EFB074EFB5F4AA0
            SEQUENCE :
               OBJECT IDENTIFIER :  cRLDistributionPoints [2.5.29.31]
               OCTET STRING :
                  SEQUENCE :
                     SEQUENCE :
                        CONTEXT SPECIFIC (0) :
                           CONTEXT SPECIFIC (0) :
                              CONTEXT SPECIFIC (6) :
                                 'ldap:///CN=TestEnrollment,CN=dcross'
                                 '-stress,CN=CDP,CN=Public%20Key%20Se'
                                 'rvices,CN=Services,CN=Configuration'
                                 ',DC=contoso,DC=com?certificateRevoc'
                                 'ationList?base?objectClass=cRLDistr'
                                 'ibutionPoint'
                              CONTEXT SPECIFIC (6) :
                                 'http://dcross-stress.contoso.com/Ce'
                                 'rtEnroll/TestEnrollment.crl'
            SEQUENCE :
               OBJECT IDENTIFIER :  authorityInfoAccess
[1.3.6.1.5.5.7.1.1]
               OCTET STRING :
                  SEQUENCE :
                     SEQUENCE :
                        OBJECT IDENTIFIER :  caIssuers
[1.3.6.1.5.5.7.48.2]
                        CONTEXT SPECIFIC (6) :
                           'ldap:///CN=TestEnrollment,CN=AIA,CN=Publi'
                           'c%20Key%20Services,CN=Services,CN=Configu'
                           'ration,DC=contoso,DC=com?cACertificate?ba'
```

```
                                        'se?objectClass=certificationAuthority'
                                    SEQUENCE :
                                       OBJECT IDENTIFIER :  caIssuers
[1.3.6.1.5.5.7.48.2]
                                       CONTEXT SPECIFIC (6) :
                                          'http://dcross-stress.contoso.com/CertEnro'
                                          'll/dcross-stress.contoso.com_TestEnrollme'
                                          'nt.crt'
                                SEQUENCE :
                                   OBJECT IDENTIFIER :  subjectAltName [2.5.29.17]
                                   OCTET STRING :
                                      SEQUENCE :
                                         CONTEXT SPECIFIC (0) :
                                            OBJECT IDENTIFIER :  [1.3.6.1.4.1.311.20.2.3]
                                            CONTEXT SPECIFIC (0) :
                                               UTF8 STRING :
                                                  'avibm@contoso.com'
                        SEQUENCE :
                           OBJECT IDENTIFIER :  sha1withRSAEncryption
[1.2.840.113549.1.1.5]
                           NULL :
                        BIT STRING UnusedBits:0 :
                           9D0000D2CC5668BEE443EBDE5EE4CADA5D61C17C00B262A3F231726FD2E7A8
                           500603B89BE123D577FA2AE592567FB96743A6AE9B57AE089B1C205D6552F5
                           5D60DD825D94D27301527FDB275035473DFC16A4F0C4886036A50CA1D320E3
                           D284744CC0E552D1FFB24CD6110E6B17C86F830B5CC7A7E1791930320373CA
                           C4E667BC372983597713CF8608389A6C82F9079FF8666C867BF2243DE5A22C
                           20DBDBAD788A77758B68D9260EA5040A2F5C97C1AD80144F06F714D20BF671
                           96BE5774D16080A9EAA5933C3C7EA34AE3F41DC001E0C2F83EA7AFAADA4812
                           D0F27C48E288A20C44F085F328CCE6F478D6E4E89131D8EF43DA7B23DA39C9
                           8CB15DE2EBA2BC8F
                     SET :
                        SEQUENCE :
                           INTEGER : 1
                           SEQUENCE :
                              SEQUENCE :
                                 SET :
```

```
                                    SEQUENCE :
                                       OBJECT IDENTIFIER :  domainComponent
                [0.9.2342.19200300.100.1.25]
                                       IA5 STRING : 'com'
                                 SET :
                                    SEQUENCE :
                                       OBJECT IDENTIFIER :  domainComponent
                [0.9.2342.19200300.100.1.25]
                                       IA5 STRING : 'contoso'
                                 SET :
                                    SEQUENCE :
                                       OBJECT IDENTIFIER :  commonName [2.5.4.3]
                                       PRINTABLE STRING :
                                          'TestEnrollment'
                              INTEGER :
                                 172B1FB96BBBF2BA49A64EBEA41833EF
                           SEQUENCE :
                              OBJECT IDENTIFIER :  sha1 [1.3.14.3.2.26]
                              NULL :
                           CONTEXT SPECIFIC (0) :
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  contentType [1.2.840.113549.1.9.3]
                                 SET :
                                    OBJECT IDENTIFIER :  [1.3.6.1.5.5.7.12.3]
                              SEQUENCE :
                                 OBJECT IDENTIFIER :  messageDigest [1.2.840.113549.1.9.4]
                                 SET :
                                    OCTET STRING :
                                       17CEEAA968CDD0A92DFC7E9AA174F87755AD8A87
                           SEQUENCE :
                              OBJECT IDENTIFIER :  rsaEncryption [1.2.840.113549.1.1.1]
                              NULL :
                           OCTET STRING :
                              C5D3AC35D5AAE5766640A2EF87D8ED005BB9BD63D51B10D803EEFEA1261161
                              3031241F695A2EDFF0240EE624D22FECB5AB6B74FD97A5DB12B3B873558AD5
                              0BC6DB59E438A7150A27749F53CBA447CD0751D7D49EEE3EBD1BBB20234887
                              5DD11DE26764DDEB2EBFA1E0023DD8CECF9C2530E2D0886FF26EAB747635A7
```

```
A57B7CA154BD0083A1DA891A35C3CD7EF5BA735FBCD2FD811FABE68C988C4D
172572BE63AE0575CF646756D4E66B2B127A699119368AAFB8B54661D317AF
2DF2622A0FFF01F18D5EF261E830107BD7F58848813CA6C0F8BF681A214E37
13618340D6DE9594829FB2B2DB1CFF973DC01F22D982846E474DDB9767D1BF
51E8C66F934593B5
```

Recovery BLOB Structure

When stored in the CA database, the private key is stored as a PKCS #7 message, encrypted with a 3DES symmetric key that is encrypted to the KRA(s) public key as a column in the CA database. When the recovery BLOB is retrieved by the certutil –getkey command, the encrypted PKCS #7 and the KRA certificate hashes are retrieved from the database. Also, the encrypted PKCS #7 is wrapped inside a signed PKCS #7 to allow collecting the previous certificates and attaching them to the signed PKCS #7. The PKCS #7 is not protected with a password since it is already protected by the public key of the recovery agent(s). The outer PKCS #7 wrapper can contain the certificate chains for the recovery agent(s) and the end-entity to facilitate the recovery operations and construction of the end-entity PKCS #12 file. Figure 47 illustrates the recovery BLOB structure.

The recovery BLOB consists of wrapping the encrypted PKCS #7 in the database in another (signed) PKCS #7 to allow a number of certificates to be included in the recovery BLOB. The returned certificates include the full chain of the user certificate being recovered, the chain of the signing CA certificate (which may differ from the CA certificate under which the user certificate was issued), and the KRA certificates to which the key was encrypted. The szOID_ARCHIVED_KEY_CERT_HASH(1.3.6.1.4.1.311.21.16) is an attribute containing the SHA-1 hash of the certificate for the key being recovered, attached as an authenticated attribute to the CA signature of the recovery BLOB. This allows certutil -recoverkey *recoveryblobfile* to also display the Subject name of the KRA certificate(s) used to protect the private key BLOB.

Figure 47: Recovery BLOB

ASN.1 Structure

The following is the ASN.1 structure of the PKCS #7 EnvelopedData object.

```
EnvelopedData ::= SEQUENCE {

version                    Version,

recipientInfos             RecipientInfos,

encryptedContentInfo       EncryptedContentInfo

}
```

Storing the recovery BLOB as an enveloped PKCS #7 enables a recovery agent to retrieve the recovery BLOB from the CA database. The recovery agent's private key is used to decrypt the EncryptedContentInfo to extract the PKCS #12 data. The following is the ASN.1 structure of the EncryptedContentInfo body.

```
EncryptedContentInfo ::= SEQUENCE {

contentType                ContentType,

contentEncryptionAlgorithm    ContentEncryptionAlgorithmIdentifier,

encryptedContent[0]        IMPLICIT EncryptedContent OPTIONAL

}
```

By definition, there can be multiple recovery agent certificates specified by RecipientInfo, where IssuerAndSerialNumber is used to disambiguate between multiple recovery agent certificates. Only the recovery agent certificates included in the RecipientInfo body of the enveloped PKCS #7 object can be used to recover the archived key material. The following is the ASN.1 structure of the RecipientInfo body.

```
RecipientInfo ::= SEQUENCE {

version                    Version,

issuerAndSerialNumber    IssuerAndSerialNumber,

keyEncryptionAlgorithm    KeyEncryptionAlgorithmIdentifier,

encryptedKey                EncryptedKey

}
```

Top of page

Appendix B: Additional Information

General

Windows Server 2003 documentation

http://www.microsoft.com/windowsserver2003/

White Papers

Implementing and Administering Certificate Templates in Windows Server 2003

http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/security/ws03crtm.mspx

Certificate Autoenrollment in Windows XP

http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/certenrl.mspx

Windows Server 2003 PKI Operations Guide

http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/security/ws03pkog.mspx

What's New in Windows XP and Windows Server 2003 PKI

http://www.microsoft.com/technet/prodtechnol/winxppro/plan/pkienh.mspx

Encrypting File System in Windows XP and Windows Server 2003

http://www.microsoft.com/technet/prodtechnol/winxppro/deploy/cryptfs.mspx

Data Recovery

http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/windows/xp/all/reskit/en-us/prnb_efs_raon.asp

Qualified Subordination

http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/security/ws03qswp.mspx

Standards

Internet X.509 Public Key Infrastructure – Certificate and Certificate Revocation List (CRL) Profile

http://www.ietf.org/rfc/rfc3280.txt?number=3280

Certificate Management Messages over CMS (CMC)

http://www.ietf.org/rfc/rfc2797.txt

PKCS #12 - Personal Information Exchange Syntax Standard

http://www.rsasecurity.com/rsalabs/node.asp?id=2138

Common Criteria

http://www.commoncriteria.org

CIMC Protection Profile

http://www.niap.nist.gov/cc-scheme/pp/PP_CIMCPP_SL3_V1.0.html

Top of page

Appendix C: Useful Commands

The following commands are available with certutil.exe in the Windows Server 2003 Administration Tools pack in performing key archival and recovery functions.

- **certutil –ImportKMS** accepts a *.pfx, *.epf, or a KMS export file, and archives the contents in the CA database.
- **certutil –ConvertEPF** converts a *.pfx (PKCS #12) file to a *.epf-formatted file for import into Outlook.
  - *certutil –ConvertEPF* and *–MergePFX* are similar in that they both accept a comma-separated list of input files that are merged and placed in the output file. Input files for both commands may be *.pfx files, *.epf files, or a mixture of the two.
  - The –cast parameter should be used to specify the CAST encryption algorithm.
  - The –silent option may be specified to suppress the UI.
- **certutil –ImportPFX** accepts a *.pfx or a *.epf file, and installs the certificates and keys into the HKLM (local machine) MY store. If the –User parameter is specified, the key and certificate will be imported into the HKCU (user profile) MY store. The CSP to be used may also be overridden by specifying the name with the –csp parameter.
- **certutil –getkey** retrieves the archived private key recovery BLOB from the CA database.
- **certutil –recoverkey** recovers the archived private key.
- **certutil –verifykeys** verifies the private/public key pair.

Top of page

Certificate Templates Troubleshooting

Updated: January 21, 2005

Troubleshooting

What problem are you having?

The Certificate Templates Microsoft Management Console (MMC) does not list any templates after prompting to install new certificate templates.

Cause: The certificate templates have not yet replicated to the certification authority (CA) that the computer is connected to. This replication is part of Active Directory replication.

Solution: Wait for the certificate templates to replicate and then reopen the Certificate Templates MMC.

Certificates are not being issued to clients.

Cause: The certification authority (CA) issuing certificate has a shorter remaining lifetime than the template overlap period configured for the request certificate template. This means the issued certificate would be immediately eligible for reenrollment. Instead of issuing and

endlessly renewing this certificate the certificate request is not processed.

Solution: Renew the issuing certificate used by the CA.

Certificates are issued to subjects, but cryptographic operations with those certificates fail.

Cause: Cryptographic service provider does not match Key Usage settings.

Solution: Confirm that you set the cryptographic service provider in the template to one that supports the type of cryptographic operation that the certificate will be used for.

See also: Key type and cryptographic service provider type; Modify a Certificate Template

Domain controllers are not obtaining a domain controller certificate.

Cause: Autoenrollment is turned off by way of Group Policy on domain controllers. Domain controllers obtain their certificates through autoenrollment.

Solution: Enable autoenrollment for domain controllers.

See also: Modify a Certificate Template

Cause: The default Automatic Certificate Request setting for Domain Controllers has been removed from the Default Domain Controllers Policy.

Solution: Create a new Automatic Certificate Request in the Default Domain Controllers Policy for the Domain Controller certificate template.

See also: Automatic certificate request settings

Clients are unable to obtain certificates via autoenrollement.

Cause: Security permissions must be set to allow intended subjects to both enroll and autoenroll on the certificate template. Both permissions are required to enable autoenrollment.

Solution: Modify the access control list on the certificate template to grant **Read**, **Enroll** and **Autoenroll** permissions for the subjects that you want.

See also: Allow subjects to request a certificate that is based on the template

Names of certificate templates in the snap-in are inconsistent between views or windows.

Cause: Active Directory Sites and Services is being used to view the certificate templates. This tool may not provide as accurate a display as Certificate Templates.

Solution: Use the Certificate Templates snap-in to administer certificate templates.

See also: Modify a Certificate Template

The private key cannot be exported from smart card certificates, even when Allow private key to be exported is selected in the certificate template.

Cause: Smart cards do not allow private keys to be exported once they are written to the smart card.

Solution: None

See also: Smart Cards

The certificate template is modified, but some certification authorities (CAs) still have the unmodified version.

Cause: Certificate templates are replicated between CAs with the Active Directory replication process. Because this replication is not instantaneous, there may be a short delay before the new version of the template is available on all CAs.

Solution: Wait until the modified template is replicated to all CAs. To display the certificate templates that are available on the CA, use the Certutil.exe command.

See also: Understanding Sites and Replication and Certutil.

The private key is not being archived even though I selected the Archive subject's encryption private key option and configured the CA to require key recovery.

Cause: Private keys will not be archived when the key usage for the certificate template is set to **Signature**. This is because the digital signature usage requires the key to not be recoverable.

Solution: None

See also: Establishing key options and key archival and Key archival and recovery.

Autoenrollment is prompting me to renew a certificate that isn't mine, and I have certificates in my private certificate store that I didn't put there.

Cause: When using the smart card enrollment station on the administrator's computer to renew or change the certificate stored on the smart card, the certificate from the smart card is copied

to the administrator's private certificate store. This certificate may be processed by autoenrollment and prompt you to begin the renewal process.

Solution: Click **Start** to begin the autoenrollment renewal process. Because the certificate is not yours, the autoenrollment process will disappear at that point. If you want to remove the certificates from your private store, they can be deleted manually.

See also: Delete a certificate.

Top of page

Results 1 - 10 for: **renew certificate**

All Results

View results in another search category by clicking a link in the right column...

Show Me:

All Results

Downloads

Product Information

Support & Troubleshooting

Technical Resources

Training & Books

Partner & Business Resources

Communities & Newsgroups

Microsoft News & Corporate Information

Related Links

- Building an Enterprise Root Certification Authority in Small and Medium Businesses
- Microsoft Learning Home Page

Renew a certificate with the same key

To renew a certificate with the same key Open an MMC console that contains Certificates. In the console tree, under Personal, click Certificates. Where? Certificates - Certificate holder Personal Certificates In the details pane, click the certificate you
http://www.microsoft.com/windows2000/en/advanced/help/sag_CMprocsRenewCertSameKey.htm

Renew a certificate with a new key

To renew a certificate with a new key Open an MMC console that contains Certificates. In the console tree, under Personal, click Certificates. Where? Certificates - Certificate holder Personal Certificates In the details pane, click the certificate you
http://www.microsoft.com/windows2000/en/advanced/help/sag_CMprocsRenewCert.htm

How To Renew or Create New Certificate Signing Request While Another Certificate Is Currently Installed

This article describes how you can create a new certificate signing request (CSR) or generate a renewal request without having to remove the existing certificate from your Web site. To create a new CSR or generate a renewal request while another...
http://support.microsoft.com/default.aspx?scid=kb;en-us;295281

To Renew a Server Licensor Certificate

To Renew a Server Licensor Certificate
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/RMS/RMSOperationsTC/affce9cf-8b46-4293-8e1c-ee06f2ca6537.mspx

How To Renew VeriSign SSL Certificate with New Key in IIS 5.0 MMC

If you try to renew a Secure Sockets Layer (SSL) Server Certificate from VeriSign, and you want to generate new keys, the Internet Information Server (IIS) 5.0 Microsoft Management Console (MMC) tries to connect to your local Certificate Authority...
http://support.microsoft.com/default.aspx?scid=kb;en-us;295329

Microsoft Windows XP - Renew a certificate with a new key

To renew a certificate with a new key Open Certificates. In the console tree, under Personal, click Certificates. Where? Certificates - Certificate holder Personal Certificates In the details pane, click the certificate you are renewing.
http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_cmprocsrenewcert.mspx

Microsoft Windows XP - Renew a certificate with a new key

To renew a certificate with a new key Open Certificates. In the console tree, under Personal, click Certificates. Where? Certificates - Certificate holder Personal Certificates In the details pane, click the certificate you are renewing.
http://www.microsoft.com/technet/prodtechnol/winxppro/proddocs/sag_cmprocsrenewcert.mspx

Microsoft Windows XP - Renew a certificate with the same key

To renew a certificate with the same key Open Certificates. In the console tree, under Personal, click Certificates. Where? Certificates - Certificate holder Personal Certificates In the details pane, click the certificate you are renewing.
http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_cmprocsrenewcertsamekey.mspx

Microsoft Windows XP - Renew a certificate with the same key

To renew a certificate with the same key Open Certificates. In the console tree, under Personal, click Certificates. Where? Certificates - Certificate holder Personal Certificates In the details pane, click the certificate you are renewing.
http://www.microsoft.com/technet/prodtechnol/winxppro/proddocs/sag_cmprocsrenewcertsamekey.mspx

Renew a certificate with a new key: Public Key

Renew a certificate with a new key
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/19abea04-6cdf-4496-b259-52401b5eeb2f.mspx

0.078 seconds

Results 1 - 10 **Next** >

Advanced Search|Search Preferences|Search Help

Search Microsoft.com for

renew certificate Go

Search Microsoft.com Worldwide

Choose a different location

Didn't find it here?

Search the entire Internet using MSN Search

To renew a certificate with the same key

1. Open an MMC console that contains Certificates.
2. In the console tree, under **Personal**, click **Certificates**.

   Where?

   - Certificates - *Certificate holder*
   - Personal
   - Certificates
3. In the details pane, click the certificate you are renewing.
4. On the **Action** menu, point to **All Tasks**, and then click **Renew Certificate with Same Key** to start the Certificate Renewal wizard.
5. In the Certificate Renewal wizard, do one of the following:
   - Use the default values to renew the certificate.
   - Provide your own certificate renewal settings. You need to know the certification authority issuing the certificate.
6. After the Certificate Renewal wizard has successfully finished, click **Install Certificate**.

Note

- If you have not already created an MMC console that contains Certificates, see Related Topics.
- To perform this procedure, the view mode must be organized by Logical Certificate Stores. See Related Topics.
- Once renewed, the old certificate will be archived. For information on how to view archived certificates, see Related Topics.
- You can use this procedure to request certificates from an enterprise certification authority only. To request certificates from a stand-alone certification authority, you need to request certificates via Web pages. A Windows 2000 certification authority has its Web pages located at http:*servername*\certsrv, where *servername* is the name of the Windows 2000 server hosting the certification authority.
- You can renew certificates issued to Internet Information Services (IIS) 5.0 Web servers using the Web Site Certificate wizard in Internet Services Manager instead of the Certificates snap-in. If you have IIS installed on your Windows 2000 server, for instructions on using the Web Site Certificate wizard to import the contents of a .key file, see the IIS Help topic Using the new security task wizards.

Manage certificates for your user account

Manage certificates for a computer

Manage certificates for a service

Display certificate stores in Logical Store mode

Display archived certificates

Renew a certificate with a new key


Using Windows 2000 Certificate Services Web pages


Send Us Your Feedback

To renew a certificate with a new key

1. Open an MMC console that contains Certificates.
2. In the console tree, under **Personal**, click **Certificates**.

   Where?

   - Certificates - *Certificate holder*
   - Personal
   - Certificates
3. In the details pane, click the certificate you are renewing.
4. On the **Action** menu, point to **All Tasks**, and then click **Renew Certificate with New Key** to open the Certificate Renewal wizard.
5. In the Certificate Renewal wizard, do one of the following:
   - Use the default values to renew the certificate.
   - (For advanced users only) Provide your own certificate renewal settings. You need to know the cryptographic service provider (CSP) and the certification authority (CA) issuing the certificate.

     You can also choose to enable strong private key protection. Enabling strong private key protection will ensure that you are prompted for a password every time the private key is used. This is useful if you want to make sure that the private key is not used without your knowledge.
6. After the Certificate Renewal wizard has successfully finished, click **Install Certificate**.

Note

- If you have not already created an MMC console that contains Certificates, see Related Topics.
- To perform this procedure, the view mode must be organized by Logical Certificate Stores. See Related Topics.
- Once renewed, the old certificate will be archived. For information on how to view archived certificates, see Related Topics.
- You can use this procedure to request certificates from an enterprise certification authority only. To request certificates from a stand-alone certification authority, you need to request certificates via Web pages. A Windows 2000 certification authority has its Web pages located at http:*servername*\certsrv, where *servername* is the name of the Windows 2000 server hosting the certification authority.
- You can renew certificates issued to Internet Information Services (IIS) 5.0 Web servers using the Web Site Certificate wizard in Internet Services Manager instead of the Certificates snap-in. If you have IIS installed on your Windows 2000 server, for instructions on using the Web Site Certificate wizard to import the contents of a .key file, see the IIS Help topic Using the new security task wizards.

Manage certificates for your user account

Manage certificates for a computer

Manage certificates for a service

Display certificate stores in Logical Store mode

Display archived certificates

Renew a certificate with the same key

Using Windows 2000 Certificate Services Web pages

Send Us Your Feedback

How To Renew or Create New Certificate Signing Request While Another Certificate Is Currently Installed

View products that this article applies to.

Article ID	:	295281
Last Review	:	June 29, 2004
Revision	:	1.0

This article was previously published under Q295281

SUMMARY

This article describes how you can create a new certificate signing request (CSR) or generate a renewal request without having to remove the existing certificate from your Web site.

MORE INFORMATION

To create a new CSR or generate a renewal request while another certificate exists on your Web site, follow these steps:

1. In the Microsoft Management Console (MMC), right-click the default Web site, click **New**, and then click **Site**.
2. Create a new site and give it a temporary name.
3. Right-click the new site, click **Properties**, click the **Directory Security** tab, and then click **Server certificate**.
4. Select **Create new certificate** and follow the wizard to create a new CSR. When prompted, select **Prepare the request now but send it later**.
5. Use the CSR that you just created to request a new certificate from the certificate authority (CA) that issued the original certificate.

   NOTE: If you are renewing a VeriSign certificate, see the following Web site:

   http://www.verisign.com/repository/digidren.html

   If you are unable to renew the certificate by using this Web site, you can reach VeriSign's renewal department at the following e-mail address or telephone numbers:

   E-mail: renewal@verisign.com
   Technical Support: (877) 438-8776
   Sales: (650) 429-3347
6. When you receive the certificate from VeriSign or another third-party CA, save it to your hard drive. Remember the serial number of this certificate and where you save it.

Article Translations
French

Related Support Centers
- Internet Information Services 5.0

Other Support Options
- Contact Microsoft
  Phone Numbers, Support Options and Pricing, Online Help, and more.
- Customer Service
  For non-technical assistance with product purchases, subscriptions, online services, events, training courses, corporate sales, piracy issues, and more.
- Newsgroups
  Pose a question to other users. Discussion groups and Forums about specific Microsoft products, technologies, and services.

Page Tools
Print this page
E-mail this page
Microsoft Worldwide
Save to My Support Favorites
Go to My Support Favorites
Send Feedback
Sign In

7.	Right-click the temporary site that you created in step 2, click **Properties**, click the **Directory Security** tab, click **Server certificate**, and then click **Next**. Follow the wizard. When prompted, select **Process the pending request**.
8.	After the certificate has been installed, click **OK**, and then stop and start the Web site.
9.	Right-click the temporary site that you created in step 2, click **Properties**, click **Directory Security**, and then click **Server certificate**.
10.	Select **Remove the current certificate** and follow the wizard. This removes the certificate from IIS, but the certificate remains in the certificate store.
11.	Right-click the Web site that has the original server certificate installed (that is, the certificate that you are renewing or replacing), click **Properties**, click **Directory Security**, click **Server certificate**, and then select **Replace the current certificate**.
12.	Select the certificate that you just installed. If you see duplicate certificate names, make sure that you select the certificate that matches the serial number that you noted in step 6.

NOTE: The list of available certificates is populated from the personal certificate store, which is located under **Certificates (Local Computer)** in the MMC. To view the personal certificate store, add the **Certificates** snap-in for the **Computer Account** to your MMC.

NOTE: If IIS does not display the new certificate, you may need to copy it from the personal certificate store that is located under **Certificates - Current User** in the MMC into the personal certificate store that is located under **Certificates (Local Computer)**. To view the personal certificate store, add the **Certificates** snap-in for the **User Account** to your MMC.

APPLIES TO

- Microsoft Internet Information Services 5.0

Back to the top

Keywords: kbinfo KB295281

To renew a certificate with a new key

1.Open Certificates.
2.In the console tree, under **Personal**, click **Certificates**.

Where?

Certificates - *Certificate holder* > Personal > Certificates

3.In the details pane, click the certificate you are renewing.
4.On the **Action** menu, point to **All Tasks**, and then click **Renew Certificate with New Key** to open the Certificate Renewal Wizard.
5.In the Certificate Renewal Wizard, do one of the following:

- Use the default values to renew the certificate.
- (For advanced users only) Provide your own certificate renewal settings. You need to know the cryptographic service provider (CSP) and the certification authority (CA) issuing the certificate.

  You need to select the key length (measured in bits) of the public key associated with the certificate.

  You can also choose to enable strong private key protection. Enabling strong private key protection ensures that you are prompted for a password every time the private key is used. This is useful if you want to make sure that the private key is not used without your knowledge.

6.After the Certificate Renewal Wizard has successfully finished, click **OK**.

Note

- To open Certificates, click **Start**, click **Run**, type **mmc**, and then click **OK**. On the **File** menu, click **Open**, click the console that you want to open, and then click **Open**. Then, in the console tree, click **Certificates**.
- If you have not already created an MMC console that contains Certificates, see Related Topics.
- To perform this procedure, the view mode must be organized by Logical Certificate Stores. See Related Topics.
- Once renewed, the old certificate will be archived. For information on how to view archived certificates, see Related Topics.

- You can use this procedure to request certificates from an enterprise certification authority only. To request certificates from a stand-alone certification authority, you need to request certificates via Web pages. A Windows certification authority has its Web pages located at http://*ServerName*/Certsrv, where *ServerName* is the name of the server that hosts the certification authority.

Top of page

Related Topics

- Using Windows 2000 Certificate Services Web pages
- Renew a certificate with the same key
- Display archived certificates
- Display certificate stores in Logical Store mode
- Manage certificates for a service
- Manage certificates for a computer
- Manage certificates for your user account

Top of page

To renew a certificate with the same key

1.Open Certificates.
2.In the console tree, under **Personal**, click **Certificates**.

Where?

Certificates - *Certificate holder* > Personal > Certificates

3.In the details pane, click the certificate you are renewing.
4.On the **Action** menu, point to **All Tasks**, and then click **Renew Certificate with Same Key** to start the Certificate Renewal Wizard.
5.In the Certificate Renewal Wizard, do one of the following:

- Use the default values to renew the certificate.
- Provide your own certificate renewal settings. You need to know the certification authority issuing the certificate.

6.After the Certificate Renewal Wizard has successfully finished, click **OK**.

Note

- To open Certificates, click **Start**, click **Run**, type **mmc**, and then click **OK**. On the **File** menu, click **Open**, click the console that you want to open, and then click **Open**. Then, in the console tree, click **Certificates**.
- If you have not already created an MMC console that contains Certificates, see Related Topics.
- To perform this procedure, the view mode must be organized by Logical Certificate Stores. See Related Topics.
- Once renewed, the old certificate will be archived. For information on how to view archived certificates, see Related Topics.
- You can use this procedure to request certificates from an enterprise certification authority only. To request certificates from a stand-alone certification authority, you need to request certificates via Web pages. A Windows certification authority has its Web pages located at http://*ServerName*/certsrv, where *ServerName* is the name of the server that hosts the certification authority.

Top of page

Related Topics

- Using Windows 2000 Certificate Services Web pages
- Renew a certificate with a new key

- Display archived certificates
- Display certificate stores in Logical Store mode
- Manage certificates for a service
- Manage certificates for a computer
- Manage certificates for your user account

Top of page

Renew a certificate with a new key

Updated: January 21, 2005

To renew a certificate with a new key

1. Open Certificates - Current User.
2. Confirm that you are in Certificate Purpose View.
3. In the console tree, under **Personal**, click **Certificates**.

   Where?

   •Certificates - *Current User*/Personal/Certificates
4. In the details pane, click the certificate you are renewing.
5. On the **Action** menu, point to **All Tasks**, and then click **Renew Certificate with New Key** to open the Certificate Renewal Wizard.
6. In the Certificate Renewal Wizard, do one of the following:

   •Use the default values to renew the certificate.
   •(For advanced users only) Provide your own certificate renewal settings. You need to know the cryptographic service provider (CSP) and the certification authority (CA) issuing the certificate.

   You need to select the key length (measured in bits) of the public key associated with the certificate.

   You can also choose to enable strong private key protection. Enabling strong private key protection ensures that you are prompted for a password every time the private key is used. This is useful if you want to make sure that the private key is not used without your knowledge.
7. After the Certificate Renewal Wizard has successfully finished, click **OK**.

Notes

Related Links

•Manage certificates for your user account
•Manage certificates for a computer
•Manage certificates for a service
•Display certificate stores in Logical Store mode
•Display archived certificates
•Renew a certificate with the same key
•Using Windows 2000 Certificate Services Web pages

- The link in the first step opens **Certificates** for the current user. To open **Certificates** for a service account or computer account, follow the directions below.
- To open Certificates, click **Start**, click **Run**, type **mmc**, and then click **OK**. On the **File** menu, click **Open**, click the console that you want to open, and click **Open**. Then, in the console tree, click **Certificates**.
- If you have not already created an MMC console that contains Certificates, see Related Topics.
- To perform this procedure, the view mode must be organized by Logical Certificate Stores. For more information, see Related Topics.
- Once renewed, the old certificate will be archived. For information on how to view archived certificates, see Related Topics.
- You can use this procedure to request certificates from an enterprise certification authority only. To request certificates from a stand-alone certification authority, you need to request certificates via Web pages. A Windows certification authority has its Web pages located at http://*ServerName*/Certsrv, where *ServerName* is the name of the server that hosts the certification authority.

Top of page

Information about functional differences

- Your server might function differently based on the version and edition of the operating system that is installed, your account permissions, and your menu settings. For more information, see Viewing Help on the Web.

Top of page

Results 11 - 20 for: **renew certificate**

All Results

View results in another search category by clicking a link in the right column...

Show Me:

All Results

Downloads

Product Information

Support & Troubleshooting

Technical Resources

Training & Books

Partner & Business Resources

Communities & Newsgroups

Microsoft News & Corporate Information

Related Links

- Building an Enterprise Root Certification Authority in Small and Medium Businesses
- Microsoft Learning Home Page

Renew a certificate with the same key: Public Key
Renew a certificate with the same key
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/7c442dc8-0c1a-432d-8969-5c04408c35fc.mspx

Common tasks for certificate users on the Internet
Task Reference Review certificate concepts. Get a certification authority's root certificate to establish trust. Get the current certificate revocation list from the certification authority. Requesting certificates Request a certificate using the certification authority's Web pages. (For advanced
http://www.microsoft.com/windows2000/en/advanced/help/sag_CM_tasks_inter.htm

Common tasks for certificate users on an intranet
Task Reference Review certificate concepts. Requesting certificates Request a certificate using the Certificate Request wizard. (For advanced requests) Request a certificate using the certification authority's Web pages. Submit a certificate request using a PKCS #10 file. Useful in cases where
http://www.microsoft.com/windows2000/en/advanced/help/sag_CM_tasks_intra.htm

The secure bindings for an IIS 6.0 Web site are removed when you install or renew a server certificate
Discusses an issue where the secure bindings for a Web site in IIS 6.0 are removed when you use the Web Server Certificate Wizard to install a new server certificate or to renew an existing certificate.
http://support.microsoft.com/default.aspx?scid=kb;en-us;896284

How to Renew Certificates That Are Used with IIS 4.0
Certificates that are installed on computers running Internet Information Server (IIS) 4.0 are usually set to expire in one year from the issue date depending on the Certificate Authority that issued them. If you have a certificate that is about...
http://support.microsoft.com/default.aspx?scid=kb;en-us;277893

How to Renew Certificates That Are Used with IIS 5.0
Certificates that are installed on computers running Internet Information Services (IIS) 5.0 are usually set to expire in one year from the issue date depending on the Certificate Authority that issued them. If you have a certificate that is about...
http://support.microsoft.com/default.aspx?scid=kb;en-us;277891

Renew a subordinate certification authority
To renew a subordinate certification authority Log on to the system as an Administrator. Open Certification Authority In the console tree, click the name of the certification authority (CA) Where? Certification Authority (computer) CA name On the
http://www.microsoft.com/windows2000/en/advanced/help/sag_CSprocs_RenewSub.htm

Renew certificates for Message Queuing
To renew certificates for Message Queuing Open Message Queuing in Control Panel. On the Security tab, under Internal Certificate, click Renew Internal Certificate. A warning appears, stating that authenticated messages that you have already sent using the current internal certificate may
http://www.microsoft.com/windows2000/en/advanced/help/msmq_sec_renew_internal_certificate.htm

Certificate Enrollment and Renewal Methods

Certificate Enrollment and Renewal Methods Windows 2000 Certificate Services supports the following certificate enrollment and renewal methods: Manual certificate requests that use the Certificate

http://www.microsoft.com/resources/documentation/Windows/2000/server/reskit/en-us/distrib/dscj_mcs_fspg.asp

Renew a root certification authority

To renew a root certification authority Log on to the system as an Administrator. Open Certification Authority In the console tree, click the name of the certification authority (CA) Where? Certification Authority (computer) CA name On the Action

http://www.microsoft.com/windows2000/en/advanced/help/sag_CSprocs_RenewRoot.htm

0.093 seconds

Results 11 - 20 < **Previous**|**Next** >

Advanced Search|Search Preferences|Search Help

Search Microsoft.com for

renew certificate Go

Search Microsoft.com Worldwide

Choose a different location

Didn't find it here?

Search the entire Internet using MSN Search

Common tasks for certificate users on the Internet

Task	Reference
Review certificate concepts.	Certificates overview
Get a certification authority's root certificate to establish trust.	To retrieve a certification authority certificate
Get the current certificate revocation list from the certification authority.	To retrieve a certificate revocation list
Requesting certificates	
Request a certificate using the certification authority's Web pages.	To submit a user certificate request via the Web
(For advanced requests) Request a certificate using the certification authority's Web pages.	To submit an advanced certificate request via the Web
Submit a certificate request using a PKCS #10 file. Useful in cases where the certificate requester does not have a network connection to the certification authority.	To request a certificate using a PKCS #10 or PKCS #7 file
Check on a pending certificate request.	To check on a pending certificate request
Managing certificates and keys	
Export a certificate.	To export a certificate
Export a certificate with a private key.	To export a certificate with the private key
Import a certificate.	To import a certificate
Renew certificates.	To renew a certificate with a new key
Renew a certificate with the same key.	To renew a certificate with the same key

Send Us Your Feedback

To request a certificate using a PKCS #10 or PKCS #7 file

1. Open Internet Explorer
2. In Internet Explorer, connect to http://*servername*/certsrv, where *servername* is the name of the Windows 2000 Web server where the certification authority (CA) you want to access is located.
3. Click **Request a certificate**, and then click **Next**. Click **Advanced request**, and click **Next**.
4. Click **Submit a certificate request using a Base-64-encoded PKCS #10 file or a renewal request using a PKCS #7 file**, and then click **Next**.
5. Do one of the following:
   - In Notepad, click **File**, click **Open**, select the PKCS #10 or PKCS #7 file, click **Edit**, click **Select all**, click **Edit**, and click **Copy**. On the Web page, click in the **Saved request** scroll box. Click **Edit** and then click **Paste** to paste the contents of certificate request into the scroll box.
   - Click **Browse** to locate the file you want to use for the certificate request. If you get a warning about the ActiveX control, click **Yes** to allow it to run, then click the **Browse** button. After locating and selecting the file you want to use for the certificate request, click **Open**. On the Web page, click **Read!** to paste the contents of the file into the scroll box. See the note about using **Browse**.
6. If you are connected to an enterprise CA, choose the certificate template you want to use.
7. Click **Submit**.
8. Do one of the following:
   - If you see the **Certificate Pending** Web page, see Related Topics below for the procedure to check on a pending certificate.
   - If you see the **Certificate Issued** Web page, click **Download certificate**. Choose to save the file to your hard disk, and then import the certificate into your certificate store For the procedure to import a certificate, see Related Topics.

Note

- To open Internet Explorer, click **Start**, point to **Programs**, and then click **Internet Explorer**.
- To open Notepad, click **Start**, point to **Programs**, point to **Accessories**, and click **Notepad**.
- In general, you use a PKCS #10 file to submit a request for a new certificate and a PKCS #7 file to submit a request to renew an existing certificate. Submitting requests with files is useful when the certificate requester is unable to submit a request online to the certification authority.
- You might need to make http://*servername* a trusted site for Internet Explorer in order to browse for a file on the computer's disk drive. To make http://*servername* a trusted site, in Internet Explorer, click **Tools**, then point to **Internet Options**, point to **Security**, point to **Trusted Sites**, and click **Sites**. Type **http://*servername***, and click **OK**.
- If you submit the request and immediately get a message asking you if you want to submit the request even though it does not contain a "BEGIN" or "END" tag, click **OK**.
- For the procedure to export a certificate to create a PKCS #7 file, see Related Topics.

Using Windows 2000 Certificate Services Web pages

Save a certificate request to a PKCS #10 file

 Export a certificate

 Import a certificate

---

 Send Us Your Feedback

To renew a root certification authority

1. Log on to the system as an Administrator.
2. Open Certification Authority
3. In the console tree, click the name of the certification authority (CA)

   Where?

   - Certification Authority (*computer*)
   - *CA name*
4. On the **Action** menu, point to **All Tasks**, and click **Renew CA Certificate**.
5. Do one of the following:
   - Click **Yes** if you want to generate a new public and private key pair for the certification authority's certificate.
   - Click **No** if you want to reuse the current public and private key pair for the certification authority's certificate.

Note

- To open Certification Authority, click **Start**, point to **Programs**, point to **Administrative Tools**, and then click **Certification Authority**.

Working with MMC console files

Renewing certification authorities

Renew a subordinate certification authority

Send Us Your Feedback

Results 1 - 10 for: **pkcs#10**

All Results

View results in another search category by clicking a link in the right column...

Show Me:

All Results

Downloads

Product Information

Support & Troubleshooting

Technical Resources

Training & Books

Partner & Business Resources

Communities & Newsgroups

Microsoft News & Corporate Information

Microsoft Windows XP - Request a certificate using a PKCS #10 or PKCS #7 file

Open Internet Explorer In Address, type http://servername/certsrv, where servername is the name of the Windows 2000 Web server where the certification authority (CA) you want to access is located. Click Request a certificate, and then click Next.
http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_cswprocs_reqfile.mspx

Microsoft Windows XP - Request a certificate using a PKCS #10 or PKCS #7 file

Open Internet Explorer In Address, type http://servername/certsrv, where servername is the name of the Windows 2000 Web server

where the certification authority (CA) you want to access is located. Click Request a certificate, and then click Next.
http://www.microsoft.com/technet/prodtechnol/winxppro/proddocs/sag_cswprocs_reqfile.mspx

Request a certificate using a PKCS #10 or PKCS #7 file: Public Key

Request a certificate using a PKCS #10 or PKCS #7 file
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/03e6c8ac-818d-4660-87ca-9be116997cdd.mspx

Request a certificate from a Windows Server 2003 CA using a PKCS #10 or PKCS #7 file

Request a certificate from a Windows Server 2003 CA using a PKCS #10 or PKCS #7 file
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/7e918937-b0cc-4094-9009-4e0798986bef.mspx

Creating a PKCS #10 Request

Microsoft Windows CE .NET 4.2 Creating a PKCS #10 Request After a client is authenticated, Enroll.exe creates a base64-encoded PKCS #10 certificate request message and sends
http://msdn.microsoft.com/library/en-us/wcedsn40/html/cetskcreatingpkcs10request.asp

Creating a PKCS #10 Request

After a client is authenticated, Enroll.exe creates a base64-encoded PKCS #10 certificate request message and sends it to the Windows 2000 Certificate Server.
http://msdn.microsoft.com/library/en-us/wcesecurity5/html/wce50concreatingapkcs10request.asp

Save a certificate request to a Windows Server 2003 CA in a PKCS #10 file:

Save a certificate request to a Windows Server 2003 CA in a PKCS #10 file
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/a0985346-ed4c-4429-bf85-73d75296ccad.mspx

Save a certificate request to a PKCS #10 file: Public Key

Save a certificate request to a PKCS #10 file
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/87779f46-b50c-41fe-910a-083f34457242.mspx

Microsoft Windows XP - Save a certificate request to a PKCS #10 file

Open Internet Explorer In Address, type http://servername/certsrv, where servername is the name of the Windows 2000 Web server where the certification authority you want to access is located. Click Request a certificate, and then click Next.
http://www.microsoft.com/technet/prodtechnol/winxppro/proddocs/sag_cswprocs_makereqfile.mspx

Microsoft Windows XP - Save a certificate request to a PKCS #10 file

Open Internet Explorer In Address, type http://servername/certsrv, where servername is the name of the Windows 2000 Web server where the certification authority you want to access is located. Click Request a certificate, and then click Next.
http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_cswprocs_makereqfile.mspx

0.124 seconds

Results 1 - 10 **Next** >

Search Microsoft.com for

pkcs#10 Go

Search Microsoft.com Worldwide

Choose a different location

Didn't find it here?

Search the entire Internet using MSN Search

To request a certificate using a PKCS #10 or PKCS #7 file

1. Open Internet Explorer
2. In **Address**, type http://*servername*/certsrv, where *servername* is the name of the Windows 2000 Web server where the certification authority (CA) you want to access is located.
3. Click **Request a certificate**, and then click **Next**. Click **Advanced request**, and then click **Next**.
4. Click **Submit a certificate request using a Base-64-encoded PKCS #10 file or a renewal request using a PKCS #7 file**, and then click **Next**.
5. Do one of the following:

   - Open Notepad On the **File** menu, click **Open**. Select the PKCS #10 or PKCS #7 file and click **Open**. On the **Edit** menu, click **Select all**, and then, on the **Edit** menu, click **Copy**. On the Web page, click in the **Saved request** scroll box. On the **Edit** menu, click **Paste** to paste the contents of certificate request into the scroll box.
   - Click **Browse** to locate the file you want to use for the certificate request. If you get a warning about the ActiveX control, click **Yes** to allow it to run, then click **Browse**. After locating and selecting the file you want to use for the certificate request, click **Read!**. On the Web page, click **Read!** to paste the contents of the file into the scroll box. See the note about using **Browse**.
6. If you are connected to an enterprise CA, choose the certificate template you want to use.
7. Click **Submit**.
8. Do one of the following:

   - If you see the **Certificate Pending** Web page, see Related Topics below for the procedure to check on a pending certificate.
   - If you see the **Certificate Issued** Web page, click **Download certificate**. Choose to save the file to your hard disk, and then import the certificate into your certificate store For the procedure to import a certificate, see Related Topics.

Note

- To open Internet Explorer, click **Start**, point to **All Programs**, and then click **Internet Explorer**.
- To open Notepad, click **Start**, point to **All Programs**, point to **Accessories**, and then click **Notepad**.

- In general, you use a PKCS #10 file to submit a request for a new certificate and a PKCS #7 file to submit a request to renew an existing certificate. Submitting requests with files is useful when the certificate requester is unable to submit a request online to the certification authority.
- If you use a PKCS #7 file to submit a request to renew an existing certificate, the renewal request file must be generated by a program. This procedure will not work with a PKCS #7 file that merely contains an exported certificate.
- You might need to make http://*servername* a trusted site for Internet Explorer in order to browse for a file on the computer's disk drive. To make http://*servername* a trusted site, in Internet Explorer, click **Tools**, then point to **Internet Options**, point to **Security**, point to **Trusted Sites**, and click **Sites**. Type **http://*servername***, and click **OK**.
- If you submit the request and immediately get a message asking you if you want to submit the request even though it does not contain a "BEGIN" or "END" tag, click **OK**.
- For the procedure to export a certificate to create a PKCS #7 file, see Related Topics.

Top of page

Related Topics

- Importing and exporting certificates
- Import a certificate
- Export a certificate
- Save a certificate request to a PKCS #10 file
- Using Windows 2000 Certificate Services Web pages

Top of page

HOW TO: Install a Certificate for Use with IP Security

View products that this article applies to.

Article ID	:	253498
Last Review	:	September 22, 2003
Revision	:	4.0

This article was previously published under Q253498

On This Page

SUMMARY

When IP Security (IPSec) is configured to use a certification authority (CA) for mutual authentication, you must obtain a local computer certificate. You can obtain this certificate from a third-party CA or you can install Certificate Services in Windows to create your own CA. This article describes how to install a local computer certificate for use with IPSec from a stand-alone Windows CA.

The request for the local computer certificate is requested by using HTTP. Because a local computer certificate must be used with IPSec, you must submit an advanced request to the CA to specify this.

Back to the top

Installing a local Computer Certificate from a Stand-Alone Windows Certificate Authority

1. The request is a Web address that contains the IP address or name of the Certificate server, with "/certsrv" appended. In your Web browser, type the following Web address

   http://*IP address of CA*/certsrv

   Where *IP address of CA* is the IP address or name of the Certificate server.
2. In the initial Welcome screen of the Certificate server, click **Request a certificate**, and then click **Next**.
3. In the "Choose Request Type" screen, click **Advanced request**, and then click **Next**.
4. In the "Advanced Certificate Requests" screen, click **Submit a certificate request to this CA using a form**, and then click **Next**.
5. In the "Advanced Certificate Request" screen, type your name and your e-mail name in the appropriate boxes.
6. Under **Intended Purpose**, select **Client Authentication Certificate** or **IPSec Certificate**. If you choose **IPSec Certificate**, then this certificate will only be used for IPSec.
7. Under **Key Options**, click **Microsoft Base Cryptographic Provider v1.0**, **Signature** for **Key Usage** and **1024** for **Key Size**.
8. Leave the **Create new key set** option enabled (you can clear the **Container Name** check box unless you want to specify a specific name), and then click **Use local machine store**.
9. Leave all the other options set to the default value unless you need to make a specific change.
10. Click **Submit**.
11. If the Certificate Authority is configured to issue certificates automatically, the "Certificate Issued" screen should appear. Click **Install this Certificate**. The "Certificate Installed" screen should appear with the message "Your new certificate has been successfully installed."
12. If the Certificate Authority is not configured to issue certificates automatically a "Certificate Pending" screen appears, requesting that you wait for an administrator to issue the certificate that was requested. To retrieve a certificate that an administrator has issued, return to the Web address and click **Check on a pending certificate**. Click the requested certificate, and then click **Next**. If the certificate is still pending, the "Certificate Pending" screen appears. If the certificate has been issued, the "Install this Certificate" screen appears.

Back to the top

Installing a Local Computer Certificate from an Enterprise Windows 2000 Certificate Authority

1. The request is a Web address that contains the IP address or name of the Certificate server, with **/certsrv** appended. In your Web browser, type the following Web address: **http://*IP address of CA*/certsrv**

   Where *IP address of CA* is the IP address or name of the Certificate server.
2. If the machine you are using is not logged onto the domain already, a prompt to supply domain credentials appears.
3. In the initial Welcome screen of the Certificate server, click **Request a Certificate**, and then click **Next**.
4. In the **Choose Request Type** screen, click **Advanced Request**, and then click **Next**.
5. In the **Advanced Certificate Requests** screen, click **Submit a certificate request to this CA using a form**, and then click **Next**.
6. In the **Advanced Certificate Request** screen for the **Certificate Template** option, select **Administrator**.
7. Under **Key** Options, click **Microsoft Base Cryptographic Provider v1.0**, **Signature** for **Key Usage** and **1024** for **Key Size**.
8. Leave the **Create new key set** option enabled (you can clear the Container Name check box unless you want to specify a specific name), and then click **Use local machine store**.

9.	Leave all the other options set to the default value unless you need to make a specific change.
10.	Click **Submit**.
11.	The **Certificate Issued** screen should appear. Click **Install this Certificate**. The **Certificate Installed** screen should appear with the message: **Your new certificate has been successfully Installed**

Back to the top

Verifying That the Local Computer Certificate Has Been Installed

After the certificate is installed, verify the location of the certificate by using the Certificate (Local Computer) snap-in in Microsoft Management Console (MMC). Your certificate should appear under **Personal**.

If the certificate you have installed does not appear here, the certificate was installed as a "User certificate request," or you did not click **Use local machine store** within the advanced request.

Back to the top

REFERENCES

For information about installing Certificate Services in Windows, see the following article in the Microsoft Knowledge Base:

> 231881 How to Install/Uninstall a Public Key Certificate Authority

For more information, see the "Step-by-Step Guide to End-to-End Security: An Introduction to Internet Protocol" document located at the following Microsoft Web site:

> http://www.microsoft.com/windows2000/techinfo/planning/security/ipsecsteps.asp

Back to the top

APPLIES TO

- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Professional Edition

Back to the top

Keywords: kbhowtomaster kbipsec kbenv KB253498

Back to the top

Results 11 - 20 for: **install certificate**

All Results

View results in another search category by clicking a link in the right column...

Show Me:

All Results

Downloads

Product Information

Support & Troubleshooting

Technical Resources

Training & Books

Partner & Business Resources

Communities & Newsgroups

Microsoft News & Corporate Information

Related Links

- Building an Enterprise Root Certification Authority in Small and Medium Businesses
- Platform SDK: Windows Installer
- Microsoft Learning Home Page

XFOR: Using a Verisign Certificate with Exchange Server Secure Sockets Layer
When you set up Exchange Server to use Secure Sockets Layer (SSL) for Internet protocols such as Network News Transfer Protocol (NNTP), Simple Mail Transfer Protocol (SMTP), Post Office Protocol version 3 (POP3), and Internet Message Access...
http://support.microsoft.com/default.aspx?scid=kb;en-us;249029

Installing a Root Certificate
To install a root certificate in a Windows Mobile-based device after manufacture you must do the following: first ensure that it is a Base-64 encoded certificate, then place it in a provisioning XML document containing the code required to install the
http://msdn.microsoft.com/library/en-us/mobilesdk5/html/wce51conCreatingProvisioningXMLForSignedBinaryFiles.asp

Microsoft Office Assistance: Step 10: Install the Server Certificate on the Remaining Network Load-Balanced Front-End Web Servers
( ) Client Deployment Server Deployment Related Web Sites Worldwide Feedback Security Chapter: Go Step 10: Install the Server Certificate on the Remaining Network Load-Balanced Front-End Web Servers On each remaining network load-balanced front-end Web server, you must do the following:
http://office.microsoft.com/en-us/assistance/HA011648211033.aspx

Microsoft Office Assistance: Step 4: Install the Server Certificate on Your Primary Front-End Web Server
( ) Client Deployment Server Deployment Related Web Sites Worldwide Feedback Security Chapter: Go Step 4: Install the Server Certificate on Your Primary Front-End Web Server You must install the server certificate from the previous step on the primary front-end Web server. Open Internet
http://office.microsoft.com/en-us/assistance/HA011648301033.aspx

Machine certificates for L2TP over IPSec VPN connections
The use of machine certificates for machine-level authentication of VPN clients and VPN server is required for L2TP over IPSec-based VPN connections. In order to create an L2TP over IPSec connection, you must install a machine certificate, also known as a computer certificate, on the VPN client
http://www.microsoft.com/windows2000/en/advanced/help/sag_VPN_us26.htm

Install a Server Certificate (IIS 6.0)
Install a Server Certificate (IIS 6.0)
http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/a2f35fcd-d3b6-4f39-ba93-041a86f7e17f.mspx

Install Certification Authorities

Install Certification Authorities You must install the CA hierarchies necessary to provide the required certificate services for your organization. Certification hierarchies with Windows 2000 CAs can
http://www.microsoft.com/resources/documentation/windows/2000/server/reskit/en-us/distrib/dscj_mcs_tpuv.asp

HOW TO: Install Imported Certificates on a Web Server in Windows Server 2003
View products that this article applies to. Article ID : 816794 Last Review : December 19, 2003 Revision : 4.0 For a Microsoft Windows 2000 version of this article, see (http://support.microsoft.com/kb/310178/EN-US/). On This Page IN THIS TASK Summary Install the Certificates Import
http://support.microsoft.com/default.aspx?scid=kb;en-us;816794

FP2000: VeriSign Trial Certificate Is Not Recognized
If a 40-bit trial certificate from VeriSign has been installed on a Web server, you cannot open the Web site with Microsoft FrontPage. You may receive the following message: An unrecognized certificate issuing authority provided the security...
http://support.microsoft.com/default.aspx?scid=kb;en-us;264923

Installing Certificates for VPN Connections: Virtual Private Network (VPN)
Installing Certificates for VPN Connections
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/DepKit/b6eb6bd9-534d-4696-9850-d35cdcdfab7c.mspx

0.125 seconds

Results 11 - 20 < **Previous**|**Next** >

Search Microsoft.com for

install certificate Go

Search Microsoft.com Worldwide

Choose a different location

Didn't find it here?

Search the entire Internet using MSN Search

Windows Mobile Version 5.0
SDK

Installing a Root Certificate

Send Feedback on this topic to the authors

To install a root certificate in a Windows Mobile-based device after manufacture you must do the following: first ensure that it is a Base-64 encoded certificate, then place it in a provisioning XML document containing the code required to install the certificate in the appropriate certificate store (in this case ROOT), finally, you must send the provisioning XML document to the device.

To install a root certificate on a Windows Mobile-based device

- Convert the root certificate (.cer) to a Base-64 Encoded x.509 certificate. For more information, see Converting a Root Certificate.
- Create the provisioning XML to install the certificate in the appropriate certificate store on the device. For more information, see Creating a Provisioning XML Document For The Root Certificate or, if you are provisioning through an DM server, see Creating a Provisioning XML Document For The Root Certificate (OMA DM)
- Deliver the certificate to the device.
  After you create the provisioning file you have the following options for delivering the file to a Windows Mobile-based device:
  - You can send the provisioning file over the air using an OMA DM Server. For more information see Provisioning OTA Through an OMA DM Server.
  - You can wrap the provisioning file in a .cpf file and send it using one of these delivery methods: Internet Explorer Mobile, ActiveSync, SI/SL, or Storage Card. For more information see Creating a .cpf File and Delivering Applications.

    Note Microsoft recommends that you package and sign provisioning documents in the CAB Provisioning Format (.cpf). An XML provisioning document may not install on a Windows Mobile-based device if the file containing the document is not signed. For more information about a .cpf file, see Cab Provisioning Format (CPF) File
- You can send the provisioning file over-the-air using an OMA Client Provisioning (formerly WAP) server. For more information see Provisioning OTA Through a WAP Push.

See Also

CertificateStore Configuration Service Provider | Certificate Management in Windows Mobile-based Devices | Cab Provisioning Format (CPF) File

Last updated on Wednesday, July 13, 2005

Send Feedback on this topic to the authors

Step 10: Install the Server Certificate on the Remaining Network Load-Balanced Front-End Web Servers

On each remaining network load-balanced front-end Web server, you must do the following:

1. Open Internet Information Services (IIS) Manager.
2. In the console tree, expand the computer name node.
3. Expand the **Web Sites** node, right-click **Default Web Site**, and then click **Properties**.
4. On the **Directory Security** tab, in the **Secure communications** section, click **Server Certificate**.
5. On the Welcome to the Web Server Certificate Wizard page, click **Next**.
6. On the Server Certificate page, click **Import a certificate from a .pfx file**, and then click **Next**.
7. On the Import Certificate page, do the following:
    1. Click **Browse**, and navigate to the location of the .pfx file that you exported in the "Export the Server Certificate for Use on the Primary Front-End Web Server" step.
    2. Click **Open**.
    3. Click **Next**.
8. On the Import Certificate Password page, in the **Password** box, type the password that you entered when you exported the certificate on the primary front-end Web server, and then click **Next**.
9. On the SSL Port page, in the **SSL port this web site should use** box, type **443**, and then click **Next**.

    Note If you chose an alternate port number on the primary front-end Web server, use that same number on the other front-end servers.

10. On the Imported Certificate Summary page, click **Next**.
11. On the Completing the Web Server Certificate Wizard page, click **Finish**.
12. Click **OK** to close the **Default Web Site Properties** dialog box.
13. Perform Step 5, Step 6, Step 7 (needed if you required SSL on the primary front-end Web server), and Step 8 (needed if you required SSL on the primary front-end Web server).

You might need to install the certificate authority root. If you receive a certificate warning that states, "The security certificate was issued by a company you have chosen not to trust," you must install the Trust Root Authority, as described in the "Fix Errors by Downloading and Installing the Certificate Authority Root" step, later in this paper. If you receive any other warning, review the steps you used to create and install the certificate, and try again.

Next Topic

Printer-friendly version

Install a Server Certificate (IIS 6.0)

Web server certificates contain information about the server that allows the client to positively identify the server over a network before sharing sensitive information. This process is called *authentication*. If you use Secure Sockets Layer (SSL) to protect confidential information exchanged between the Web server and the client and you have exported the certificates from the source server to the target server, the server certificate needs to be installed on the Web server before you can assign the server certificate to Web sites for use with SSL

Requirements

•Credentials: Membership in the Administrators group on the local computer.
•Tools: Certificates MMC snap-in.

Top of page

Recommendation

As a security best practice, log on to your computer using an account that is not in the Administrators group, and then use the **Run as** command to run IIS Manager as an administrator. At the command prompt, type **runas /user:***administrative_accountname* **mmc %systemroot%\system32\inetsrv\iis.msc**.

Top of page

Procedures

To add the Certificates Snap-in to MMC

1.In the **Run** dialog box, type **mmc**, and then click **OK**.

The Microsoft Management Console appears.
2.On the **File** menu, click **Add/Remove Snap-in**.
3.On the **Standalone** tab, click **Add**.
4.In the **Available Standalone Snap-ins** list box, click **Certificates**, and then click **Add**.
5.Click the **Computer account** option, and then click **Next**.
6.Click the **Local computer (the computer this console is running on)** option, and then click **Finish**.

7.Click **Close**, and then click **OK**.

To install a server certificate on a Web server

1.In MMC, open the **Certificates** snap-in.
2.In the console tree, click the logical store where you want to import the certificate.

The default location of the logical store for certificates is on the Console Root in the **Certificates (Local Computer)/ Personal/Certificates** folder.
3.On the **Action** menu, point to **All Tasks**, and then click **Import** to start the Certificate Import Wizard.

Important

You should only import certificates obtained from trusted sources. Importing an altered or unreliable certificate could compromise the security of any system component that uses the imported certificate.
4.Click **Next**.
5.Type the name of the file that contains the certificate to be imported, or click **Browse** and navigate to the file.

Certificates can be stored in several different file formats. The most secure format is Public-Key Cryptography Standard (PKCS) #12, an encryption format that requires a password to encrypt the private key. It is recommended that you send certificates using this format for optimum security.

If the certificate file is in a format other than PKCS #12, skip to step 8.

If the certificate file is in the PKCS #12 format, do the following:

- In the **Password** box, type the password used to encrypt the private key. You must have access to the password that was originally used to secure the file.
- (Optional) If you want to be able to use strong private key protection, select the **Enable strong private key protection** check box, if available.
- (Optional) If you want to back up or transport your keys at a later time, select the **Mark key as exportable** check box.

6.Click **Next**.
7.In the **Certificate Store** dialog box, do one of the following:

- If the certificate should be automatically placed in a certificate store based on the type of certificate, select **Automatically select the certificate store based on the type of certificate**.
- If you want to specify where the certificate is stored, select **Place all certificates in the following store**, click **Browse**, and select the certificate store to use.

8.Click **Next**, and then click **Finish**.

The file from which you import certificates remains intact after you have completed importing the certificates. You can use Windows Explorer to delete the file if it is no longer needed.

Top of page

HOW TO: Install Imported Certificates on a Web Server in Windows Server 2003

View products that this article applies to.

Article ID	:	816794
Last Review	:	December 19, 2003
Revision	:	4.0

For a Microsoft Windows 2000 version of this article, see 310178.

On This Page

IN THIS TASK

Summary

This step-by-step article describes how to import a Web site certificate into the certificate store of the local computer and assign the certificate to the Web site.

Install the Certificates

The Windows 2003 Internet Information Server (IIS) 6.0 supports Secure Sockets Layer (SSL) communications. A whole Web site, a folder on the Web site, or a particular file that is located in a folder on the site can require a secure SSL connection. However, before the Web server can support SSL sessions, a Web site certificate must be installed.
You can use one of the following methods to install a certificate in IIS 6.0:

- Make an online request by using the IIS Web Server Certificate Wizard and install the certificate at the time of the request.
- Make an offline request by using the IIS Web Server Certificate Wizard and obtain and install the certificate later.
- Request a certificate without using the IIS Web Server Certificate Wizard.

Note If you use the second or third method, you must install the certificate manually.

To install the Web site certificate, you must complete the following tasks:

- Import the certificate into the computer's certificate store.
- Assign the installed certificate to the Web site.

Import the Certificate into the Local Computer Store

To import the certificate into the local computer store, follow these steps:

1. On the IIS 6.0 Web server, click **Start**, and then click **Run**.
2. In the **Open** box, type **mmc**, and then click **OK**.
3. On the **File** menu click **Add/Remove snap-in**.
4. In the **Add/Remove Snap-in** dialog box, click **Add**.
5. In the **Add Standalone Snap-in** dialog box, click **Certificates**, and then click **Add**.
6. In the **Certificates snap-in** dialog box, click **Computer account**, and then click **Next**.
7. In the **Select Computer** dialog box, click **Local computer: (the computer this console is running on)**, and then click **Finish**.
8. In the **Add Standalone Snap-in** dialog box, click **Close**.
9. In the **Add/Remove Snap-in** dialog box, click **OK**.
10. In the left pane of the console, double-click **Certificates (Local Computer)**.
11. Right-click **Personal**, point to **All Tasks**, and then click **Import**.
12. On the **Welcome to the Certificate Import Wizard** page, click **Next**.
13. On the **File to Import** page, click **Browse**, locate your certificate file, and then click **Next**.
14. If the certificate has a password, type the password on the **Password** page, and then click **Next**.
15. On the **Certificate Store** page, click **Place all certificates in the following store**, and then click **Next**.
16. Click **Finish**, and then click **OK** to confirm that the import was successful.

Assign the Imported Certificate to the Web Site

1. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the left pane, click your server.
3. In the right pane, double-click **Web Sites**.
4. In the right pane, right-click the Web site you want to assign the certificate to, and then click **Properties**.
5. Click **Directory Security**, and then click **Server Certificate**.
6. On the **Welcome to the Web Certificate Wizard** page, click **Next**.
7. On the **Server Certificate** page, click **Assign an existing certificate**, and then click **Next**.
8. On the **Available Certificates** page, click the installed certificate you want to assign to this Web site, and then click **Next**.
9. On the **SSL Port** page, configure the SSL port number. The default port of 443 is appropriate for most situations.
10. Click **Next**.
11. On the **Certificate Summary** page, review the information about the certificate, and then click **Next**.
12. On the **Completing the Web Server Certificate Wizard** page, click **Finish**, and then click **OK**.

You can now configure Web site elements to use secure communications.

Back to the top

APPLIES TO

- Microsoft Windows Server 2003, Datacenter Edition
- Microsoft Windows Server 2003, Enterprise Edition
- Microsoft Windows Server 2003, Standard Edition
- Microsoft Windows Server 2003, Web Edition
- Microsoft Windows Server 2003, Datacenter Edition
- Microsoft Windows Server 2003, Enterprise Edition for Itanium-based Systems
- Microsoft Internet Information Services 6.0

Back to the top

Keywords: kbhowto KB816794

Back to the top

Results 1 - 10 for: **pkcs#12**

All Results

View results in another search category by clicking a link in the right column...

Show Me:

All Results

Downloads

Product Information

Support & Troubleshooting

Technical Resources

Training & Books

Partner & Business Resources

Communities & Newsgroups

Microsoft News & Corporate Information

PKCS #12 File Types: Portable Protected Keys in .NET (Cryptographic API Technical Articless)

This article provides the code framework for accessing the certificates, public keys, and private keys within pfx/p12 files from .NET Framework code using P/Invoke to CryptoAPI. It requires only the .NET Framework version 1.1 with no additional support.
http://msdn.microsoft.com/library/en-us/dncapi/html/pkcs12.asp

Importing and exporting certificates

The Certificates snap-in provides administrative tools to export and import certificates , including their certification paths and private keys , if needed. You can export certificates to or import certificates from PKCS #12 files, PKCS #7 files, and binary-encoded X.509 certificate files.
http://www.microsoft.com/windows2000/en/advanced/help/sag_CMimportExport.htm

Microsoft Windows XP - Importing and exporting certificates

The Certificates snap-in provides administrative tools to export and import certificates, including their certification paths and private keys, if needed.
http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_cmimportexport.mspx

Importing and exporting certificates: Public Key
Importing and exporting certificates
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/b260a53a-dd02-4a3f-8fdf-9b24a295ee5b.mspx

Microsoft Windows XP - Importing and exporting certificates
The Certificates snap-in provides administrative tools to export and import certificates, including their certification paths and private keys, if needed.
http://www.microsoft.com/technet/prodtechnol/winxppro/proddocs/sag_cmimportexport.mspx

Backing Up and Restoring the Certificate Services Private Key [Security]
You cannot use the Certadm.dll's backup and restore functions to back up the Certificate Services private key(s).
http://msdn.microsoft.com/library/en-us/seccrypto/security/backing_up_and_restoring_the_certificate_services_private_key.asp

Trusted root certification authority policy
To establish a trusted root certification authority (CA) using Group Policy , the Group Policy object that you create must have access to the root certificate This requires that you import a copy of the root authority certificate. You can do this using the procedure To import a root CA
http://www.microsoft.com/windows2000/en/advanced/help/sag_PKPuseCertRoot.htm

Object IDs Associated with Microsoft Cryptography
This article describes the Object ID numbers (OIDs) that are defined for Microsoft. In this scope, OIDs are numerical values that enable programs to determine if a certificate is valid for a particular use. The OIDs can be used to represent...
http://support.microsoft.com/default.aspx?scid=kb;en-us;287547

Certutil tasks for backing up and restoring certificates
Certutil tasks for backing up and restoring certificates
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/ServerHelp/2272f3c8-1103-402b-a945-3dc0a1b489fb.mspx

How to Back Up Your Web Server Certificate
Security How to Back Up Your Web Server Certificate Because IIS 5.0 leverages Windows security and security tools, you can use the Certificate Manager tool in Windows 2000 Server to export
http://www.microsoft.com/resources/documentation/windows/2000/server/reskit/en-us/iisbook/c09_how_to_back_up_your_web_server_certificate.asp

0.093 seconds

Results 1 - 10 **Next** >

Advanced Search|Search Preferences|Search Help

Search Microsoft.com for

pkcs#12 Go

Search Microsoft.com Worldwide

Choose a different location

Didn't find it here?

Search the entire Internet using MSN Search

Welcome to the MSDN Library

PKCS #12 File Types: Portable Protected Keys in .NET

Michel I. Gallant, Ph.D.
JavaScience Consulting

March 2004

Applies to:
Microsoft® .NET Framework version 1.1
Microsoft® Visual Studio® .NET
Microsoft® Windows® security

Summary: This article provides the code framework for accessing the certificates, public keys, and private keys within pfx/p12 files from .NET Framework code using P/Invoke to CryptoAPI. It requires only the .NET Framework version 1.1 with no additional support. (11 printed pages)

Download the PKCS.exe code sample.

Contents

Introduction

Digital signature generation and enveloped encryption using asymmetric keys in Windows most commonly involves the use of RSA key pairs stored and protected under installed CryptoAPI CSP (Cryptographic Service Providers). In EncryptTo/DecryptTo: Encryption in .NET with CryptoAPI Certificate Stores, we discussed RSA enveloping (encryption and decryption of secret, symmetric session keys). The RSA keys were obtained from the CryptoAPI key containers associated with installed certificates in CryptoAPI certificate stores. However, anyone who has purchased a certificate for SSL, secure e-mail, or code-signing usage from any of the well-known Certificate Authorities (CA) will be familiar with the Certificate Export Wizard, which optionally can include the associated private key into a password-protected file:

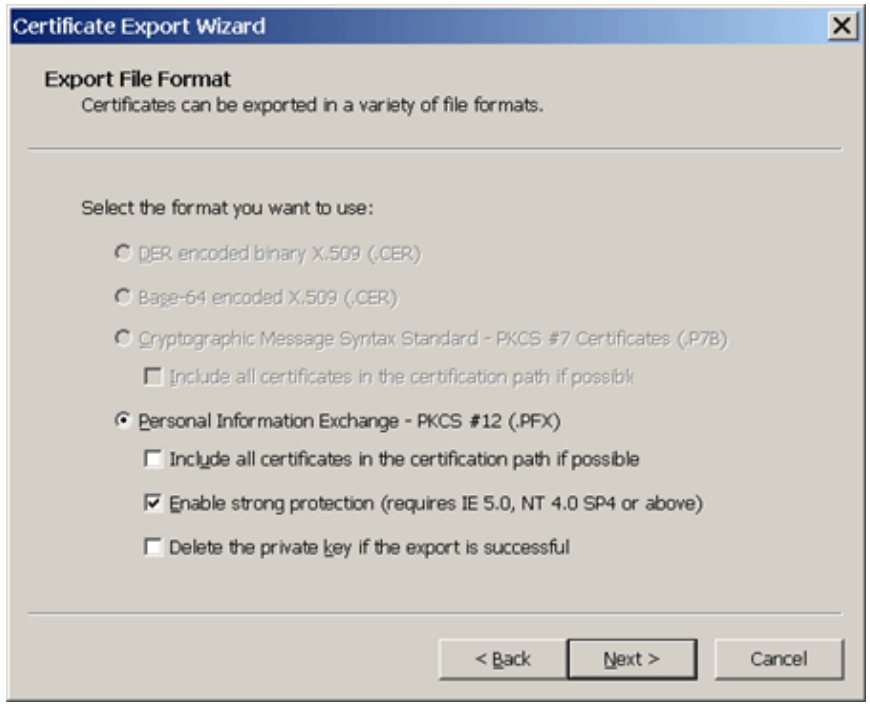

Figure 1. Certificate Export Wizard

All Windows operating systems define the extensions .pfx and .p12 as **Personal Information Exchange**, or PKCS #12, file types. The default association for these file types raises the certificate install dialog using the internal cryptographic extensions API:

```
rundll32.exe cryptext.dll,CryptExtAddPFX %1
```

Some of the most common reasons for exporting certificate keys to a PKCS #12 file format are:

- Backing up public/private RSA asymmetric key pairs in safe offsite storage
- Porting the RSA key pair to another computer for fixed installation, or temporary usage (e.g. code-signing)
- Removing sensitive EFS (Encryption File System) recovery keys for backup

The Platform SDK (PSDK) security glossary defines PKCS #12 as:

> The Personal Information Exchange Syntax Standard, developed and maintained by RSA Data Security, Inc. This syntax standard specifies a portable format for storing or transporting a user's private keys, certificates, and miscellaneous secrets.

The detailed specification for PKCS #12 is available from the RSA Security website and will not be discussed here. The Windows platform and CryptoAPI implements a subset of the full PKCS #12 specification.

This article provides the code framework for accessing the certificates, public, and private keys within pfx/p12 files from .NET Framework code using P/Invoke to CryptoAPI, and requires only the .NET Framework version 1.1 with no additional support. (From here on in, we will usually refer to any PKCS #12 file (pfx, p12 etc.) as PFX). The intent is to be able to access the protected PFX data, in a transient manner, from any computer, without permanently importing the keys into CryptoAPI-protected key storage. In this regard, the PFX file can be viewed as a kind of "soft" smart card, enabling RSA digital signature generation or decryption of enveloped content from any .NET Framework-enabled platform. Since PFX files are almost always quite small, typically less than 10 Kbytes, they are easily portable to any removable, recordable medium.

Using PFX with .NET

PFX files contain encrypted private keys, certificates, and potentially other secret information. To use PFX with .NET, we need to access the information in a form usable in .NET. This is conceptually similar to the approach used in the previous article, EncryptTo/DecryptTo: Encryption in .NET with CryptoAPI Certificate Stores, except here we derive the certificate and key credentials from a portable PFX file. To accomplish this, we will create a wrapper class, **JavaScience.PfxOpen**, which encapsulates:

- Opening a PFX file and verifying that the file is a valid PFX file
- Importing the PFX file into a temporary CryptoAPI memory store, and the associated private key(s) into a new key container under the default CSP
- Enumerating all certificates in the new memory store, and finding the first certificate with an associated private key
- Assigning fields for the CSP key container, provider name, type, and KeySpec
- Assigning fields for the public certificate, RSA public key modulus, exponent, and key size
- Providing a method to remove the key container, if required, after use

All the important fields required for cryptographic use from .NET are exposed as public **GET** properties. It is important to understand that we are not exposing the sensitive private key parameters directly in our class, but only acquiring the necessary properties, like key container names, that allow secure controlled access by the underlying CSP to the private keys. With access to the key container name holding the RSA private key, digital signature generation, or asymmetric decryption in .NET is possible with suitable initialization of an **RSACryptoServiceProvider** instance using the **container** property in the **PfxOpen** class:

```
CspParameters cp = new CspParameters();
cp.KeyContainerName = oPfxOpen.container;
RSACryptoServiceProvider RSA = new RSACryptoServiceProvider(cp);
```

Alternatively, we may only wish to use the imported public certificate properties from the PFX, for example, to RSA-encrypt to ourselves. In this case, we initialize an instance of **RSACryptoServiceProvider** using the **PfxOpen** class properties **keyexponent** and **keymodulus**:

```
RSAParameters RSAKeyInfo    = new RSAParameters();
RSAKeyInfo.Modulus    = oPfxOpen.keymodulus;
RSAKeyInfo.Exponent    = oPfxOpen.keyexponent;
RSACryptoServiceProvider oRSA    = new RSACryptoServiceProvider();
oRSA.ImportParameters(RSAKeyInfo);
```

PfxOpen overrides the **ToString()** method to display all the **PfxOpen** instance properties, which is a useful feature for debugging.

Additionally, since PFX files are almost always password-protected, we create a **JavaScience.PswdDialog** class, which is a Windows Form dialog designed to retrieve the password from the user:

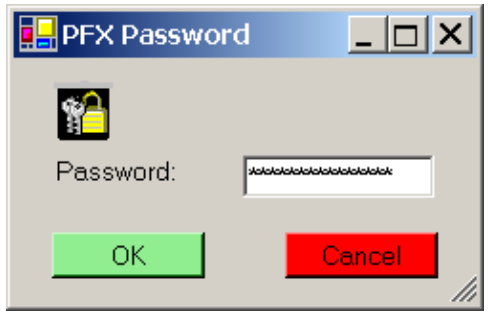

Figure 2. PFX Password dialog

Since strings are immutable objects in .NET managed code, the password data is only removed from memory by the common language runtime (CLR) garbage collection process. Unfortunately, this means that password strings are vulnerable. Thus, managed password dialogs such as that shown in Figure 2 should only be used on trusted, secured computers. For further information on protecting private data in .NET, see "Protecting Secret Data" in Writing Secure Code.

It is possible to **P/Invoke** to the **CredUI** capability, but this functionality is only available for Windows XP and later versions.

CryptoAPI Support for PKCS #12

CryptoAPI has rather modest support for handling PFX files, but it is sufficient to enable .NET digital signatures and development. The only CryptoAPI PFX functions that are of interest here are:

- **PFXImportCertStore()** to import a PFX file into a CryptoAPI transient memory store with the imported keys being managed by the default CSP
- **PFXIsPFXBlob()** to validate a buffer as a valid PFX structure

It should be noted that CAPICOM 2 has good support for importing certificates and keys from PFX files into CAPICOM Certificates and Stores. However, we will not be using CAPICOM 2 here, though similar capability is available via CAPICOM.

Code Overview

Our PFX utility application consists of four classes, all in the JavaScience namespace:

- **PfxOpen**: the main PFX decoding class discussed above
- **Win32**: encapsulates the P/Invoke declarations for CryptoAPI functions and structures
- **PswdDialog**: a Windows Form utility to retrieve user-supplied passwords
- **TextPFX**: a test class to exercise PfxOpen

With the C# source code files and the password dialog box icon file in the same directory, the application can be compiled manually using the .NET Framework SDK 1.1 compiler (or via Visual Studio .NET) with:

```
csc.exe /m:JavaScience.TestPfx /res:keylock.ico,JavaScience.keylock.ico TestPfx.cs PFxOpen.cs PswdDialog.cs Win32.cs
```

This code creates a single file console application assembly, TestPfx.exe, that contains all classes. Note that we specify the **/m** (main entry point class) switch, since the class PswdDialog also contains a main entry point that is retained for stand-alone test purposes. Also, an icon resource, keylock.ico, is embedded into the assembly as a manifest resource. The application is executed from a command prompt:

```
TestPfx  [pfxfilename]  [pswd]
```

If only a PFX filename is supplied, the user is prompted for a password via a PswdDialog. The TestPfx class:

- Validates the supplied PFX file
- Prompts the user for a password, if necessary
- Attempts to open the PFX file and instantiates a PfxOpen instance
- If successful, displays all PfxOpen properties using ToString()
- Pauses for 2 seconds, representing some useful cryptographic usage of the PFX data
- Attempts to remove the generated persistent key container and keys therein

In an extended implementation of the sample here, the public properties of the PfxOpen instance would be used to instantiate .NET asymmetric providers, as mentioned above, to actually use the imported keys for signing or encryption. A typical sample output for TestPfx shows the information acquired from the PFX file.

Code Details

Only code details not discussed in my previous article will be discussed here.

The instance method **PfxOpen.LoadPfx(String pfxfilename, ref String pswd)** loads the PFX binary file into **byte[] pfxdata**, which is then used to initialize a **CRYPT_DATA_BLOB PFX** blob struct, and memory is allocated:

```
[StructLayout(LayoutKind.Sequential)]
 public struct CRYPT_DATA_BLOB
 {
  public int cbData;
  public IntPtr pbData;
 }
...

 CRYPT_DATA_BLOB ppfx = new CRYPT_DATA_BLOB();
 ppfx.cbData = pfxdata.Length;
 ppfx.pbData = Marshal.AllocHGlobal(pfxdata.Length);
 Marshal.Copy(pfxdata, 0, ppfx.pbData, pfxdata.Length);
```

After using **Win32.PFXIsPFXBlob()** to verify that we have a valid PFX blob, we attempt to import the PFX certificates into a transient CryptoAPI memory store, with the associated private keys placed in a newly created key container using **Win32.PFXImportCertStore()**:

```
[DllImport("crypt32.dll", SetLastError=true)]
 public static extern IntPtr PFXImportCertStore(
  ref CRYPT_DATA_BLOB pPfx,
  [MarshalAs(UnmanagedType.LPWStr)] String szPassword,
  uint dwFlags);
...
hMemStore = Win32.PFXImportCertStore(ref ppfx, pswd, CRYPT_USER_KEYSET) ;
```

```
 pswd = null;
  if(hMemStore == IntPtr.Zero){
   string errormessage = new Win32Exception(Marshal.GetLastWin32Error()).Message;
   Console.WriteLine("\n{0}", errormessage);
   Marshal.FreeHGlobal(ppfx.pbData);
   return result;
  }
 Marshal.FreeHGlobal(ppfx.pbData);
```

We define CRYPT_DATA_BLOB as a managed structure that must be passed by reference to match the CryptoAPI function argument pointer **CRYPT_DATA_BLOB* pPFX**.

If **PFXImportCertStore** succeeds, we have an **IntPtr** handle to the memory store containing the imported certificates. We now release unmanaged memory assigned for the CRYPT_DATA_BLOB structure member. At this stage, persistent key containers have been generated. Note that while PKCS #12 supports containing more than one private key, but we will assume that there is only a single private key, with matching certificate in the PFX.

Next, we enumerate the certificates in the memory store using **Win32.CertEnumCertificatesInStore()**, checking each certificate to see if it has an associated private key using **Win32.CertGetCertificateContextProperty()** with **dwPropID = CERT_KEY_PROV_INFO_PROP_ID**. The first certificate with a matching private key is used to return an IntPtr representing a handle to a **CRYPT_KEY_PROV_INFO** structure. This IntPtr is marshaled to return the managed CRYPT_KEY_PROV_INFO structure. The structure members are parameters needed for private key access, and are assigned to the instance fields of PfxOpen:

```
 [StructLayout(LayoutKind.Sequential)]
  public struct CRYPT_KEY_PROV_INFO
  {
   [MarshalAs(UnmanagedType.LPWStr)] public String ContainerName;
   [MarshalAs(UnmanagedType.LPWStr)] public String ProvName;
   public uint ProvType;
   public uint Flags;
   public uint ProvParam;
   public IntPtr rgProvParam;
   public uint KeySpec;
  }
 ....

     if(Win32.CertGetCertificateContextProperty(hCertCntxt, CERT_KEY_PROV_INFO_PROP_ID, pProvInfo, ref provinfosize))
   {
    CRYPT_KEY_PROV_INFO ckinfo = (CRYPT_KEY_PROV_INFO)Marshal.PtrToStructure(pProvInfo, typeof(CRYPT_KEY_PROV_INFO));
    Marshal.FreeHGlobal(pProvInfo);

    this.pfxcontainer= ckinfo.ContainerName;
    this.pfxprovname = ckinfo.ProvName;
    this.pfxprovtype = ckinfo.ProvType;
    this.pfxkeyspec  = ckinfo.KeySpec;
    this.pfxcert     = new X509Certificate(hCertCntxt);
    if(!this.GetCertPublicKey(pfxcert))
      Console.WriteLine("Couldn't get certificate public key");
    result = true;
    break;
       }
```

The retrieved certificate handle **hCertCntxt** is also used to instantiate a .NET X509Certificate object using the overloaded constructor **pfxcert = new X509Certificate(hCertCntxt)**. The certificate is passed to **PfxOpen.GetCertPublicKey()** and the public key data is decoded from the certificate using **Win32.CryptDecodeObject()**, parsed and the public key modulus, exponent and key size are assigned to instance fields of PfxOpen. Finally, we must release all opened handles to unmanaged objects:

```
//-------  Clean Up  -----------
  if(pProvInfo != IntPtr.Zero)
   Marshal.FreeHGlobal(pProvInfo);
  if(hCertCntxt != IntPtr.Zero)
   Win32.CertFreeCertificateContext(hCertCntxt);
  if(hMemStore != IntPtr.Zero)
   Win32.CertCloseStore(hMemStore, 0) ;
  return result;
```

Removing Key Containers

The CryptoAPI function **PFXImportCertStore()** creates a temporary memory store, and also new default CSP key container(s). However, the key containers are persistent, even after the memory store is removed. Therefore, the key container(s) must be manually deleted after use, if key persistence is not required. PfxOpen provides the **DeleteKeyContainer()** function to facilitate removal of key containers and associated keys. This is implemented using CryptoAPI **CryptAcquireContext()** with the **CRYPT_DELETEKEYSET** flag:

```
 internal bool DeleteKeyContainer(String containername, String provname, uint provtype)
 {
  const uint CRYPT_DELETEKEYSET   = 0x00000010;
  IntPtr hCryptProv = IntPtr.Zero;
  if(containername == null || provname == null || provtype == 0)
   return false;
  if (Win32.CryptAcquireContext(ref hCryptProv, containername, provname, provtype, CRYPT_DELETEKEYSET)){
```

```
    return true;
   }
  else
   {
    PfxOpen.showWin32Error(Marshal.GetLastWin32Error());
    return false;
   }
 }
```

Note that CryptoAPI key containers can also be removed from purely managed code:

```
internal bool NetDeleteKeyContainer(String containername, String provname, int provtype){
   CspParameters cp = new CspParameters();
   cp.KeyContainerName = containername;
   cp.ProviderName = provname;
   cp.ProviderType = provtype;
   RSACryptoServiceProvider oRSA = new RSACryptoServiceProvider(cp);
   oRSA.PersistKeyInCsp = false;
   oRSA.Clear();
   return true;
 }
```

The CAPICOM handling of PFX files imported into memory stores is slightly different with regard to the key containers. According to the **CAPICOM.Store.Load** documentation:

```
If the Load method is called on a memory store, any key containers that are
created will be deleted when the memory store is deleted.
```

Thus, CAPICOM automatically manages the deletion of the key containers associated with PFX loaded to memory stores, unlike CryptoAPI **PFXImportCertStore()**.

Conclusion

This article has provided the code framework required to use PKCS #12 pfx/p12 files from .NET Framework code using P/Invoke. Being able to port and use your private RSA keys provides some very interesting and cool possibilities. It is not difficult to merge the code presented here, with the enveloped encryption/decryption classes presented in my previous article.

> **Note** Using keys imported into memory stores from pfx files as discussed here exposes the private key material, making it vulnerable to memory monitors.

References

- PKCS #12 - Personal Information Exchange Syntax Standard
- Platform SDK: Security: PFXImportCertStore()
- Platform SDK: Security: CAPICOM.Store.Load()
- Extending .NET Cryptography with CAPICOM and P/Invoke
- EncryptTo/DecryptTo: Encryption in .NET with CryptoAPI Certificate Stores
- HOW TO: Export Certificates in Windows 2000

About the Author

Michel I. Gallant, Ph.D., has over 20 years experience in the telecommunications industry. He has worked as a senior photonic designer, and as a security analyst and architect in a major Canadian telecommunications corporation. He has extensive experience in code-signing and applied cryptography. He was awarded an MVP in Security for 2003. Michel lives in Ottawa, Canada and enjoys playing surf music, designing puzzles, and designing innovative electronic projects.

Importing and exporting certificates

The Certificates snap-in provides administrative tools to export and import certificates, including their certification paths and private keys, if needed. You can export certificates to or import certificates from PKCS #12 files, PKCS #7 files, and binary-encoded X.509 certificate files.

Importing a certificate

You might want to import a certificate:

- To install a certificate that was sent to you in a file by another user, computer, or certification authority
- To restore a damaged or lost certificate that you previously backed up.
- To install a certificate and its associated private key from a computer that the certificate holder was previously using.

When you import a certificate, you copy the certificate from a file that uses a standard certificate storage format to a certificate store for your user account or your computer account.

Exporting a certificate

You might want to export a certificate:

- To back up a certificate.
- To back up a certificate and its associated private key.
- To copy a certificate for use on another computer.
- To remove a certificate and its private key from the certificate holder's current computer for installation on another computer.

When you export a certificate, you are copying the certificate from its certificate store to a file that uses a standard certificate storage format.

Standard certificate file formats

You can import and export certificates in the following formats:

- **Personal Information Exchange (PKCS #12)**

  The Personal Information Exchange format (PFX, also called PKCS #12) enables the transfer of certificates and their corresponding private keys from one computer to another or from a computer to removable media.

  PKCS #12 is an industry format suitable for transport or backup and restoration of a certificate and its associated private key. This can be between products from the same vendor or different vendors.

  To use the PKCS #12 format, the cryptographic service provider (CSP) must recognize the certificate and keys as exportable. If a certificate was issued from a Windows 2000 certification authority, the private key for that certificate is only exportable if one of the following is true:

  - The certificate is for EFS (encrypting file system) or EFS recovery.
  - The certificate was requested through the Advanced Certificate Request certification authority Web page with the **Mark keys as exportable** check box selected.

Because exporting a private key might expose it to unintended parties, the PKCS #12 format is the only format supported in Windows 2000 for exporting a certificate and its associated private key.

- **Cryptographic Message Syntax Standard (PKCS #7)**

  The PKCS #7 format enables the transfer of a certificate and all the certificates in its certification path from one computer to another, or from a computer to removable media. PKCS #7 files typically use the .p7b extension.

- **DER Encoded Binary X.509**

  This format might be used by certification authorities that are not on Windows 2000 servers, so it is supported for interoperability. DER certificate files use the .cer extension.

- **Base64 Encoded X.509**

  This format might be used by certification authorities that are not on Windows 2000 servers, so it is supported for interoperability. Base64 certificate files use the .cer extension.

See also:

- To import a certificate
- To export a certificate
- To export a certificate with the private key
- To view the certificates in a PKCS #7 file

Send Us Your Feedback

Importing and exporting certificates

The Certificates snap-in provides administrative tools to export and import certificates, including their certification paths and private keys, if needed. You can export certificates to or import certificates from PKCS #12 files, PKCS #7 files, and binary-encoded X.509 certificate files.

Importing a certificate

You might want to import a certificate:

- To install a certificate that was sent to you in a file by another user, computer, or certification authority
- To restore a damaged or lost certificate that you previously backed up.
- To install a certificate and its associated private key from a computer that the certificate holder was previously using.

When you import a certificate, you copy the certificate from a file that uses a standard certificate storage format to a certificate store for your user account or your computer account.

Top of page

Exporting a certificate

You might want to export a certificate:

- To back up a certificate.
- To back up a certificate and its associated private key.
- To copy a certificate for use on another computer.
- To remove a certificate and its private key from the certificate holder's current computer for installation on another computer.

When you export a certificate, you are copying the certificate from its certificate store to a file that uses a standard certificate storage format.

Top of page

Standard certificate file formats

You can import and export certificates in the following formats:

- **Personal Information Exchange (PKCS #12)**

The Personal Information Exchange format (PFX, also called PKCS #12) enables the transfer of certificates and their corresponding private keys from one computer to another or from a computer to removable media.

PKCS #12 (Public Key Cryptography Standard #12) is an industry format that is suitable for transport or backup and restoration of a certificate and its associated private key. This can be between products from the same vendor or different vendors.

To use the PKCS #12 format, the cryptographic service provider (CSP) must recognize the certificate and keys as exportable. If a certificate was issued from a Windows 2000 certification authority, the private key for that certificate is only exportable if one of the following is true:

- The certificate is for EFS (encrypting file system) or EFS recovery.
- The certificate was requested through the Advanced Certificate Request certification authority Web page with the **Mark keys as exportable** check box selected.

Because exporting a private key might expose it to unintended parties, the PKCS #12 format is the only format supported in Windows XP for exporting a certificate and its associated private key.

- **Cryptographic Message Syntax Standard (PKCS #7)**

The PKCS #7 format enables the transfer of a certificate and all the certificates in its certification path from one computer to another, or from a computer to removable media. PKCS #7 files typically use the .p7b extension and are compatible with the ITU-T X.509 standard.

PKCS #7 allows for attributes such as countersignatures to be associated with signatures. Attributes such as signing time can be authenticated along with message content.

For more information on PKCS #7, see the PKCS #7 page at the RSA Labs Web site.

- **DER Encoded Binary X.509**

DER (Distinguished Encoding Rules) for ASN.1, as defined in ITU-T Recommendation X.509, is a more restrictive encoding standard than the alternative BER (Basic Encoding Rules) for ASN.1, as defined in ITU-T Recommendation X.209, upon which DER is based. Both BER and DER provide a platform-independent method of encoding objects (such as certificates and messages) for transmission between devices and applications.

During certificate encoding, most applications use DER because a portion of the certificate (the Certification Request's Certification Request Info) must be DER-encoded to be signed.

This format might be used by certification authorities that are not on Windows 2000 servers, so it is supported for interoperability. DER certificate files use the .cer extension.

For more information, see the document "ITU-T Recommendation X.509, Information Technology -- Open Systems Interconnection -- The Directory: Authentication Framework," at the International Telecommunication Union (ITU) Web site.

- **Base64 Encoded X.509**

This is an encoding method developed for use with Secure/Multipurpose Internet Mail Extensions (S/MIME) which is a popular, standard method for transferring binary attachments over the Internet. Base64 encodes files into ASCII text format, making corruption less likely as the files are sent through Internet gateways, while S/MIME provides some cryptographic security services for electronic messaging applications, including non-repudiation of origin using digital signatures, privacy and data security using encryption, authentication, and message integrity.

The MIME (Multipurpose Internet Mail Extensions) specification (RFC 1341 and successors) defines a mechanism for encoding arbitrary binary information for transmission by electronic mail.

Because all MIME-compliant clients can decode Base64 files, this format might be used by certification authorities that are not on Windows 2000 servers, so it is supported for interoperability. Base64 certificate files use the .cer extension.

For more information, see "RFC 2633 S/MIME Version 3 Message Specification, 1999," at the Internet Engineering Task Force (IETF) Web site and "A Layman's Guide to a Subset of ASN.1, BER and DER," an RSA Laboratories technical note.

Top of page

Choosing an export format

If you are exporting certificates for import onto a computer running Windows, PKCS #7 format is the preferred export format, primarily because this format preserves the chain of certificate

authorities, or the certification path, of any certificate that includes countersignatures associated with signatures.

If you are exporting certificates for import onto a computer running another operating system, it is possible that the PKCS #7 format is supported. If it is not supported, the DER Encoded Binary format or the Base64 Encoded format are provided for interoperability.

For more information, see To import a certificate, To export a certificate, To export a certificate with the private key, and To view the certificates in a PKCS #7 file

Web addresses can change, so you might be unable to connect to the Web site or sites mentioned here.

Top of page

Importing and exporting certificates

Updated: January 21, 2005

Importing and exporting certificates

The Certificates snap-in provides administrative tools to export and import certificates, including their certification paths and private keys, if needed. You can export certificates to or import certificates from PKCS #12 files, PKCS #7 files, and binary-encoded X.509 certificate files.

Top of page

Importing a certificate

You might want to import a certificate:

- To install a certificate that was sent to you in a file by another user, computer, or certification authority.
- To restore a damaged or lost certificate that you previously backed up.
- To install a certificate and its associated private key from a computer that the certificate holder was previously using.

When you import a certificate, you copy the certificate from a file that uses a standard certificate storage format to a certificate store for your user account or your computer account.

Top of page

Exporting a certificate

You might want to export a certificate:

- To back up a certificate.
- To back up a certificate and its associated private key.
- To copy a certificate for use on another computer.

- To remove a certificate and its private key from the certificate holder's current computer for installation on another computer.

When you export a certificate, you are copying the certificate from its certificate store to a file that uses a standard certificate storage format.

Top of page

Standard certificate file formats

You can import and export certificates in the following formats:

- **Personal Information Exchange (PKCS #12)**

The Personal Information Exchange format (PFX, also called PKCS #12) enables the transfer of certificates and their corresponding private keys from one computer to another or from a computer to removable media.

PKCS #12 (Public Key Cryptography Standard #12) is an industry format that is suitable for transport or backup and restoration of a certificate and its associated private key. This can be between products from the same vendor or different vendors.

To use the PKCS #12 format, the cryptographic service provider (CSP) must recognize the certificate and keys as exportable. If a certificate was issued from a Windows Server 2003 or Windows 2000 certification authority, the private key for that certificate is only exportable if one of the following is true:

- The certificate is for EFS (Encrypting File System) or EFS recovery.
- The certificate was requested through the Advanced Certificate Request certification authority Web page with the **Mark keys as exportable** check box selected.

Because exporting a private key might expose it to unintended parties, the PKCS #12 format is the only format supported in the Windows Server 2003 family for exporting a certificate and its associated private key.

- **Cryptographic Message Syntax Standard (PKCS #7)**

The PKCS #7 format enables the transfer of a certificate and all the certificates in its certification path from one computer to another, or from a computer to removable media. PKCS #7 files typically use the .p7b extension, and are compatible with the ITU-T X.509 standard. PKCS #7 allows for attributes, such as countersignatures, to be associated with signatures, and attributes such as signing time can be authenticated along with message content. For more information on PKCS #7, see the PKCS #7 page at the RSA Labs Web site.

- **DER Encoded Binary X.509**

DER (Distinguished Encoding Rules) for ASN.1, as defined in ITU-T Recommendation X.509, is a more restrictive encoding standard than the alternative BER (Basic Encoding Rules) for ASN.1, as defined in ITU-T Recommendation X.209, upon which DER is based. Both BER and DER provide a platform-independent method of encoding objects such as certificates and messages for transmission between devices and applications.

During certificate encoding, most applications use DER because a portion of the certificate (the CertificationRequest's CertificationRequestInfo) must be DER-encoded to be signed.

This format might be used by certification authorities that are not on computers running Windows Server 2003 , so it is supported for interoperability. DER certificate files use the .cer extension.

For more information, see the document "ITU-T Recommendation X.509, Information Technology--Open Systems Interconnection--The Directory: Authentication Framework," at the International Telecommunication Union (ITU) Web site.

- **Base64 Encoded X.509**

This is an encoding method developed for use with Secure/Multipurpose Internet Mail Extensions (S/MIME), which is a popular, standard method for transferring binary attachments over the Internet. Base64 encodes files into ASCII text format, making corruption less likely as the files are sent through Internet gateways, while S/MIME provides some cryptographic security services for electronic messaging applications, including non-repudiation of origin using digital signatures, privacy and data security using encryption, authentication, and message integrity.

The MIME (Multipurpose Internet Mail Extensions) specification (RFC 1341 and successors) defines a mechanism for encoding arbitrary binary information for transmission by electronic mail.

Because all MIME-compliant clients can decode Base64 files, this format might be used by certification authorities that are not on computers running Windows Server 2003 , so it is supported for interoperability. Base64 certificate files use the .cer extension.

For more information, see the document "RFC 2633 S/MIME Version 3 Message Specification, 1999," at the Internet Engineering Task Force (IETF) Web site.

Top of page

Choosing an export format

If you are exporting certificates to be imported onto a computer running Windows, PKCS #7 format is the preferred export format, primarily because this format preserves the chain of

certification authorities, or the certification path, of any certificate that includes countersignatures associated with signatures.

If you are exporting certificates for import onto a computer running another operating system, it is possible that the PKCS #7 format is supported. If it is not supported, the DER Encoded Binary format or the Base64 Encoded format are provided for interoperability.

For more information, see Import a certificate, Export a certificate, Export a certificate with the private key, and View the certificates in a PKCS #7 file.

Note

- Web addresses can change, so you might be unable to connect to the Web site or sites mentioned here.

Top of page

Platform SDK: Cryptography

Backing Up and Restoring the Certificate Services Private Key

You cannot use the Certadm.dll's backup and restore functions to back up the Certificate Services *private key(s)*. Private keys cannot be backed up by these functions because these functions are intended to backup and restore the Certificate Services database (and related files), and this database does not contain any private keys (even for self-issued certificates).

To back up a Certificate Services private key, use the Certification Authority MMC snap-in, or the certutil command (with -backup or -backupkey specified). Backing up the private key with the Certification Authority MMC snap-in or certutil results in the private key being written to PKCS #12 file. Even though this PKCS #12 file is password-protected, it should be considered extremely sensitive and must be stored securely; the password to the PKCS #12 file should also be guarded from unauthorized persons.

Similarly, private keys cannot be restored by the Certificate Services backup and restore functions. A Certificate Services backup key contained in a PKCS #12 file can be restored by the Certification Authority MMC snap-in, or by the certutil command (specifying the -restore or -restorekey verbs); note that the person performing the restore operation will need to know the password for the PKCS #12 file.

There are only two cases in which a Certificate Services private key must be backed up. The first case is after the installation of Certificate Services. The second case is after any renewal operation of the Certificate Services certificate.

Trusted root certification authority policy

To establish a trusted root certification authority (CA) using Group Policy, the Group Policy object that you create must have access to the root certificate This requires that you import a copy of the root authority certificate.

You can do this using the procedure Add a trusted root certification authority to a Group Policy object

To import a root CA certificate, you will need the root certificate in a PKCS #12 file, PKCS #7 file, or binary-encoded X.509 files. See Importing and exporting certificates for more information about these file formats.

See also:

- Policies to establish trust of root certification authorities
- Delete a trusted root certification authority from a Group Policy object

---

Send Us Your Feedback

Object IDs Associated with Microsoft Cryptography

View products that this article applies to.

Article ID : 287547
Last Review : December 18, 2003
Revision : 3.0

This article was previously published under Q287547

SUMMARY

This article describes the Object ID numbers (OIDs) that are defined for Microsoft.

MORE INFORMATION

In this scope, OIDs are numerical values that enable programs to determine if a certificate is valid for a particular use. The OIDs can be used to represent components like X509 extensions, PKCS #7 extensions and PKCS #7 contents. Each subtree in the Microsoft OID is assigned to a specific area.

```
Microsoft OID.....................................1.3.6.1.4.1.311

Authenticode.....................................1.3.6.1.4.1.311.2
     Software Publishing (with associated encoders/decoders)
        SPC_INDIRECT_DATA_OBJID                  1.3.6.1.4.1.311.2.1.4
        SPC_STATEMENT_TYPE_OBJID                 1.3.6.1.4.1.311.2.1.11
        SPC_SP_OPUS_INFO_OBJID                   1.3.6.1.4.1.311.2.1.12
        SPC_PE_IMAGE_DATA_OBJID                  1.3.6.1.4.1.311.2.1.15
        SPC_SP_AGENCY_INFO_OBJID                 1.3.6.1.4.1.311.2.1.10
        SPC_MINIMAL_CRITERIA_OBJID               1.3.6.1.4.1.311.2.1.26
        SPC_FINANCIAL_CRITERIA_OBJID             1.3.6.1.4.1.311.2.1.27
        SPC_LINK_OBJID                           1.3.6.1.4.1.311.2.1.28
        SPC_HASH_INFO_OBJID                      1.3.6.1.4.1.311.2.1.29
        SPC_SIPINFO_OBJID                        1.3.6.1.4.1.311.2.1.30

     Software Publishing (with NO associated encoders/decoders)
        SPC_CERT_EXTENSIONS_OBJID                1.3.6.1.4.1.311.2.1.14
        SPC_RAW_FILE_DATA_OBJID                  1.3.6.1.4.1.311.2.1.18
```

```
        SPC_STRUCTURED_STORAGE_DATA_OBJID       1.3.6.1.4.1.311.2.1.19
        SPC_JAVA_CLASS_DATA_OBJID               1.3.6.1.4.1.311.2.1.20
        SPC_INDIVIDUAL_SP_KEY_PURPOSE_OBJID     1.3.6.1.4.1.311.2.1.21
        SPC_COMMERCIAL_SP_KEY_PURPOSE_OBJID     1.3.6.1.4.1.311.2.1.22
        SPC_CAB_DATA_OBJID                      1.3.6.1.4.1.311.2.1.25
        SPC_GLUE_RDN_OBJID                      1.3.6.1.4.1.311.2.1.25

     CTL for Software Publishers Trusted CAs    1.3.6.1.4.1.311.2.2
     (sub-subtree is defined for Software Publishing trusted CAs)
        szOID_TRUSTED_CODESIGNING_CA_LIST       1.3.6.1.4.1.311.2.2.1
        szOID_TRUSTED_CLIENT_AUTH_CA_LIST       1.3.6.1.4.1.311.2.2.2
        szOID_TRUSTED_SERVER_AUTH_CA_LIST       1.3.6.1.4.1.311.2.2.3

Time Stamping.................................1.3.6.1.4.1.311.3
(with Associated encoder/decoders)
        SPC_TIME_STAMP_REQUEST_OBJID            1.3.6.1.4.1.311.3.2.1

Permissions...................................1.3.6.1.4.1.311.4

Crypto 2.0....................................1.3.6.1.4.1.311.10
     PKCS #7 ContentType Object Identifier for Certificate Trust List (CTL)
        szOID_CTL                               1.3.6.1.4.1.311.10.1
     Sorted CTL Extension
        szOID_SORTED_CTL                        1.3.6.1.4.1.311.10.1.1

     Next Update Location extension or attribute. Value is an encoded GeneralNames
        szOID_NEXT_UPDATE_LOCATION              1.3.6.1.4.1.311.10.2

     Enhanced Key Usage (Purpose)
        Signer of CTLs
        szOID_KP_CTL_USAGE_SIGNING              1.3.6.1.4.1.311.10.3.1

        Signer of TimeStamps
        szOID_KP_TIME_STAMP_SIGNING             1.3.6.1.4.1.311.10.3.2

     Can use strong encryption in export environment
        szOID_SERVER_GATED_CRYPTO               1.3.6.1.4.1.311.10.3.3
        szOID_SERIALIZED                        1.3.6.1.4.1.311.10.3.3.1

     Can use encrypted file systems (EFS)
        szOID_EFS_CRYPTO                        1.3.6.1.4.1.311.10.3.4
        szOID_EFS_RECOVERY                      1.3.6.1.4.1.311.10.3.4.1

     Can use Windows Hardware Compatible (WHQL)
        szOID_WHQL_CRYPTO                       1.3.6.1.4.1.311.10.3.5

     Signed by the NT5 build lab
        szOID_NT5_CRYPTO                        1.3.6.1.4.1.311.10.3.6

     Signed by and OEM of WHQL
```

```
   szOID_OEM_WHQL_CRYPTO                    1.3.6.1.4.1.311.10.3.7

Signed by the Embedded NT
   szOID_EMBEDDED_NT_CRYPTO                 1.3.6.1.4.1.311.10.3.8

Signer of a CTL containing trusted roots
   szOID_ROOT_LIST_SIGNER                   1.3.6.1.4.1.311.10.3.9

Can sign cross-cert and subordinate CA requests with qualified
subordination (name constraints, policy mapping, etc.)
   szOID_KP_QUALIFIED_SUBORDINATION         1.3.6.1.4.1.311.10.3.10

Can be used to encrypt/recover escrowed keys
   szOID_KP_KEY_RECOVERY                    1.3.6.1.4.1.311.10.3.11

Signer of documents
   szOID_KP_DOCUMENT_SIGNING                1.3.6.1.4.1.311.10.3.12

Microsoft Attribute Object Identifiers
   szOID_YESNO_TRUST_ATTR                   1.3.6.1.4.1.311.10.4.1

Microsoft Music
   szOID_DRM                                1.3.6.1.4.1.311.10.5.1

Microsoft DRM EKU
   szOID_DRM_INDIVIDUALIZATION              1.3.6.1.4.1.311.10.5.2

Microsoft Licenses
   szOID_LICENSES                           1.3.6.1.4.1.311.10.6.1
   szOID_LICENSE_SERVER                     1.3.6.1.4.1.311.10.6.2

Microsoft CERT_RDN attribute Object Identifiers
   szOID_MICROSOFT_RDN_PREFIX               1.3.6.1.4.1.311.10.7
Special RDN containing the KEY_ID. Its value type is CERT_RDN_OCTET_STRING.
   szOID_KEYID_RDN                          1.3.6.1.4.1.311.10.7.1

Microsoft extension in a CTL to add or remove the certificates. The
extension type is an INTEGER. 0 => add certificate, 1 => remove certificate
   szOID_REMOVE_CERTIFICATE                 1.3.6.1.4.1.311.10.8.1

Microsoft certificate extension containing cross certificate distribution
points. ASN.1 encoded as follows:
    CrossCertDistPoints ::= SEQUENCE {
        syncDeltaTime               INTEGER (0..4294967295) OPTIONAL,
        crossCertDistPointNames     CrossCertDistPointNames
    } --#public--
         CrossCertDistPointNames ::= SEQUENCE OF GeneralNames

   szOID_CROSS_CERT_DIST_POINTS             1.3.6.1.4.1.311.10.9.1
```

```
     Microsoft CMC OIDs                          1.3.6.1.4.1.311.10.10

     Similar to szOID_CMC_ADD_EXTENSIONS. Attributes replaces Extensions.
        szOID_CMC_ADD_ATTRIBUTES                 1.3.6.1.4.1.311.10.10.1

     Microsoft certificate property OIDs         1.3.6.1.4.1.311.10.11
     The OID component following the prefix contains the PROP_ID (decimal)
        szOID_CERT_PROP_ID_PREFIX                1.3.6.1.4.1.311.10.11.

     CryptUI                                     1.3.6.1.4.1.311.10.12
        szOID_ANY_APPLICATION_POLICY             1.3.6.1.4.1.311.10.12.1

Catalog.....................................1.3.6.1.4.1.311.12
        szOID_CATALOG_LIST                       1.3.6.1.4.1.311.12.1.1
        szOID_CATALOG_LIST_MEMBER                1.3.6.1.4.1.311.12.1.2
        CAT_NAMEVALUE_OBJID                      1.3.6.1.4.1.311.12.2.1
        CAT_MEMBERINFO_OBJID                     1.3.6.1.4.1.311.12.2.2

Microsoft PKCS10 OIDs.......................1.3.6.1.4.1.311.13
        szOID_RENEWAL_CERTIFICATE                1.3.6.1.4.1.311.13.1
        szOID_ENROLLMENT_NAME_VALUE_PAIR         1.3.6.1.4.1.311.13.2.1
        szOID_ENROLLMENT_CSP_PROVIDER            1.3.6.1.4.1.311.13.2.2

Microsoft Java..............................1.3.6.1.4.1.311.15

Microsoft Outlook/Exchange..................1.3.6.1.4.1.311.16
          Outlook Express                        1.3.6.1.4.1.311.16.4
          Used by OL/OLEXP to identify which certificate signed the PKCS # 7 message

Microsoft PKCS12 attributes.................1.3.6.1.4.1.311.17
        szOID_LOCAL_MACHINE_KEYSET               1.3.6.1.4.1.311.17.1

Microsoft Hydra.............................1.3.6.1.4.1.311.18

Microsoft ISPU Test.........................1.3.6.1.4.1.311.19

Microsoft Enrollment Infrastructure.............1.3.6.1.4.1.311.20
        szOID_AUTO_ENROLL_CTL_USAGE              1.3.6.1.4.1.311.20.1
     Extension contain certificate type
        szOID_ENROLL_CERTTYPE_EXTENSION          1.3.6.1.4.1.311.20.2
        szOID_ENROLLMENT_AGENT                   1.3.6.1.4.1.311.20.2.1
        szOID_KP_SMARTCARD_LOGON                 1.3.6.1.4.1.311.20.2.2
        szOID_NT_PRINCIPAL_NAME                  1.3.6.1.4.1.311.20.2.3
        szOID_CERT_MANIFOLD                      1.3.6.1.4.1.311.20.3

Microsoft CertSrv Infrastructure.............1.3.6.1.4.1.311.21
     CertSrv (with associated encoders/decoders)
        szOID_CERTSRV_CA_VERSION                 1.3.6.1.4.1.311.21.1
```

```
Microsoft Directory Service.....................1.3.6.1.4.1.311.25
        szOID_NTDS_REPLICATION                  1.3.6.1.4.1.311.25.1

IIS.............................................1.3.6.1.4.1.311.30

Windows updates and service packs...............1.3.6.1.4.1.311.31
        szOID_PRODUCT_UPDATE                    1.3.6.1.4.1.311.31.1

Fonts...........................................1.3.6.1.4.1.311.40

Microsoft Licensing and Registration............1.3.6.1.4.1.311.41

Microsoft Corporate PKI (ITG)...................1.3.6.1.4.1.311.42

CAPICOM.........................................1.3.6.1.4.1.311.88
        szOID_CAPICOM                           1.3.6.1.4.1.311.88      Reserved for CAPICOM.
        szOID_CAPICOM_VERSION                   1.3.6.1.4.1.311.88.1    CAPICOM version
        szOID_CAPICOM_ATTRIBUTE                 1.3.6.1.4.1.311.88.2    CAPICOM attribute
        szOID_CAPICOM_DOCUMENT_NAME             1.3.6.1.4.1.311.88.2.1  Document type attribute
        szOID_CAPICOM_DOCUMENT_DESCRIPTION      1.3.6.1.4.1.311.88.2.2  Document description attribute
        szOID_CAPICOM_ENCRYPTED_DATA            1.3.6.1.4.1.311.88.3    CAPICOM encrypted data message.
        szOID_CAPICOM_ENCRYPTED_CONTENT         1.3.6.1.4.1.311.88.3.1  CAPICOM content of encrypted data.
```

APPLIES TO

- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Professional Edition
- Microsoft Windows 2000 Datacenter Server

Back to the top

Keywords: kbinfo KB287547

Certutil tasks for backing up and restoring certificates

Updated: January 21, 2005

Certutil tasks for backing up and restoring certificates

Related Links

- Command-line reference A-Z
- Command shell overview

Certification authorities should be backed up regularly and restored when necessary to provide their services. You can use **certutil** to perform these tasks.

To view the syntax for a specific task, click a task:

- To back up Certificate Services
- To back up a CA database
- To back up the CA certificate and keys
- To restore the CA database, certificates, and keys
- To restore the CA database
- To restore the CA certificate and keys from a backup directory or a PKCS #12 (.pfx) file
- To dump the CA database schema, for example, column names, types, and max sizes

To back up Certificate Services

Syntax

certutil -backup [**-f**] [**-gmt**] [**-seconds**] [**-v**] [**-config** *CAMachineName\CAName*] [**-p** *Password*] *BackupDirectory*[**incremental**] [**keeplog**]

Parameters

-backup

Backs up Certificate Services.

-f

Overwrites existing files or keys.

-gmt

Displays time as Greenwich mean time.

-seconds

Displays time with seconds and milliseconds.

-v

Specifies verbose output.

-config *CAMachineName\CAName*

processes the operation by using the CA specified in the configuration string (that is, *CAMachineName\CAName*).

-p *Password*

Specifies a password.

BackupDirectory

Specifies the backup directory.

incremental

Implements an incremental backup instead of a full backup.

keeplog

Preserves database log files.

-?

Displays a list of **certutil** commands.

Remarks

- You must specify the *CAComputerName* or *CAName* in **-config *CAComputerName*** ***CAName***. Otherwise, the Select Certificate Authority dialog box appears and displays a list of all CAs that are available.

- If you use **-config -** instead of **-config *CAComputerName\CAName***, the operation is processed using the default CA.
- With a PKCS #12 (.pfx) file, 32 characters is the maximum length allowed for a password.
- If you do not specify **keeplog**, **certutil-backup** combines the database log files into a single log file that is retained upon the successful completion of **-backup** .
- If you do not specify **incremental**, **certutil-backup** performs a full backup.
- You can use the **-f** option to overwrite existing files in *BackupDirectory*.

Examples

To back up keys and certificates for a CA named EnterpriseCA, type:

certutil –p p@ssw23 f:\Backup2\EnterpriseCA

certutil -p p@ssw23 f:\Backup2\EnterpriseCA incremental

certutil -p p@ssw23 f:\Backup2\EnterpriseCA keeplog

To back up a CA database

Syntax

certutil-backupdb [**-f**] [**-gmt**] [**-seconds**] [**-v**] [**-config** *CAMachineName\CAName*] *BackupDirectory*[[**incremental**] [**keeplog**]]

Parameters

-backupdb

Backs up the Certificate Services database.

-f

Overwrites existing files or keys.

-gmt

Displays time as Greenwich mean time.

-seconds

Displays time with seconds and milliseconds.

-v

Specifies verbose output.

-config *CAMachineName\CAName*

processes the operation by using the CA specified in the configuration string (that is, *CAMachineName\CAName*).

BackupDirectory

Specifies the backup directory.

incremental

Implements an incremental backup instead of a full backup.

keeplog

Preserves database log files.

-?

Displays a list of **certutil** commands.

Remarks

- You must specify the *CAComputerName* or *CAName* in **-config *CAComputerName*** ***CAName***. Otherwise, the Select Certificate Authority dialog box appears and displays a list of all CAs that are available.
- If you use **-config -** instead of **-config *CAComputerName**CAName***, the operation is processed using the default CA.
- You can run this command locally or remotely. The server and the CA must be running. Typically, administrators use this command to perform infrequent full backups followed by frequent incremental backups. Each backup must be made into a separate directory tree. Starting with the most recent full backup, all backups are required to correctly restore the database.
- If you do not specify **keeplog**, **certutil-backup** combines the database log files into a single log file that is retained upon the successful completion of **-backup** .
- If you do not specify **incremental**, **certutil-backup** performs a full backup.
- You can use the **-f** option to overwrite existing files in *BackupDirectory*.

To back up the CA certificate and keys

Syntax

certutil-backupkey [**-f**] [**-gmt**] [**-seconds**] [**-v**] [**-config** *CAMachineName\CAName*] [**-p** *Password*] *BackupDirectory*

Parameters

-backupkey

Backs up the Certificate Services certificate and private key.

-f

Overwrites existing files or keys.

-gmt

Displays time as Greenwich mean time.

-seconds

Displays time with seconds and milliseconds.

-v

Specifies verbose output.

-config *CAMachineName\CAName*

processes the operation by using the CA specified in the configuration string (that is, *CAMachineName\CAName*).

-p *Password*

Specifies a password.

BackupDirectory

Specifies the backup directory.

-?

Displays a list of **certutil** commands.

Remarks

- You must specify the *CAComputerName* or *CAName* in **-config *CAComputerName**CAName***. Otherwise, the Select Certificate Authority dialog box appears and displays a list of all CAs that are available.
- If you use **-config -** instead of **-config *CAComputerName**CAName***, the operation is processed using the default CA.
- With a PKCS #12 (.pfx) file, 32 characters is the maximum length allowed for a password.
- You can use the **-f** option to overwrite existing files in *BackupDirectory*.

To restore the CA database, certificates, and keys

Syntax

certutil-restore [**-f**] [**-gmt**] [**-seconds**] [**-v**] [**-config** *CAMachineName\CAName*] [**-p** *Password*] *BackupDirectory*

Parameters

-restore

Restores the CA database, certificates, and keys from the specified *BackupDirectory*.

-f

Overwrites existing files or keys.

-gmt

Displays time as Greenwich mean time.

-seconds

Displays time with seconds and milliseconds.

-v

Specifies verbose output.

-config *CAMachineName\CAName*

processes the operation by using the CA specified in the configuration string (that is, *CAMachineName\CAName*).

-p *Password*

Specifies a password.

BackupDirectory

Specifies the backup directory from which you want to restore the CA database, certificates, and keys.

-?

Displays a list of **certutil** commands.

Remarks

- You must specify the *CAComputerName* or *CAName* in **-config *CAComputerName**CAName***. Otherwise, the Select Certificate Authority dialog box appears and displays a list of all CAs that are available.
- If you use **-config -** instead of **-config *CAComputerName**CAName***, the operation is processed using the default CA.
- With a PKCS #12 (.pfx) file, 32 characters is the maximum length allowed for a password.

To restore the CA database

Syntax

certutil-restoredb [**-f**] [**-gmt**] [**-seconds**] [**-v**] [**-config** *CAMachineName**CAName*] *BackupDirectory*

Parameters

-restoredb

Restores CA database from the specified *BackupDirectory*.

-f

Overwrites existing files or keys.

-gmt

Displays time as Greenwich mean time.

-seconds

Displays time with seconds and milliseconds.

-v

Specifies verbose output.

-config *CAMachineName\CAName*

processes the operation by using the CA specified in the configuration string (that is, *CAMachineName\CAName*).

BackupDirectory

Specifies the backup directory from which you want to restore the CA database.

-?

Displays a list of **certutil** commands.

Remarks

- You must specify the *CAComputerName* or *CAName* in **-config *CAComputerName*** ***CAName***. Otherwise, the Select Certificate Authority dialog box appears and displays a list of all CAs that are available.
- If you use **-config -** instead of **-config *CAComputerName\CAName***, the operation is processed using the default CA.
- The CA server and must not be running. You can run this command locally or remotely.
- To restore a full backup and incremental backups, you must restore the full backup first, and then restore all subsequent incremental backups in any order. To overwrite the existing server database files with the full restore, use **-f** . Do not start the server until all backups are restored.
- When you start the CA server, you initiate database recovery. If you successfully start the CA server (that is, as recorded in the application event log), this indicates restore and recovery were completed successfully. If the server fails to start after you run **-restore** , you receive an error code. For more information if **-restore** fails, you can also view the **RESTOREINPROGRESS** registry key.

To restore the CA certificate and keys from a backup directory or a PKCS #12 (.pfx) file

Syntax

certutil-restorekey [**-f**] [**-gmt**] [**-seconds**] [**-v**] [**-config** *CAMachineName\CAName*] [**-p** *Password*] *BackupDirectory|PFXFile*

Parameters

-restorekey

Restores Certificate Services certificate and private key from the specified *BackupDirectory* or PKCS #12*PFXFile*.

-f

Overwrites existing files or keys.

-gmt

Displays time as Greenwich mean time.

-seconds

Displays time with seconds and milliseconds.

-v

Specifies verbose output.

-config *CAMachineName\CAName*

processes the operation by using the CA specified in the configuration string (that is, *CAMachineName\CAName*).

-p *Password*

Specifies a password.

BackupDirectory

Specifies the backup location of the PKCS #12 PFX file.

PFXFile

Specifies the PKCS #12 PFX file.

-?

Displays a list of **certutil** commands.

Remarks

- You must specify the *CAComputerName* or *CAName* in **-config *CAComputerName**CAName***. Otherwise, the Select Certificate Authority dialog box appears and displays a list of all CAs that are available.
- If you use **-config -** instead of **-config *CAComputerName**CAName***, the operation is processed using the default CA.
- With a PKCS #12 (.pfx) file, 32 characters is the maximum length allowed for a password.

To dump the CA database schema, for example, column names, types, and max sizes

Syntax

certutil-schema [**-f**] [**-gmt**] [**-seconds**] [**-v**] [**-config** *CAMachineName\CAName*] [{**ext** | **attib** | **crl**}]

Parameters

-schema

Dumps the CA database schema.

-f

Overwrites existing files or keys.

-gmt

Displays time as Greenwich mean time.

-seconds

Displays time with seconds and milliseconds.

-v

Specifies verbose output.

-config *CAMachineName\CAName*

processes the operation by using the CA specified in the configuration string (that is, *CAMachineName\CAName*).

ext

Displays the schema for Ext table.

attib

Displays the schema for Attib table.

crl

Displays the schema for the certificate revocation list (CRL).

-?

Displays a list of **certutil** commands.

Remarks

- You must specify the *CAComputerName* or *CAName* in **-config *CAComputerName**CAName***. Otherwise, the Select Certificate Authority dialog box appears and displays a list of all CAs that are available.
- If you use **-config -** instead of **-config *CAComputerName**CAName***, the operation is processed using the default CA.

Examples

To view the CA database schema, type:

certutil -schema

Top of page

Formatting legend

Format	Meaning
Italic	Information that the user must supply
Bold	Elements that the user must type exactly as shown
Ellipsis (...)	Parameter that can be repeated several times in a command line
Between brackets ([])	Optional items

Between braces ({}); choices separated by pipe (|). Example: {even|odd}	Set of choices from which the user must choose only one
Courier font	Code or program output

Top of page

How to Back Up Your Web Server Certificate

Because IIS 5.0 leverages Windows security and security tools, you can use the Certificate Manager tool in Windows 2000 Server to export your IIS 5.0 certificates. Historically, the Key Manager tool performed a backup; however, Key Manager is no longer used by IIS 5.0.

First you need to make sure you are looking at the correct Local Computer certificate store. If you are not, you will have to set this up before you can export certificates.

To view the correct certificate store

1. **Open** Microsoft Management Console.
2. Select the **Add/Remove** Snap-in option on the **Console** menu.
3. Click **Add**.
4. Select the **Certificate Manager** tool.
5. Click **Add**.
6. Select the **Computer account** option.
7. Select the **Local computer** option.
8. Click **Finish**, **Close**, then **OK**.

You are now looking at the correct certificate store.

To export a certificate

1. Click the **Certificate Manager** (Local Computer) node to expand it.
2. Click the **Personal** node to expand it, and then expand the **Certificates** node.
3. Right-click the certificate in question.
4. Select **Export** from the **All tasks** menu option.
5. Click **Next**.
6. Select **Yes** to export the private key.
7. Choose the **PKCS #12** format and enable strong encryption.
8. Type and confirm your password.
9. Enter a file name.
10. Click **Next**, then click **Finish**.

How to Restore Your Web Server Certificate

You can restore certificates by simply double-clicking the PKCS #12 certificate file just created.

Results 11 - 20 for: **pkcs#12**

All Results

View results in another search category by clicking a link in the right column...

Show Me:

All Results

Downloads

Product Information

Support & Troubleshooting

Technical Resources

Training & Books

Partner & Business Resources

Communities & Newsgroups

Microsoft News & Corporate Information

Install a Server Certificate (IIS 6.0)
Install a Server Certificate (IIS 6.0)
http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/a2f35fcd-d3b6-4f39-ba93-041a86f7e17f.mspx

PFXIsPFXBlob
This function attempts to decode the outer layer of a BLOB as a Personal Information Exchange (PFX) packet.

http://msdn.microsoft.com/library/en-us/wcesecurity5/html/wce50lrfPFXIsPFXBlob.asp

Countries Missing from List in the IIS Certificate Wizard

Under Geographical Information in the IIS Certificate Wizard, the entries for Cayman Islands (KY), Grenada (GD), and San Marino (SM) are missing. To work around this problem, use a different tool (such as the Web client that is installed with...
http://support.microsoft.com/default.aspx?scid=kb;en-us;300399

ICEnroll4 | CEnroll createPFX method [Security]

Saves the accepted certificate chain and private key in a Personal Information Exchange (PFX) format string. The PFX format is also known as PKCS #12. This method was first defined in the ICEnroll4 interface.
http://msdn.microsoft.com/library/en-us/seccrypto/security/icenroll4_createpfx.asp

PFXVerifyPassword

This function attempts to decode the outer layer of a BLOB as a Personal Information Exchange (PFX) packet and to decrypt it with the given password. No data from the BLOB is imported.
http://msdn.microsoft.com/library/en-us/wcesecurity5/html/wce50lrfPFXVerifyPassword.asp

RSAPKCS1SignatureFormatter Class (.NET Framework)

Creates an RSA PKCS #1 version 1.5 signature.
http://msdn.microsoft.com/library/en-us/cpref/html/frlrfsystemsecuritycryptographyrsapkcs1signatureformatterclasstopic.asp

DSASignatureDeformatter Class (.NET Framework)

Verifies a Digital Signature Algorithm (DSA) PKCS#1 v1.5 signature.
http://msdn.microsoft.com/library/en-us/cpref/html/frlrfsystemsecuritycryptographydsasignaturedeformatterclasstopic.asp

Methods of IEnroll4 [Security]

The following table shows the methods defined by the IEnroll4 interface, as well as the methods the IEnroll4 interface inherits from IEnroll and IEnroll2.
http://msdn.microsoft.com/library/en-us/seccrypto/security/methods_of_ienroll4.asp

CreateSignature Method (.NET Framework)

DSASignatureFormatter.CreateSignature Method - Creates the signature.
http://msdn.microsoft.com/library/en-us/cpref/html/frlrfsystemsecuritycryptographydsasignatureformatterclasscreatesignaturetopic.asp

DSASignatureFormatter Class (.NET Framework)

Creates a Digital Signature Algorithm (DSA) PKCS#1 v1.5 signature.
http://msdn.microsoft.com/library/en-us/cpref/html/frlrfsystemsecuritycryptographydsasignatureformatterclasstopic.asp

0.093 seconds

Results 11 - 20 < **Previous**|**Next** >

Advanced Search|Search Preferences|Search Help

Search Microsoft.com for

pkcs#12 Go

Search Microsoft.com Worldwide

Choose a different location

Didn't find it here?

Search the entire Internet using MSN Search