Active Directory Certificate Services Migration Guide

Microsoft Corporation

Published: June 2008

Abstract

This white paper discusses the planning and implementation of a migration from an existing Windows public key infrastructure (PKI) to Windows Server® 2008 R2. It describes common migration scenarios, identifies features and scenarios that are supported and recommended, and provides step-by-step instructions for the most common tasks.

Copyright Information

Contents

Active Directory Certificate Services Migration Guide

About this guide

This document provides guidance for migrating a certification authority (CA) to a server that is running Windows Server® 2008 R2 from a server that is running Windows Server 2003, Windows Server 2003 R2, or Windows Server 2008.

Tip

To download a copy of this guide, see http://go.microsoft.com/fwlink/?LinkId=116454.

Target audience

- Administrators or IT operations engineers responsible for planning and performing CA migration to Windows Server 2008 R2.
- Administrators or IT operations engineers responsible for the day-to-day management and troubleshooting of networks, servers, client computers, operating systems, or applications.
- IT operations managers accountable for network and server management.
- IT architects responsible for computer management and security throughout an organization.

Supported migration scenarios

This guide provides you with instructions for migrating an existing server that is running Active Directory® Certificate Services (AD CS) to a server that is running Windows Server 2008 R2. This guide does not contain instructions for migration when the source server is running multiple roles. If your server is running multiple roles, you should design a custom migration procedure that is specific to your server environment, based on the information provided in other role migration guides. To view migration guides for additional roles, see Migrate Server Roles to Windows Server 2008 R2 (http://go.microsoft.com/fwlink/?LinkID=128554).

Note

This guide can be used to migrate a CA from a source server that is also a domain controller to a destination server with a different name. However, migration of a domain controller is not covered by this guide. For information about Active Directory Domain Services (AD DS) migration, see Active Directory Domain Services and DNS Server Migration Guide (http://go.microsoft.com/fwlink/?LinkId=179357).

Supported operating systems

This guide supports migrations from source servers running the operating system versions and service packs listed in the following table. All migrations described in this document assume that the destination server is running Windows Server 2008 R2 (either the full or Server Core installation option) on x64-based hardware.

Source server processor	Source server operating system	Destination server operating system	Destination server processor
x86-based or x64-based	Windows Server 2003 with Service Pack 2	Windows Server 2008 R2, both full and Server Core installation options	x64-based
x86-based or x64-based	Windows Server 2003 R2	Windows Server 2008 R2, both full and Server Core installation options	x64-based
x86-based or x64-based	Windows Server 2008	Windows Server 2008 R2, both full and Server Core installation options	x64-based
x64-based	Windows Server 2008 R2	Windows Server 2008 R2, both full and Server Core installation options	x64-based

What this guide does not provide

- Procedures to upgrade to Windows Server 2008 R2
- Procedures to migrate additional server roles
- Procedures to migrate additional AD CS role services

In general, migration is not required for the following AD CS role services. Instead, you can install and configure these role services on computers running Windows Server 2008 R2 by completing the role service installation procedures. For information about the impact of CA migration on other AD CS role services, see Impact of migration on other computers in the enterprise.

- CA Web Enrollment (http://go.microsoft.com/fwlink/?LinkId=179360)
- Online Responder (http://go.microsoft.com/fwlink/?LinkId=143098)
- Network Device Enrollment (http://go.microsoft.com/fwlink/?LinkId=179362)
- Certificate Enrollment Web Services (http://go.microsoft.com/fwlink/?LinkId=179363)

CA migration overview

Preparing to migrate

- Preparing your destination server
- Backing up your source server
- Preparing your source server

Migrating the certification authority

- Backing up a CA database and private key
- Backing up CA registry settings
- Backing up CAPolicy.inf
- Removing the CA role service from the source server
- Removing the source server from the domain
- Joining the destination server to the domain
- Adding the CA role service to the destination server
- Restoring the CA database and configuration on the destination server
- Granting permissions on AIA and CDP containers
- Additional procedures for failover clustering (optional)

Verifying the migration

- Verifying certificate enrollment
- Verifying CRL publishing

Post-migration tasks

- Upgrading certificate templates in Active Directory Domain Services (AD DS)
- Retrieving certificates after a host name change
- Restoring Active Directory Certificate Services (AD CS) to the source server in the event of migration failure
- Troubleshooting migration

Impact of migration

Impact of migration on the source server

The CA migration procedures described in this guide include decommissioning the source server after migration is completed and CA functionality on the destination server has been verified. If the source server is not decommissioned, then the source server and destination server must

have different names. Additional steps are required to update the CA configuration on the destination server if the name of the destination server is different from the name of the source server.

Impact of migration on other computers in the enterprise

During migration, the CA cannot issue certificates or publish CRLs.

To ensure that revocation status checking can be performed by domain members during CA migration, it is important to publish a CRL that is valid beyond the planned duration of the migration.

Because the authority identification access and CRL distribution point extensions of previously issued certificates may reference the name of the source CA, it is important to either continue to publish CA certificates and CRLs to the same location or provide a redirection solution. For an example of configuring IIS redirection, see Redirecting Web Sites in IIS 6.0 (http://go.microsoft.com/fwlink/?LinkID=179366).

Permissions required to complete the migration

To install an enterprise CA or a standalone CA on a domain member computer, you must be a member of the Enterprise Admins group or Domain Admins group in the domain. To install a standalone CA on a server that is not a domain member, you must be a member of the local Administrators group. Removal of the CA role service from the source server has the same group membership requirements as installation.

Estimated duration

The simplest CA migration can typically be completed within one to two hours. The actual duration of CA migration depends on the number of CAs and the sizes of CA databases.

See also

- AD CS Migration: Preparing to Migrate
- AD CS Migration: Migrating the Certification Authority
- AD CS Migration: Verifying the Migration
- AD CS Migration: Post-Migration Tasks
- AD CS Migration: Appendix A

AD CS Migration: Preparing to Migrate

To reduce the duration of the migration process, you can complete the procedures detailed in this topic before beginning the migration process and taking the certification authority (CA) offline.

Preparing your destination server

Hardware requirements for the destination server

The hardware requirements to install any of the Active Directory Certificate Services (AD CS) role services are the same as the minimum and recommended configurations for installation of Windows Server 2008 R2. This section includes the general hardware recommendations for Windows Server 2008 R2. For detailed requirements, see Windows Server 2008 R2 System Requirements (http://go.microsoft.com/fwlink/?LinkId=117345).

Hardware requirements for AD CS in Windows Server 2008 R2

In addition to the hardware requirements for Windows Server 2008 R2, consider these storage and performance requirements for optimal CA performance and availability:

- The disk space requirements for a CA database depend on the number of certificates that the CA issues. Because a CA stores certificate requests, the issued certificates, and optionally, archived key material, 64 KB of database space per certificate is recommended.
- The operating system, the CA database, and the CA log files should be stored on separate physical disk drives in a multidisk configuration. For optimal CA performance and reliability, consider a redundant array of independent disks (RAID) system, such as RAID 5 for the CA database and log files and RAID 1 or RAID 0+1 for the operating system. A recommended minimum hard disk speed is 10,000 RPM.
- Processor power is generally more important to CA performance than system memory capacity.
- Failover clusters have additional hardware, software, and networking requirements. For more information, see Failover Cluster Requirements (http://go.microsoft.com/fwlink/?LinkId=179369).
- If a hardware security module (HSM) is used by the CA, consult with your HSM vendor to verify compatibility with Windows Server 2008 R2.

Software requirements for the destination server

Enterprise CAs can be installed on computers running Windows Server 2008 R2 Foundation, Windows Server 2008 R2 Standard, Windows Server 2008 R2 Enterprise, or Windows Server 2008 R2 Datacenter. Enterprise CAs cannot be installed on computers running Windows Web Server 2008 R2.

When AD CS in Windows Server 2008 R2 is installed in an Active Directory Domain Services (AD DS) domain, the AD DS schema version must be at least 30 and all domain controllers in the domain must be running one of the following operating systems:

- Windows Server 2008 with Service Pack 1 (SP1)
- Windows Server 2008
- Windows Server 2003 R2
- Windows Server 2003 with Service Pack 2 (SP2)
- Windows Server 2003 with SP1
- Windows Server 2003

Note
Domain controllers running Windows 2000 Server with Service Pack 4 (SP4) or Windows 2000 Server with Service Pack 3 (SP3) are technically compatible with AD CS deployments. However, the use of Windows 2000 Server is not recommended because Mainstream Support is no longer available for this operating system. For more information, see Microsoft Support Lifecycle (http://go.microsoft.com/fwlink/?LinkId=117347).

If an HSM is used by the CA, consult your HSM vendor to verify cryptographic service provider (CSP) and key service provider (KSP) compatibility with Windows Server 2008 R2.

Installing Windows Server 2008 R2

To reduce the duration of the migration process, you can prepare the destination server by completing the following procedures before beginning the migration process and taking the source CA offline.

- Review the hardware and software requirements in the previous sections.
- Install Windows Server 2008 R2. Review **Step-by-Step: Basic Windows Deployment for IT Professionals** and **Deploy Windows Server 2008 R2** for guidance and procedures. For Server Core installations, complete the procedures described in **Deploying a Server core installation: Overview**.
- Install updates by using Windows Update.
- (Optional) Install failover clustering by reviewing the Failover Cluster Deployment Guide (http://go.microsoft.com/fwlink/?LinkId=179364) and completing the procedures described in Checklist: Setting Up a Clustered Instance of a Service or Application. Detailed procedures for installing AD CS and configuring the CA for clustering are described in Adding the CA role service to the destination server.

If you are migrating to a Server Core installation you should configure the server for remote management, which is disabled by default.

Configure remote management on Server Core

1. Log on as an administrator.
2. Type **sconfig.cmd** and press ENTER.
3. Perform the following tasks by completing the procedures described in **Configuring a Server Core installation of Windows Server 2008 R2 with Sconfig.cmd**:
   a. Configure network settings as required for your environment.

b. Join the server to your domain. This step is required if you are setting up an enterprise CA and optional if you are setting up a standalone CA.
c. Configure Remote Management to enable **MMC Remote Management** or **Server Manager Remote Management**.
d. Enable **Remote Desktop** (optional).

4. Type **13** and press ENTER to close sconfig.cmd.

Backing up your source server

Back up your source server to prepare for recovery of the source CA in the event of migration failure.

For more information about creating backups in Windows Server 2008, see the Windows Server Backup Step-by-Step Guide for Windows Server 2008 (http://go.microsoft.com/fwlink/?LinkId=119141).

For more information about creating system state backups in Windows Server 2003, see article 326216 in the Microsoft Knowledge Base (http://go.microsoft.com/fwlink/?LinkId=117369).

Detailed procedures for backing up the source CA database, private key, and registry settings are provided in the topic AD CS Migration: Migrating the Certification Authority.

Preparing your source server

To reduce the duration and impact of CA migration, the following procedures should be completed before you begin migration:

- Back up the CA templates list (required only for enterprise CAs).
- Record the CA's CSP and signature algorithm.
- Publish a CRL with an extended validity period.

Backing up a CA templates list

An enterprise CA can have certificate templates assigned to it. You should record the assigned certificate templates before beginning the CA migration. The information is not backed up with the CA database or registry settings backup. This is because certificate templates and their association with enterprise CAs are stored in AD DS. You will need to add the same list of templates to the destination server to complete CA migration.

Note
It is important that the certificate templates assigned to the source CA are not changed after this procedure is completed.

You can determine the certificate templates assigned to a CA by using the Certification Authority snap-in or the **Certutil.exe –catemplates** command.

To record a CA templates list by using the Certification Authority snap-in

1. Log on with local administrative credentials to the CA computer.
2. Open the Certification Authority snap-in.
3. In the console tree, expand **Certification Authority**, and click **Certificate Templates**.
4. Record the list of certificate templates by taking a screen shot or by typing the list into a text file.

To record a CA templates list by using Certutil.exe

1. Log on with local administrative credentials to the CA computer.
2. Open a Command Prompt window.
3. Type **certutil.exe –catemplates > catemplates.txt** and press ENTER.
4. Verify that the catemplates.txt file contains the templates list.

   Note
   If no certificate templates are assigned to the CA, the file contains an error message: 0x80070490 (Element not found).

Recording a CA's signature algorithm and CSP

During CA installation on the destination server, you can specify the signature algorithm and CSP used by the CA, or accept the default configuration. If your source CA is not using the default configuration, then you should complete the following procedure to record the CSP and signature algorithm.

Note
If an HSM is used by the source CA, follow procedures provided by the HSM vendor to determine the HSM CSP.

To record a CA's CSP by using Certutil.exe

1. Log on with local administrative credentials to the CA computer.
2. Open a Command Prompt window.
3. Type **certutil.exe –getreg ca\csp* > csp.txt** and press ENTER.
4. Verify that the csp.txt file contains the CSP details.

Publishing a CRL with an extended validity period

Before beginning CA migration, it is a good practice to publish a CRL with a validity period that extends beyond the planned migration period. The validity period of the CRL should be at least the length of time that is planned for the migration. This is necessary to enable certificate validation processes on client computers to continue during the migration period.

You should publish a CRL with an extended validity period for each CA being migrated. This procedure is particularly important in the case of a root CA because of the potentially large number of certificates that would be affected by the unavailability of a CRL.

By default, the CRL validity period is equal to the CRL publishing period plus 10 percent. After determining an appropriate CRL validity period, set the CRL publishing interval and manually publish the CRL by completing the following procedures:

Important
Record the value of the CRL publishing period before changing it. After migration is complete, the CRL publishing period should be reset to its previous value.

- Schedule the publication of the certificate revocation list
- Manually publish the certificate revocation list

Caution
Client computers download a new CRL only after the validity period of a locally cached CRL expires. Therefore, you should not use a CRL validity period that is excessively long.

Next steps

After completing the procedures to prepare the source and destination servers, you should review the topic AD CS Migration: Migrating the Certification Authority and complete the procedures appropriate for your specific migration scenario.

See also

- Active Directory Certificate Services Migration Guide
- AD CS Migration: Migrating the Certification Authority
- AD CS Migration: Verifying the Migration
- AD CS Migration: Post-Migration Tasks
- AD CS Migration: Appendix A

AD CS Migration: Migrating the Certification Authority

Review all procedures in this topic and complete only the procedures that are required for your migration scenario.

- Backing up a CA database and private key
- Backing up CA registry settings
- Backing up CAPolicy.inf
- Removing the CA role service from the source server
- Removing the source server from the domain
- Joining the destination server to the domain

- Adding the CA role service to the destination server
- Restoring the CA database and configuration on the destination server
- Granting permissions on AIA and CDP containers
- Additional procedures for failover clustering (optional)

Backing up a CA database and private key

You can back up the CA database and private key by using the Certification Authority snap-in or by using Certutil.exe at a command prompt. Complete either one of the backup procedures described in this section.

Note
If a hardware security module (HSM) is used by the CA, back up the private keys by following procedures provided by the HSM vendor.

After completing backup steps, the Active Directory Certificate Services service (Certsvc) should be stopped to prevent issuance of additional certificates. Before adding the CA role service to the destination server, the CA role service should be removed from the source server.

The backup files created during these procedures should be stored in the same location to simplify the migration. The location should be accessible from the destination server; for example, removable media or a shared folder on the destination server or another domain member.

Backing up a CA database and private key by using the Certification Authority snap-in

The following procedure describes the steps to back up the CA database and private key by using the Certification Authority snap-in while logged on to the source CA.

You must use an account that is a CA administrator. On an enterprise CA, the default configuration for CA administrators includes the local Administrators group, the Enterprise Admins group, and the Domain Admins group. On a standalone CA, the default configuration for CA administrators includes the local Administrators group.

To back up a CA database and private key by using the Certification Authority snap-in

1. Choose a backup location and attach media, if necessary.
2. Log on to the source CA.
3. Open the **Certification Authority** snap-in.
4. Right-click the node with the CA name, point to **All Tasks**, and then click **Back Up CA**.
5. On the **Welcome** page of the CA Backup wizard, click **Next**.
6. On the **Items to Back Up** page, select the **Private key and CA certificate** and **Certificate database and certificate database log** check boxes, specify the backup location, and then click **Next**.
7. On the **Select a Password** page, type a password to protect the CA private key, and

click **Next**.

Security

Use a strong password; for example, at least eight characters long with a combination of uppercase and lowercase characters, numbers, and punctuation characters.

8. On the **Completing the Backup Wizard** page, click **Finish**.
9. After the backup completes, verify the following files in the location you specified:
   - *CAName*.p12 containing the CA certificate and private key
   - Database folder containing files certbkxp.dat, edb#####.log, and *CAName*.edb
10. Open a Command Prompt window, and type **net stop certsvc** to stop the Active Directory Certificate Services service.

    Important

    The service should be stopped to prevent issuance of additional certificates. If certificates are issued by the source CA after a database backup is completed, repeat the CA database backup procedure to ensure the database backup contains all issued certificates.
11. Copy all backup files to a location that is accessible from the destination server; for example, a network share or removable media.

    Security

    The private key must be protected against compromise. Protect a shared folder by limiting its access control list to authorized CA administrators. Protect removable media against unauthorized access and damage.

Backing up a CA database and private key by using Certutil.exe

The following procedure describes the steps to back up the CA database and private key by using Certutil.exe while logged on to the source CA.

You must use an account that is a CA administrator. On an enterprise CA, the default configuration for CA administrators includes the local Administrators group, the Enterprise Admins group, and the Domain Admins group. On a standalone CA, the default configuration for CA administrators includes the local Administrators group.

To back up a CA database and private key by using Certutil.exe

1. Log on with local administrative credentials to the CA computer.
2. Open a Command Prompt window.
3. Type **Certutil.exe –backupdb** <*BackupDirectory*> and press ENTER.
4. Type **Certutil.exe –backupkey** <*BackupDirectory*> and press ENTER.

   Note

BackupDirectory specifies the directory in which the backup files are created. The specified value can be a relative or absolute path. If the specified directory does not exist, it is created. The backup files are created in a subdirectory named Database.

5. Type a password at the prompt, and press ENTER. You must retain a copy of the password to access the key during CA installation on the destination server.

   Security

   Use a strong password; for example, at least eight characters with a combination of uppercase and lowercase characters, numbers, and symbols.

6. Type **net stop certsvc** and press ENTER to stop the Active Directory Certificate Services service. The service must be stopped to prevent issuance of additional certificates.
7. After the backup completes, verify the following files in the location you specified:
   - *CAName*.p12 containing the CA certificate and private key
   - Database folder containing files certbkxp.dat, edb#####.log, and *CAName*.edb

8. Copy all backup files to a location that is accessible from the destination server; for example, a network share or removable media.

   Security

   The private key must be protected against compromise. Protect a shared folder by granting permission to only authorized CA administrators. Protect removable media against unauthorized access and damage.

Backing up CA registry settings

Complete one of the following procedures to back up the CA registry settings.

The files created during the backup procedure should be stored in the same location as the database and private key backup files to simplify the migration. The location should be accessible from the destination server; for example, removable media or a shared folder on the destination server or another domain member.

You must be logged on to the source CA using an account that is a member of the local Administrators group.

To back up CA registry settings by using Regedit.exe

1. Click **Start**, point to **Run**, and type **regedit** to open the Registry Editor.
2. In `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc`, right-click **Configuration**, and then click **Export**.
3. Specify a location and file name, and then click **Save**. This creates a registry file containing CA configuration data from the source CA.
4. Copy the registry file to a location that is accessible from the destination server; for

example, a shared folder or removable media.

To back up CA registry settings by using Reg.exe

1. Open a Command Prompt window.
2. Type **reg export HKLM\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration** *<output file.reg>* and press ENTER.
3. Copy the registry file to a location that is accessible from the destination server; for example, a shared folder or removable media.

Backing up CAPolicy.inf

If your source CA is using a custom CAPolicy.inf file, you should copy the file to the same location as the source CA backup files.

The CAPolicy.inf file is located in the %SystemRoot% directory, which is usually C:\Windows.

Removing the CA role service from the source server

It is important to remove the CA role service from the source server after completing backup procedures and before installing the CA role service on the destination server. Enterprise CAs and standalone CAs that are domain members store in Active Directory Domain Services (AD DS) configuration data that is associated with the common name of the CA. Removing the CA role service also removes the CA's configuration data from AD DS. Because the source CA and destination CA share the same common name, removing the CA role service from the source server after installing the CA role service on the destination server removes configuration data that is required by destination CA and interferes with its operation.

The CA database, private key, and certificate are not removed from the source server by removing the CA role service. Therefore, reinstalling the CA role service on the source server restores the source CA if migration fails and performing a rollback is required. See Restoring AD CS to the source server in the event of migration failure.

Warning

Although it is not recommended, some administrators may choose to leave the CA role service installed on the source server to enable the source CA to be brought online quickly in the case of migration failure. If you choose not to remove the CA role service from the source server before installing the CA role service on the destination server, it is important that you disable the Active Directory Certificate Services service (Certsvc) and shut down the source server before installing the CA role service on the destination server. Do not remove the CA role service from the source server after completing the migration to the destination server. Removing the CA role service from the source server

after migrating to the destination server interferes with the operation of the destination CA.

- To remove the CA on a computer running Windows Server 2003, use the Add/Remove Windows Components wizard.
- To remove the CA on a computer running Windows Server 2008, use the Remove Roles Wizard in Server Manager.

Removing the source server from the domain

Because computer names must be unique within an Active Directory domain, it is necessary to remove the source server from its domain and delete the associated computer account from Active Directory before joining the destination server to the domain.

If you have access to a domain member computer running Windows Server 2008 or Windows Server 2008 R2, complete the following procedure to remove the source server from the domain by using Netdom.exe.

If you do not have access to a computer running Windows Server 2008 or Windows Server 2008 R2, then complete the procedure Join a Workgroup (http://go.microsoft.com/fwlink/?LinkId=207683). Joining a workgroup also removes a domain member computer from its domain.

To remove the source server from the domain by using Netdom.exe

1. On a domain member computer running Windows Server 2008 or Windows Server 2008 R2, open an elevated Command Prompt window.
2. Type **netdom remove** <*source server name*> /d:<*domain name*> /ud:<*domain user account*> /pd:* and press ENTER. For additional command-line options, see Netdom remove syntax (http://go.microsoft.com/fwlink/?LinkID=207681).
3. Shut down the source server.

After removing the source server from its domain, delete the source server's computer account from AD DS by completing the procedure Delete a Computer Account (http://go.microsoft.com/fwlink/?LinkID=138386).

Joining the destination server to the domain

Before joining the destination server to the domain, change the computer name to the same name as the source server. Then complete the procedure to join the destination server to the domain.

If your destination server is running on the Server Core installation option, you must use the command-line procedure.

To rename the destination server, you must be a member of the local Administrators group. To join the server to the domain, you must be a member of the Domain Admins or Enterprise Admins

groups, or have delegated permissions to join the destination server to an organizational unit (OU) in the domain.

Note
If you are migrating a standalone CA that is not a domain member, complete only the steps to rename the destination server and do not join the destination server to the domain.

To join the destination server to the domain by using Netdom.exe

1. On the destination server, open an elevated Command Prompt window.
2. Type **netdom renamecomputer** <*computer name*> /newname:<*new computer name*>
3. Restart the destination server.
4. After the destination server restarts, log on by using an account that has permission to join computers to the domain.
5. Open an elevated Command Prompt window, type **netdom join** <*computer name*> /d:<*domain name*> /ud:<*domain user account*> /pd:* [/ou:<*OU name*>] and press ENTER. For additional command-line options, see Netdom join syntax (http://go.microsoft.com/fwlink/?LinkID=207680).
6. Restart the destination server.

Adding the CA role service to the destination server

This section describes two different procedures for adding the CA role service to the destination server, including special instructions for using failover clustering.

Review the following statements to determine which procedures to complete.

- If your destination server is running the Server Core installation option of Windows Server 2008 R2, you must complete the procedure Adding the CA role service by using SetupCA.vbs.
- If you are migrating to a CA that uses failover clustering, you must review the section "Special instructions for migrating to a failover cluster" and complete the procedures Importing the CA certificate and Adding the CA role service by using Server Manager.
- If you are migrating to a CA that uses an HSM, you must complete the procedures Importing the CA certificate and Adding the CA role service by using Server Manager.
- If none of the above statements describes your migration scenario, you can use either procedure to add the CA role service: Adding the CA role service by using Server Manager or Adding the CA role service by using SetupCA.vbs. If you use Server Manager, you must also complete the procedure Importing the CA certificate.

Special instructions for migrating to a failover cluster

If you are migrating to a failover cluster, the procedures to import the CA certificate and add the CA role service must be completed on each cluster node. After the CA role service is added to each node, you should stop the Active Directory Certificate Services service (Certsvc).

Additionally, it is important to ensure that the shared storage used by the CA is online and assigned to the node you are adding the CA role service to.

The CA database and log files must be located on shared storage. Specify the shared storage location during step 12 of the CA installation procedure.

To verify shared storage is online

1. Log on to the destination server.
2. Start Server Manager.
3. In the console tree, double-click **Storage**, and click **Disk Management**.
4. Ensure that the shared storage is online and assigned to the node you are logged on to.

Importing the CA certificate

If you are adding the CA role service by using Server Manager, you must complete the following procedure to import the CA certificate.

If you are adding the CA role service by using SetupCA.vbs, skip the following procedure and continue at section Adding the CA role service by using SetupCA.vbs.

To import the CA certificate

1. Start the Certificates snap-in for the local computer account.
2. In the console tree, double-click **Certificates (Local Computer)**, and click **Personal**.
3. On the **Action** menu, click **All Tasks**, and then click **Import** to open the Certificate Import Wizard. Click **Next**.
4. Locate the <*CAName*>.p12 file created by the CA certificate and private key backup on the source CA, and click **Open**.
5. Type the password, and click **OK**.
6. Click **Place all certificates in the following store**.
7. Verify **Personal** is displayed in **Certificate store**. If it is not, click **Browse**, click **Personal**, and click **OK**.

   Note
   If you are using a network HSM, complete steps 8 through 10 to repair the association between the imported CA certificate and the private key that is stored in the HSM.
8. In the console tree, double-click **Personal Certificates**, and click the imported CA certificate.
9. On the **Action** menu, click **Open**. Click the **Details** tab, copy the serial number to the

Clipboard, and then click **OK**.

10. Open a Command Prompt window, type **certutil –repairstore My "{SerialNumber}"** and then press ENTER.

Adding the CA role service by using Server Manager

If your destination server is a domain member, you must use an account that is a member of the Domain Admins or Enterprise Admins group in order for the installation wizard to access objects in AD DS.

To add the CA role service by using Server Manager

1. Log on to the destination server, and start Server Manager.
2. In the console tree, click **Roles**.
3. On the **Action** menu, click **Add Roles**.
4. If the **Before you Begin** page appears, click **Next**.
5. On the **Select Server Roles** page, select the **Active Directory Certificate Services** check box, and click **Next**.
6. On the **Introduction to AD CS** page, click **Next**.
7. On the **Role Services** page, click the **Certification Authority** check box, and click **Next**.

   Note
   If you plan to install other role services on the destination server, you should complete the CA installation first, and then install other role services separately. Installation procedures for other AD CS role services are not described in this guide.
8. On the **Specify Setup Type** page, specify either **Enterprise** or **Standalone**, to match the source CA, and click **Next**.
9. On the **Specify CA Type** page, specify either **Root CA** or **Subordinate CA**, to match the source CA, and click **Next**.
10. On the **Set Up Private Key** page, select **Use existing private key** and **Select a certificate and use its associated private key**.

    Note
    If an HSM is used by the CA, select the private key by following procedures provided by the HSM vendor.
11. In the **Certificates** list, click the imported CA certificate, and then click **Next**.

    Note
    If you are using a custom CSP that requires strong private key protection, click **Allow administrator interaction when the private key is accessed by the CA**. The CSPs included with Windows Server do not require this setting to be enabled.
12. On the **Configure Certificate Database** page, specify the locations for the CA database

and log files.

Note
If you are migrating the CA to a failover cluster, the specified locations for database and log files must be on shared storage that is attached to all nodes. Because the location is common to cluster nodes, click **Yes** to overwrite the existing CA database as you add the CA role service to other nodes.

Important
If you specify locations that are different from the locations used on the source CA, then you must also edit the registry settings backup file before the CA is restored. If the locations specified during setup are different from the locations specified in the registry settings, the CA cannot start.

13. On the **Confirm Installation Selections** page, review the messages, and then click **Install**.
14. If you are migrating to a failover cluster, stop the Active Directory Certificate Services service (Certsvc) and HSM service if your CA uses an HSM. Then repeat the procedures to import the CA certificate and add the CA role service on other cluster nodes.

Adding the CA role service by using SetupCA.vbs

You must complete the following procedure if your destination server is running the Server Core installation option of Windows Server 2008 R2. The procedure can also be used on full installations of Windows Server 2008 R2 if you are not using failover clustering or an HSM.

To add the CA role service on a computer running the Server Core installation option of Windows Server 2008 R2

1. Log on to the destination server as a member of the local Administrators group or the Enterprise Admins group.
2. Copy the script Setupca.vbs from AD CS Migration: Appendix A to a directory on the destination server.
3. Copy the directory containing the CA database backup files and CA certificate backup files to a directory on the destination server.

   Important
   The CA database backup files are created in a directory named **Database**. Copy the entire **Database** directory, instead of copying only the backup files within the directory.
4. Type **certutil.exe -importpfx** "<*CAName*>.*p12*" and press ENTER.
5. Type the password for the private key, and press ENTER.
6. Type **certutil.exe -store my | find "Key Container"** and press ENTER.
7. Copy the value of **Key Container** that follows the equals sign (=). Do not include any leading or trailing spaces.

8. Type **Cscript Setupca.vbs /IS /RC /SN "***<Key Container Value>***"**

   Important

   To install a standalone CA, use **/IS**.

   To install an enterprise CA, use **/IE**.

   The value of *<Key Container Name>* is the value you copied in the previous step.

9. Type **net stop certsvc** and press ENTER.

Warning

If you plan to publish the CRL and authority information access extensions on the destination CA, install IIS 7 with the **IIS 6.0 Metabase Compatibility** role feature on the destination CA before you run SetupCA.vbs. Otherwise, the **Enroll** virtual directory is not created or configured by SetupCA.vbs. Alternatively, you can create and configure the **Enroll** virtual directory by running the command **certutil –vroot** after installing IIS 7 with the **IIS 6.0 Metabase Compatibility** role feature.

Restoring the CA database and configuration on the destination server

The procedures in this section should be completed only after the CA role service has been installed on the destination server.

If you are migrating to a failover cluster, add the CA role service to all cluster nodes before restoring the CA database. The CA database should be restored on only one cluster node and must be located on shared storage.

Restoring the source CA backup includes the following tasks:

- Restoring the source CA database on the destination server
- Restoring the source CA registry settings on the destination server
- Verifying certificate extensions on the destination CA
- Restoring the certificate templates list (required only for enterprise CAs)

Restoring the source CA database on the destination server

This section describes two different procedures for restoring the source CA database backup on the destination server.

If you are migrating to a Server Core installation of Windows Server 2008 R2, you must use the procedure "To restore the source CA database and private key backup on the destination CA by using Certutil.exe." In general, it is possible to remotely manage a CA running on a Server Core installation by using the Certification Authority snap-in and Server Manager; however, it is not possible to restore a CA database by using remote management tools.

If you are migrating to a failover cluster, ensure that shared storage is online and restore the CA database on only one cluster node.

To restore the CA database by using the Certification Authority snap-in

1. Log on to the destination server by using an account that is a CA administrator.
2. Start the Certification Authority snap-in.
3. Right-click the node with the CA name, point to **All Tasks**, and then click **Restore CA**. If prompted, click **OK** to stop the CA service.
4. On the **Welcome** page, click **Next**.
5. On the **Items to Restore** page, select **Certificate database and certificate database log**.
6. Click **Browse**, and locate the **Database** directory that contains the CA database files created during the CA database backup.

   
   Note
   Do not select the **Database** directory. Select its parent directory.
7. Type the password that you used to back up the CA database on the source CA.
8. Click **Finish**, and then click **Yes** to restart the CA service.

To restore the CA database by using Certutil.exe

1. Log on to the destination server by using an account that is a CA administrator.
2. Open a Command Prompt window.
3. Type **certutil.exe -f -restoredb** <*CA Database Backup Directory*> and press ENTER.

   
   Note
   The value of <*CA Database Backup Directory*> is the parent directory of the **Database** directory. For example, if the CA database backup files are located in C:\Temp\Database, then the value of <*CA Database Backup Directory*> is C:\Temp.

Restoring the source CA registry settings on the destination server

The CA configuration information is stored in the registry in:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc

Before importing the registry settings from the source CA to the target CA, create a backup of the default target CA registry configuration by using the procedure **Exporting Registry Configuration**. Be sure to perform these steps on the target CA and to name the registry file a name such as "DefaultRegCfgBackup.reg" to avoid confusion.


Important
Some registry parameters should be migrated without changes from the source CA computer, and some should not be migrated. If they are migrated, they should be updated in the target system after migration because some values are associated with

the CA itself, whereas others are associated with the domain environment, the physical host, the Windows version, or other factors that may be different in the target system.

A suggested way of performing the registry configuration import is first to open the registry file you exported from the source CA in a text editor and analyze it for settings that may need to be changed or removed. The following table shows the configuration parameters that should be transferred from the source CA to the target CA.

Registry location	**Configuration parameter**
HKEY_LOCAL_MACHINE\system\currentcontrolset\services\certsvc\Configuration	LDAPFlags
HKEY_LOCAL_MACHINE\system\currentcontrolset\services\certsvc\Configuration*CAname*	DSConfigDN ForceTeletex CRLEditFlags CRLFlags InterfaceFlags (required only if has been changed manually) EnforceX500NameLengths SubjectTemplate ValidityPeriod ValidityPeriodUnits KRACertHash KRACertCount KRAFlags CRLPublicationURLs CRLPeriod CRLPeriodUnits CRLOverlapPeriod CRLOverlapUnits CRLDeltaPeriod CRLDeltaPeriodUnits

Registry location	Configuration parameter
	CRLDeltaOverlapPeriod CRLDeltaOverlapUnits CACertPublicationURLs (check for custom entries with hard-coded host names or other data specific to the source CA) CACertHash
HKEY_LOCAL_MACHINE\system\currentcontrolset\services\certsvc\Configuration*CAname*\ExitModules\CertificateAuthority_MicrosoftDefault.Exit	PublishCertFlags
HKEY_LOCAL_MACHINE\system\currentcontrolset\services\certsvc\Configuration*CAname*\PolicyModules\CertificateAuthority_MicrosoftDefault.Policy	EnableRequestExtensionList EnableEnrolleeRequestExtensionList DisableExtensionList SubjectAltName SubjectAltName2 RequestDisposition EditFlags

To analyze the registry file

1. Right-click the .reg file created by exporting the settings from the source CA.
2. Click **Edit** to open the file in a text editor.
3. If the target CA's computer name is different from the source CA's computer name, search the file for the host name of the source CA computer. For each instance of the host name found, ensure that it is the appropriate value for the target environment. Change the host name, if necessary. Update the **CAServerName** value.

   Important
   If the host name is located in the .reg file as part of the CA name, such as in the

Active value within the **Configuration** key or the **CommonName** value within the **CAName** key, do not change the setting. The CA name must not be changed as part of the migration. This means the new target CA must have the old CA's name, even if part of that name is the old CA's host name.

4. Check any registry values that indicate local file paths, such as the following, to ensure drive letter names and paths are correct for the target CA. If there is a mismatch between the source and the target CA, either update the values in the file or remove them from the file so that the default settings are preserved on the target CA.

   These storage location settings are elected during CA setup. They exist under the Configuration registry key:

   - DBDirectory
   - DBLogDirectory
   - DBSystemDirectory
   - DBTempDirectory

   The following settings under the Configuration\{*CA Name*} registry key contain, in their default values, a local path. (Alternatively, you can update these values after importing them by using the Certification Authority snap-in. The values are located on the CA properties **Extensions** tab.)

   - CACertPublicationURLs
   - CRLPublicationURLs

Any values not listed can retain the value data installed by default with the target CA. You can remove any registry values that you do not want to import into the target CA. Once the text file is edited, it can be imported into the target CA. By importing the source server registry settings backup into the destination server, the source CA configuration is migrated to the destination server.

To import the source CA registry backup on the destination CA

1. Log on to the destination server as a member of the local Administrators group.
2. Open a Command Prompt window.
3. Type **net stop certsvc** and press ENTER.
4. Type **reg import** <*Registry Settings Backup.reg*> and press ENTER.

To edit the CA registry settings

1. Click **Start**, type **regedit.exe** in the **Search programs and files** box, and press ENTER to open the Registry Editor.
2. In the console tree, locate the key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\CertSvc\Configuration`, and click **Configuration**.
3. In the details pane, double-click **DBSessionCount**.
4. Click **Hexadecimal**. In **Value data**, type **64**, and then click **OK**.

5. Verify the locations specified in the following settings are correct for your destination server, and change them as needed to indicate the location of the CA database and log files.
   - DBDirectory
   - DBLogDirectory
   - DBSystemDirectory
   - DBTempDirectory

   Important
   Complete steps 6 through 8 only if the name of your destination server is different from the name of your source server.
6. In the console tree of the registry editor, expand **Configuration**, and click your CA name.
7. Modify the values of the following registry settings by replacing the source server name with the destination server name.

   Note
   In the following list, CACertFileName and ConfigurationDirectory values are created only when certain CA installation options are specified. If these two settings are not displayed, you can proceed to the next step.
   - CAServerName
   - CACertFileName
   - ConfigurationDirectory – This value should appear in Windows Registry under the following location:
     `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\CertSvc\Configuration`.

Verifying certificate extensions on the destination CA

The steps described for importing the source CA registry settings and editing the registry in case of a server name change are intended to retain the network locations that were used by the source CA to publish CRLs and CA certificates. If the source CA was published to default Active Directory locations, after completing the previous procedure, there should be an extension with publishing options enabled and an LDAP URL that references the source server's NetBIOS name; for example,

`ldap:///CN=<CATruncatedName><CRLNameSuffix>,CN=<ServerShortName>,CN=CDP,CN=Public Key Services,CN=Services,<ConfigurationContainer><CDPObjectClass>`.

Because many administrators configure extensions that are customized for their network environment, it is not possible to provide exact instructions for configuring CRL distribution point and authority information access extensions.

Carefully review the configured locations and publishing options, and ensure that the extensions are correct according to your organization's requirements.

To verify extensions by using the Certification Authority snap-in

1. Review and modify the CRL distribution point and authority information access extensions and publishing options by following example procedures described in Specify CRL Distribution Points (http://go.microsoft.com/fwlink/?LinkID=145848).
2. If the destination server name is different from the source server name, add an LDAP URL specifying a location that references the destination server's NetBIOS name with the substitution variable <*ServerShortName*>; for example `ldap:///CN=<CATruncatedName><CRLNameSuffix>,CN=<ServerShortName>,CN=CDP,CN=Public Key Services,CN=Services,<ConfigurationContainer><CDPObjectClass>`.

Restoring the certificate templates list

The following procedure is required only for an enterprise CA. A standalone CA does not have certificate templates.

To assign certificate templates to the destination CA

1. Log on with administrative credentials to the destination CA.
2. Open a command prompt window.
3. Type **certutil -setcatemplates +** <*templatelist*> and press ENTER.

   Note
   Replace <*templatelist*> with a comma-separated list of the template names that are listed in the catemplates.txt file created during the procedure "To record a CA templates list by using Certutil.exe." For example, **certutil -setcatemplates +Administrator,User,DomainController**. Review the list of templates created during Backing up a CA templates list.

Granting permissions on AIA and CDP containers

If the name of the destination server is different from the source server, the destination server must be granted permissions on the source server's CDP and AIA containers in AD DS to publish CRLs and CA certificates. Complete the following procedure in the case of a server name change.

To grant permissions on the AIA and CDP containers

1. Log on as a member of the Enterprise Admins group to a computer on which the Active Directory Sites and Services snap-in is installed.
2. Click **Start**, point to **Run**, type **dssite.msc**, and then click **OK**.
3. In the console tree, click the top node.
4. On the **View** menu, click **Show services node**.
5. In the console tree, expand **Services**, expand **Public Key Services**, and then click **AIA**.
6. In the details pane, right-click the name of the source CA, and then click **Properties**.

7. Click the **Security** tab, and then click **Add**.
8. Click **Object Types**, click **Computers**, and then click **OK**.
9. Type the name of the destination server, and click **OK**.
10. In the **Allow** column, click **Full Control**, and click **Apply**.
11. If the source server object is displayed in **Group or user names**, click the name of the source server, then click **Remove**, and then click **OK**.
12. In the console tree, expand **CDP**, and then click the name of the source server.
13. In the details pane, right-click the **cRLDistributionPoint** item at the top of the list, and then click **Properties**.
14. Click the **Security** tab, and then click **Add**.
15. Click **Object Types**, click **Computers**, and then click **OK**.
16. Type the name of the destination server, and click **OK**.
17. In the **Allow** column, click **Full Control**, and click **Apply**.
18. If the source server object is displayed in **Group or user names**, click the name of the source server, then click **Remove**, and then click **OK**.
19. Repeat steps 13 through 18 for each **cRLDistributionPoint** item.

Additional procedures for failover clustering

If you are migrating to a failover cluster, complete the following procedures after the CA database and registry settings have been migrated to the destination server.

- Configuring failover clustering for the destination CA
- Granting permissions on public key containers
- Editing the DNS name for a clustered CA in AD DS
- Configuring CRL distribution points for failover clusters

Note
Migration of a CA to a failover cluster running on the Server Core installation option of Windows Server 2008 R2 is not described in this guide.

If you are migrating to a failover cluster, complete the following procedures to configure failover clustering for AD CS.

To configure AD CS as a cluster resource

1. Click **Start**, point to **Run**, type **Cluadmin.msc**, and then click **OK**.
2. In the console tree of the Failover Cluster Management snap-in, click **Services and Applications**.
3. On the **Action** menu, click **Configure a service or Application**. If the **Before you begin** page appears, click **Next**.

4. In the list of services and applications, select **Generic Service**, and click **Next**.
5. In the list of services, select **Active Directory Certificate Services**, and click **Next**.
6. Specify a service name, and click **Next**.
7. Select the disk storage that is still mounted to the node, and click **Next**.
8. To configure a shared registry hive, click **Add**, type **SYSTEM\CurrentControlSet\Services\CertSvc**, and then click **OK**. Click **Next** twice.
9. Click **Finish** to complete the failover configuration for AD CS.
10. In the console tree, double-click **Services and Applications**, and select the newly created clustered service.
11. In the details pane, click **Generic Service**. On the **Action** menu, click **Properties**.
12. Change **Resource Name** to **Certification Authority**, and click **OK**.

If you use a hardware security module (HSM) for your CA, complete the following procedure.

To create a dependency between a CA and the network HSM service

1. Open the Failover Cluster Management snap-in. In the console tree, click **Services and Applications**.
2. In the details pane, select the previously created name of the clustered service.
3. On the **Action** menu, click **Add a resource**, and then click **Generic Service**.
4. In the list of available services displayed by the **New Resource** wizard, click the name of the service that was installed to connect to your network HSM. Click **Next** twice, and then click **Finish**.
5. Under **Services and Applications** in the console tree, click the name of the clustered services.
6. In the details pane, select the newly created **Generic Service**. On the **Action** menu, click **Properties**.
7. On the **General** tab, change the service name if desired, and click **OK**. Verify that the service is online.
8. In the details pane, select the service previously named **Certification Authority**. On the **Action** menu, click **Properties**.
9. On the **Dependencies** tab, click **Insert**, select the network HSM service from the list, and click **OK**.

Granting permissions on public key containers

If you are migrating to a failover cluster, complete the following procedures to grant all cluster nodes permissions to on the following AD DS containers:

- The AIA container
- The Enrollment container
- The KRA container

To grant permissions on public key containers in AD DS

1. Log on to a domain member computer as a member of the Domain Admins group or Enterprise Admins group.
2. Click **Start**, point to **Run**, type **dssite.msc**, and then click **OK**.
3. In the console tree, click the top node.
4. On the **View** menu, click **Show services node**.
5. In the console tree, expand **Services**, then **Public Key Services**, and then click **AIA**.
6. In the details pane, right-click the name of the source CA, and then click **Properties**.
7. Click the **Security** tab, and then click **Add**.
8. Click **Object Types**, click **Computers**, and then click **OK**.
9. Type the computer account names of all cluster nodes, and click **OK**.
10. In the **Allow** column, select the **Full Control** check box next to each cluster node, and click **OK**.
11. In the console tree, click **Enrollment Services**.
12. In the details pane, right-click the name of the source CA, and then click **Properties**.
13. Click the **Security** tab, and then click **Add**.
14. Click **Object Types**, click **Computers**, and then click **OK**.
15. Type the computer account names of all cluster nodes, and click **OK**.
16. In the **Allow** column, select the **Full Control** check box next to each cluster node, and click **OK**.
17. In the console tree, click **KRA**.
18. In the details pane, right-click the name of the source CA, then click **Properties**.
19. Click the **Security** tab, and then click **Add**.
20. Click **Object Types**, click **Computers**, and then click **OK**.
21. Type the names of all cluster nodes, and click **OK**.
22. In the **Allow** column, select the **Full Control** check box next to each cluster node, and click **OK**.

Editing the DNS name for a clustered CA in AD DS

When the CA service was installed on the first cluster node, the Enrollment Services object was created and the DNS name of that cluster node was added to the dNSHostName attribute of the Enrollment Services object. Because the CA must operate on all cluster nodes, the value of the dNSHostName attribute of the Enrollment Services object must be the service name specified in step 6 of the procedure "To configure AD CS as a cluster resource."

If you are migrating to a clustered CA, complete the following procedure on the active cluster node. It is necessary to complete the procedure on only one cluster node.

To edit the DNS name for a clustered CA in AD DS

1. Log on to the active cluster node as a member of the Enterprise Admins group.
2. Click **Start**, point to **Run**, type **adsiedit.msc**, and then click **OK**.
3. In the console tree, click **ADSI Edit**.
4. On the **Action** menu, click **Connect to**.
5. In the list of well-known naming contexts, click **Configuration**, and click **OK**.
6. In the console tree, expand **Configuration**, **Services**, and **Public Key Services**, and click **Enrollment Services**.
7. In the details pane, right-click the name of the cluster CA, and click **Properties**.
8. Click **dNSHostName**, and click **Edit**.
9. Type the service name of the CA as displayed under **Failover Cluster Management** in the Failover Cluster Manager snap-in, and click **OK**.
10. Click **OK** to save changes.

Configuring CRL distribution points for failover clusters

In a CA's default configuration, the server's short name is used as part of the CRL distribution point and authority information access locations.

When a CA is running on a failover cluster, the server's short name must be replaced with the cluster's short name in the CRL distribution point and authority information access locations. To publish the CRL in AD DS, the CRL distribution point container must be added manually.

Important
The following procedures must be performed on the active cluster node.

To change the configured CRL distribution points

1. Log on to the active cluster node as a member of the local Administrators group.
2. Click **Start**, click **Run**, type **regedit**, and then click **OK**.
3. Locate the registry key **\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration**.
4. Click the name of the CA.
5. In the right pane, double-click **CRLPublicationURLs**.
6. In the second line, replace **%2** with the service name specified in step 6 of the procedure "To configure AD CS as a cluster resource."

   Tip
   The service name also appears in the Failover Cluster Management snap-in under **Services and Applications**.
7. Restart the CA service.
8. Open a command prompt, type **certutil -CRL**, and press ENTER.

   Note

If a "Directory object not found" error message is displayed, complete the following procedure to create the CRL distribution point container in AD DS.

To create the CRL distribution point container in AD DS

1. At a command prompt, type **cd %windir%\System32\CertSrv\CertEnroll**, and press ENTER. The CRL file created by the certutil –CRL command should be located in this directory.
2. To publish the CRL in AD DS, type **certutil -f -dspublish** *"CRLFile.crl"* and press ENTER.

Next steps

After completing the procedures to migrate the CA, you should complete the procedures described in AD CS Migration: Verifying the Migration.

See also

- Active Directory Certificate Services Migration Guide
- AD CS Migration: Preparing to Migrate
- AD CS Migration: Verifying the Migration
- AD CS Migration: Post-Migration Tasks
- AD CS Migration: Appendix A

AD CS Migration: Verifying the Migration

Complete the following procedures to verify the operation of the destination certification authority (CA).

- Verifying certificate enrollment
- Verifying CRL publishing

Verifying certificate enrollment

To verify migration to an enterprise CA, complete the procedure Request a Certificate (http://go.microsoft.com/fwlink/?LinkId=179367).

You can start autoenrollment for user certificates by completing the following procedure or by running the following command: **certutil.exe -pulse**.

To verify autoenrollment

1. Log on to a domain member computer by using an account that has Autoenroll, Enroll, and Read permissions for the certificate templates that are assigned to the destination

CA.

2. Click **Start**, and then click **Run**.
3. Type **certmgr.msc**, and then click **OK** to open the Certificates snap-in.
4. In the console tree, right-click **Certificates – Current User**, click **All Tasks**, and then click **Automatically Enroll and Retrieve Certificates** to start the Certificate Enrollment wizard.
5. On the **Before You Begin** page, click **Next**.
6. On the **Request Certificates** page, a list of one or more certificate templates should be displayed. Select the check box next to each certificate template that you want to request, and then click **Enroll**.

   Note
   If the correct certificate templates are not displayed, click **Show all templates** to display all certificate templates that are assigned to the issuing CA. A status of **Unavailable** indicates the user account does not have permission to autoenroll for a certificate. Follow the steps in the "To configure certificate templates for autoenrollment" procedure earlier in this topic. For more information, see **Troubleshooting Certificate Enrollment**.
7. Click **Finish** to complete the enrollment process.
8. In the console tree, double-click **Personal**, and then click **Certificates** to display a list of installed user certificates and to verify that the certificate that you requested is displayed.

To verify migration to a standalone CA, complete the following procedure.

To verify manual enrollment by using Certreq.exe

1. Create a certificate request, and save it to a file by completing the procedure Create a Custom Certificate Request (http://go.microsoft.com/fwlink/?LinkId=179368).
2. Open a Command Prompt window.
3. Type **certreq -submit -config "<DestinationServerName\CAName>" "<CertificateRequestInput>" "<CertificateResponseOutput>"** and press ENTER.

   Note
   If a message is displayed indicating that the certificate request is pending, the certificate must be issued by a certificate manager or CA administrator by using the Certification Authority snap-in. After the certificate is issued, it must be retrieved by using the command in step 4. If the certificate is issued immediately by the CA, the file specified in <CertificateResponseOutput> contains the certificate. Use the command in step 5 to install the certificate into the certificate store.
4. Type **certreq –retrieve -config "<DestinationServerName\CAName>" <RequestID> <CertificateResponseOutput>** and press ENTER.
5. Type **certreq –accept -config "<DestinationServerName\CAName>" <CertificateResponseOutput>** and press ENTER.

Option	Description	Example
-config	The –config option is followed by a string specifying a host name and CA name in the format HostName\CAName.	Certreq.exe –submit –config Server1\CA1 C:\RequestFile.txt C:\ResponseFile.cer
DestinationServerName	The host name of the destination server.	
CAName	The CA name being migrated.	
CertificateRequestInput	The path and name of the file containing the certificate request that was created by using the procedure "Create a Custom Certificate Request."	
CertificateResponseOutput	The path and name of the file receiving the issued certificate from the CA. If the certificate request is pending, the file contains a message from the CA indicating the status of the request and the request ID. The request ID is used to retrieve the certificate after it is issued by a certificate manager or CA administrator.	

Verifying CRL publishing

If you published a certificate revocation list (CRL) with an extended validity period before beginning migration, you should change the CRL publishing period back to its pre-migration value by completing the procedure Schedule the publication of the certificate revocation list.

Manually publish a CRL by completing one of the procedures described in Manually Publish a CRL.

Next steps

After completing verification steps, you should review the topic AD CS Migration: Post-Migration Tasks and complete the procedures appropriate for your environment.

See also

- Active Directory Certificate Services Migration Guide
- AD CS Migration: Preparing to Migrate
- AD CS Migration: Migrating the Certification Authority
- AD CS Migration: Post-Migration Tasks
- AD CS Migration: Appendix A

AD CS Migration: Post-Migration Tasks

Post-migration steps can be performed after migration has been completed and the operation of the destination CA has been verified.

If verification steps have failed, review the Troubleshooting section in this topic.

- Upgrading certificate templates in Active Directory Domain Services (AD DS)
- Retrieving certificates after a host name change
- Restoring Active Directory Certificate Services (AD CS) to the source server in the event of migration failure
- Troubleshooting migration

Upgrading certificate templates in Active Directory Domain Services (AD DS)

Review the post-migration steps below and perform only those that are appropriate for your environment and migration scenario.

The following additional default certificate templates are included in enterprise certification authorities (CAs) running on Windows Server 2008 R2 and Windows Server 2008 but are not included in Windows Server 2003:

- OCSP Response Signing
- Kerberos Authentication

These certificate templates are not required for CA operation. OCSP Response Signing certificates are required if you are deploying the Online Responder role service.

If you require these additional certificate templates, complete the following procedure.

To upgrade certificate templates in AD DS by using the Certificate Templates snap-in

1. Log on to the destination server as a member of the Enterprise Admins group.
2. Open the Certificate Templates snap-in. The snap-in automatically adds the default certificate templates to AD DS.

Retrieving certificates after a host name change

If the destination server name is different from the source server name, you might need to manually retrieve any certificates that were issued by the source CA and had not been retrieved before migration.

Complete this procedure on the computer that was used to submit the certificate request to the source CA.

To retrieve a certificate by using Certreq.exe

1. Open a Command Prompt window.
2. Type **certreq –retrieve -config "<DestinationServerName\CAName>" <RequestID> <CertificateResponseOutput>** and press ENTER.
3. Type **certreq –accept <CertificateResponseOutput>** and press ENTER.

Option	Description	Example
-config	The –config option is followed by a string specifying a host name and CA name in the format HostName\CAName.	Certreq.exe –submit –config Server1\CA1 C:\RequestFile.txt C:\ResponseFile.cer
DestinationServerName	The host name of the destination server.	
CAName	The CA name being migrated.	
CertificateRequestInput	The path and name of the file containing the certificate request that was created by using the procedure "Create a Custom Certificate Request."	
CertificateResponseOutput	The path and name of the file receiving the issued certificate from the CA. If the certificate request is pending, the file contains a message from the CA indicating the status of the request and the request ID. The request ID is used to retrieve the	

Option	Description	Example
	certificate after it is issued by a certificate manager or CA administrator.	
RequestID	The Request ID value returned by a CA in response to a certificate request. The Request ID value is displayed in command output and written to the CertificateResponseOutput file.	

Restoring Active Directory Certificate Services (AD CS) to the source server in the event of migration failure

If you removed the CA role service from the source server as described in the procedure Removing the CA role service from the source server, you can restore the source CA by reinstalling the CA role service on the source server. It is important to remove the CA role service from the destination server before reinstalling the CA role service on the source server.

If you did not remove the CA role service from the source server, you should not remove the CA role service from the destination server. Simply shut down the destination CA and start the source CA.

Rollback procedures can be completed in less than one hour.

To remove the CA role service from the destination server, use the Remove Roles Wizard in Server Manager.

To add the CA role service to a source server running Windows Server 2003, use the Add/Remove Windows Components wizard.

To add the CA role service to a source server running Windows Server 2008, use the Add Roles Wizard in Server Manager.

Troubleshooting migration

If you encounter errors during verification procedures, use Event Viewer to review the Application log on the destination CA. View an Error event in the preview pane or event properties, and click **Event Log Online Help** to open a Web page with troubleshooting procedures for that event.

For the full collection of documented AD CS events, see AD CS Events and Errors.

See also

- Active Directory Certificate Services Migration Guide
- AD CS Migration: Preparing to Migrate
- AD CS Migration: Migrating the Certification Authority
- AD CS Migration: Verifying the Migration
- AD CS Migration: Appendix A

AD CS Migration: Appendix A

The script in this section can be used to automate the addition of the Certification Authority role service to a computer running Windows Server 2008 or Windows Server 2008 R2.

SetupCA.vbs

To save SetupCA.vbs to a file

1. Click **Copy Code** at the top of the code section.
2. Start Notepad.
3. On the **Edit** menu, click **Paste**.
4. On the **File** menu, click **Save**.
5. Type a path for the file, type the file name **Setupca.vbs**, and click **Save**.

```
'Copyright (c) Microsoft Corporation.  All rights reserved.


'Disclaimer
'
'This sample script is not supported under any Microsoft standard support
'program or service. This sample script is provided AS IS without warranty of
'any kind. Microsoft further disclaims all implied warranties including,
'without limitation, any implied warranties of merchantability or of fitness
'for a particular purpose. The entire risk arising out of the use or
'performance of the sample scripts and documentation remains with you. In no
'event shall Microsoft, its authors, or anyone else involved in the creation,
'production, or delivery of the scripts be liable for any damages whatsoever
'(including, without limitation, damages for loss of business profits, business
'interruption, loss of business information, or other pecuniary loss) arising
```

```
'out of the use of or inability to use this sample script or documentation,
'even if Microsoft has been advised of the possibility of such damages.




.
Option Explicit

'*********************************************************************
'Displays script-understood command line parameters
'
Sub Usage()
    Call OutputLine(ECHOMINIMAL, "SetupCA.vbs - Certificate Services Setup Automation for
Windows Server 2008 or Windows Server 2008 R2")
    Call OutputLine(ECHOMINIMAL, "")
    Call OutputLine(ECHOMINIMAL, "Parameters:")
    Call OutputLine(ECHOMINIMAL, "/SP <Prov>   - Specify Provider")
    Call OutputLine(ECHOMINIMAL, "/SK <Len>    - Specify Key length")
    Call OutputLine(ECHOMINIMAL, "/SA <Alg>    - Specify Hash algorithm")
    Call OutputLine(ECHOMINIMAL, "/SN <Name>   - Specify CA Name")
    Call OutputLine(ECHOMINIMAL, "/DN <Name>   - Specify DN Suffix for CA cert subject")
    Call OutputLine(ECHOMINIMAL, "/SR <CA>     - Specify Root CA (Required for
subordinate CA" & Chr(39) & "s and Web service)")
    Call OutputLine(ECHOMINIMAL, "")
    Call OutputLine(ECHOMINIMAL, "/OR <File>   - Save CA cert request to a file (Required
for offline root CA" & Chr(39) & "s)")
    Call OutputLine(ECHOMINIMAL, "")
    Call OutputLine(ECHOMINIMAL, "/RK <Name>   - Reuse Key")
    Call OutputLine(ECHOMINIMAL, "/RC <Name>   - Reuse Cert and Key")
    Call OutputLine(ECHOMINIMAL, "")
    Call OutputLine(ECHOMINIMAL, "/interactive - Specify whether CA will be set to
interact with desktop")
    Call OutputLine(ECHOMINIMAL, "")
    Call OutputLine(ECHOMINIMAL, "/IE          - Install Enterprise Root CA Service")
```

```
    Call OutputLine(ECHOMINIMAL, "/IS          - Install Standalone Root CA Service")
    Call OutputLine(ECHOMINIMAL, "/IF          - Install Enterprise Subordinate CA
Service")
    Call OutputLine(ECHOMINIMAL, "/IT          - Install Standalone Subordinate CA
Service")
    Call OutputLine(ECHOMINIMAL, "/IW          - Install web CA Service - works with any
of the above or by itself")
    Call OutputLine(ECHOMINIMAL, "               This option is not relevant for Server
Core installations")
    Call OutputLine(ECHOMINIMAL, "")
    Call OutputLine(ECHOMINIMAL, "/UC          - Uninstall CA Service")
    Call OutputLine(ECHOMINIMAL, "")
    Call OutputLine(ECHOMINIMAL, "/?           - Display this usage")
    Call OutputLine(ECHOMINIMAL, "")
End Sub ' Usage

'*************************************************************
'Define external constant values
'
' CA Role
Const ENTERPRISE_ROOTCA = 0
Const ENTERPRISE_SUBCA = 1
Const STANDALONE_ROOTCA = 3
Const STANDALONE_SUBCA = 4
Const NO_INSTALL_CA =  -1
Const UNINSTALL_CA = 8
Const UNINSTALL_WEB_PAGES = 9

'FileSystemObject defines
Const FILE_FLAG_READ = 1
Const FILE_FLAG_WRITE = 2
Const FILE_FLAG_APPEND = 8

'Logging level
```

```
Const ECHOMINIMAL = 1

'Error codes to handle:
Const RPC_UNAVAILABLE =  - 2147023174 '0x800706BA
Const DOMAIN_UNAVAILABLE =  - 2147023541 '0x8007054B
Const REG_VALUE_NOT_FOUND =  - 2147024894 '0x80070002
Const IMAGE_TAMPERED =  - 2147024319 '0x80070241
Const VALUE_OUT_OF_RANGE =  - 2147016574 '0x80072082
Const ROOT_CA_NOT_FOUND = 462

'Properties that can be set:
Const SETUPPROP_INVALID =  - 1
Const SETUPPROP_CATYPE = 0
Const SETUPPROP_CAKEYINFORMATION = 1
Const SETUPPROP_INTERACTIVE = 2
Const SETUPPROP_CANAME = 3
Const SETUPPROP_CADSSUFFIX = 4
Const SETUPPROP_VALIDITYPERIOD = 5
Const SETUPPROP_VALIDITYPERIODUNIT = 6
Const SETUPPROP_EXPIRATIONDATE = 7
Const SETUPPROP_PRESERVEDATABASE = 8
Const SETUPPROP_DATABASEDIRECTORY = 9
Const SETUPPROP_LOGDIRECTORY = 10
Const SETUPPROP_SHAREDFOLDER = 11
Const SETUPPROP_PARENTCAMACHINE = 12
Const SETUPPROP_PARENTCANAME = 13
Const SETUPPROP_REQUESTFILE = 14
Const SETUPPROP_WEBCAMACHINE = 15
Const SETUPPROP_WEBCANAME = 16

'*****************************************************************
'Define constants and defaults
'
Const CONST_ERROR = 0
```

```
Const CONST_WSCRIPT = 1
Const CONST_CSCRIPT = 2
Const CONST_SHOW_USAGE = 3
Const CONST_PROCEED = 4

Const DEFCANAME = ""
Const DEFDNSUFFIX = ""
Const DEFROOTCANAME = ""
Const DEF_SEL_KEY_SIZE = "2048"
Const DEF_SEL_HASH_ALG = "SHA1"
Const DEF_INSTALL_WEB_OPTION = False
Const DEF_INSTALL_SVC_OPTION = False
Const DEF_LOG_FILENAME = "_SetupCA.log"
Const DEF_INTERACTIVE = False

'example Capi1 Provider:   "Microsoft Strong Cryptographic Provider"
'example RSA CNG provider: "RSA#MicrosoftKSP"
'example ECC 256 provider: "ECDSA_P256#Microsoft Software Key Storage Provider"
'example ECC 384 provider: "ECDSA_P384#Microsoft Software Key Storage Provider"
'example ECC 521 provider: "ECDSA_P521#Microsoft Software Key Storage Provider"
Const DEF_SEL_PROVIDER = "RSA#Microsoft Software Key Storage Provider"

'Cert Server Role
Dim eCARole
eCARole = NO_INSTALL_CA

'Root CA's name (if this is a subordinate)
Dim strRootCAName
strRootCAName = DEFROOTCANAME

'This CA's name
Dim strCAName
Dim strDNSuffix
strCAName = DEFCANAME
```

```
strDNSuffix = DEFDNSUFFIX

'Crypto provider to be used to sign certs this CA Issues
Dim strSelectedCSP
strSelectedCSP = "" ' DEF_SEL_PROVIDER

'Hash algorithm to be used to sign certs this CA Issues
Dim strSelectedHashAlg
strSelectedHashAlg = "" ' DEF_SEL_HASH_ALG

'Signing key length
Dim iSelectedKeySize
iSelectedKeySize = "" ' DEF_SEL_KEY_SIZE

'Save request to file, for submitting to offline root
Dim strRequestFile
strRequestFile = ""

'Key/Cert Reuse flags
Dim bReuseKey
Dim bReuseCert
Dim bReuseDB
bReuseKey  = False
bReuseCert = False
bReuseDB   = False

'Interactive Flag
Dim bInteractive
bInteractive = DEF_INTERACTIVE

'Default to install or uninstall
Dim bInstall
bInstall = True
```

```
'Install the Web interface
Dim bWebPages
bWebPages = DEF_INSTALL_WEB_OPTION

'Install the Cert Server service.
Dim bInstallService
bInstallService = DEF_INSTALL_SVC_OPTION

'Log file
Dim OutputFile
Dim OutputFile2

'Needs to differentiate which package needs to be installed
Dim PKGCA
Dim PKGIIS
Dim PKGWEB
PKGCA  = True
PKGIIS = True
PKGWEB = True

'Set if installing on core build
Dim bIsCore
bIsCore = False

'For the 'retry once' implementation
Dim bRecursed
bRecursed = False



'Begin script logic




Call VerifyStandardStreams()
```

```
'Set up Local logging
Set OutputFile = CreateLogFile(DEF_LOG_FILENAME)

Dim g_oCASetup

'Start the script
Call Main()

'*********************************************************************
'*
'* Sub InstallPackages()
'*
'* Purpose: Install all required packagemanager packages
'*
'*********************************************************************'
Sub InstallPackages(Install)

    'Get shell object to determine system drive value
    Dim WshShell
    Set WshShell = WScript.CreateObject("WScript.Shell")

    If (Install = True) Then

        If (PKGCA = True) Then
            Call OutputLine(ECHOMINIMAL, "Installing CA Packages, this will take several
minutes...")
            Call WshShell.Run ("cmd /c servermanagercmd -install ADCS-Cert-Authority -
resultPath installResult.xml", 0 , True)
        End If

        If (PKGWEB = True) Then
            Call OutputLine(ECHOMINIMAL, "Installing Web Page Packages, this will take
several minutes...")
```

```
            Call WshShell.Run ("cmd /c servermanagercmd -install ADCS-Web-Enrollment -
resultPath installResult.xml", 0 , True)
        End If

    Else

        If (PKGWEB = True) Then
            Call OutputLine(ECHOMINIMAL, "Removing Web Page Packages, this will take
several minutes...")
            Call WshShell.Run ("cmd /c servermanagercmd -remove ADCS-Web-Enrollment -
resultPath installResult.xml", 0 , True)
        End If

        If (PKGCA = True) Then
            Call OutputLine(ECHOMINIMAL, "Removing CA Packages, this will take several
minutes...")
            Call WshShell.Run ("cmd /c servermanagercmd -remove ADCS-Cert-Authority -
resultPath installResult.xml", 0 , True)
        End If

    End If

    Call OutputLine(ECHOMINIMAL, "Installing Packages, this will take several
minutes...")

    Set WShShell = Nothing
End Sub 'InstallPackage

'*********************************************************************
'*
'* Sub Main()
'*
'* Purpose: Execute the main script logic
'* Input:
'*
```

```
'* Output:
'*
'*********************************************************************
Sub Main ()
    Dim intOpMode

    'Parse the command line
    intOpMode = intParseCmdLine()

    Select Case intOpMode

        Case CONST_SHOW_USAGE
            Call Usage()
            Exit Sub

        Case CONST_PROCEED
            'Do Nothing

        Case CONST_ERROR
            Call OutputLine(ECHOMINIMAL,"Error occurred in passing parameters.")
            Exit Sub

        Case Else                    'Default -- should never happen
            Call OutputLine(ECHOMINIMAL,"Error occurred in passing parameters.")
            Exit Sub

    End Select

    'Check if certocm.dll is present; if not, we are most likely running a Server Core
installation and need
    'to use ocsetup to install the CA package to get certocm.dll
    Dim FSO
    Set FSO = CreateObject("Scripting.FileSystemObject")
```

```
    Dim WshShell
    Dim envVars
    Dim strWinDir
    Set WshShell = WScript.CreateObject("WScript.Shell")
    Set envVars  = WshShell.Environment("process")

    strWinDir    = envVars("windir")

    wscript.echo "Checking if certocm.dll is present..."

    If Not FSO.FileExists(strWinDir + "\system32\certocm.dll") Then
        bisCore = True
        wscript.echo "Certocm.dll is not present; installing CA package..."
        Call WshShell.Run ("cmd /c start /w ocsetup CertificateServices /norestart
/quiet", 0 , True)
        wscript.echo "CA package installed..."
    Else
        wscript.echo "Certocm.dll is present; not installing CA package"
    End If

    Set WshShell = Nothing
    Set envVars  = Nothing

    Set g_oCASetup = CreateObject("certocm.CertSrvSetup")

    'Install Packages
    Call OutputLine(ECHOMINIMAL,"Proceeding to update packages ...")
    Call InstallPackages(bInstall)
wscript.echo "bInstallService: " & bInstallService
wscript.echo "eCARole: " & eCARole
wscript.echo "bWebPages: " & bWebPages
    If ((eCARole <> NO_INSTALL_CA) And (eCARole <> UNINSTALL_CA) And (eCARole <>
UNINSTALL_WEB_PAGES)) or (bWebPages <> False) Then
```

```
        Call OutputLine(ECHOMINIMAL, "Main: Info collection complete. Starting install
phase..." )



        Call OutputFile.WriteLine("Main: Installing...")


        If (True = InstallAndVerifyCA(eCARole, bInstallService, bWebPages)) Then
            Call OutputFile.WriteLine("Main: Install complete! Passed")
        Else
            Call OutputFile.WriteLine("Main: Install failed")
            Call WScript.Quit (1)
        End If 'Installed without errors
    Else
        If (eCARole = UNINSTALL_CA or eCARole = UNINSTALL_WEB_PAGES) Then
            If (eCARole = UNINSTALL_WEB_PAGES) Then
                Call OutputLine(ECHOMINIMAL, "Main: Uninstalling Web pages only...")
                'Uninstall web pages only
                Call UninstallCA(True)
                Call OutputLine(ECHOMINIMAL, "Main: web pages Uninstalled!")
            Else
                Call OutputLine(ECHOMINIMAL, "Main: Uninstalling CA...")
                'Uninstall web pages only
                Call UninstallCA(False)
                Call OutputLine(ECHOMINIMAL, "Main: Uninstalled!")
            End If
        End If
    End If


    ' Clean Up
    Call OutputFile.Close()


End Sub 'Main
```

```
'*********************************************************************
'*
'* Sub VerifyStandardStreams()
'*
'* Purpose: Verify CScript.exe was used to launch this script
'*
'*********************************************************************
Sub VerifyStandardStreams()
    On Error Resume Next

    'Attempt to write to the error stream
    Call WScript.StdOut.WriteLine()

    'If cannot display the error because cscript wasn't used,

    If (Err.Number <> 0) Then

        'Report problem
        Call WScript.Echo("Please run this script from cscript.")

        'Exit the script
        Call WScript.Quit (1)
    End If

    On Error Goto 0
End Sub 'VerifyStandardStreams

'*********************************************************************
'*
'* Sub OutputLine()
'*
'* Purpose: Control the debug output at one location
'*
```

```
'* Input:   Level   compare to verbosity - if lower, do not display
'*          string  String to output.
'*
'***********************************************************************
Sub OutputLine(ByVal level, ByVal String)

    Call OutputFile.WriteLine(String)
    WScript.StdOut.WriteLine String

End Sub ' OutputLine

'***********************************************************************
'*
'* Sub PrintErrorInfo()
'*
'* Purpose: Control the debug output at one location
'*
'* Input:   Message    Message to log
'*          Err        Error object to get info from
'*
'***********************************************************************
Sub PrintErrorInfo(ByVal Message, ByVal oErr)
    Call OutputLine(ECHOMINIMAL, Message)
    Call OutputLine(ECHOMINIMAL, "Error Info: " & oErr.Number & ": " & oErr.Description)
    Call OutputLine(ECHOMINIMAL, "Error Source: " & oErr.Source)
End Sub ' OutputLine

'***********************************************************************
'*
'* Function intParseCmdLine()
'*
'* Purpose: Parse the command line.
'*
'* Input:   none
```

```
'*
'* Output:  none
'*
'**********************************************************************
Function intParseCmdLine()
    On Error Resume Next

    Dim strFlag
    Dim intState
    Dim ArgTemp
    Dim intArgIter
    Dim objFileSystem

    If Wscript.Arguments.Count > 0 Then
        Call OutputFile.WriteLine("parsing arguments: ")

        For Each ArgTemp in WScript.Arguments

            If (InStr(ArgTemp," ") > 0) Then
                Call OutputFile.Write(Chr(34) & ArgTemp & Chr(34) & " ")
            Else
                Call OutputFile.Write(ArgTemp & " ")
            End If

        Next ' ArgTemp

        Call OutputFile.WriteLine
        strFlag = Wscript.arguments.Item(0)
    End If

    'No arguments have been received

    If IsEmpty(strFlag) Then
        intParseCmdLine = CONST_SHOW_USAGE
```

```
    Exit Function ' intParseCmdLine
End If



If (strFlag = "help") Or (strFlag = "/h") Or (strFlag = "\h") Or (strFlag = "-h") _
    Or (strFlag = "\?") Or (strFlag = "/?") Or (strFlag = "?") _
    Or (strFlag = "h") Then
    intParseCmdLine = CONST_SHOW_USAGE
    Exit Function ' intParseCmdLine
End If

'Retrieve the command line and set appropriate variables
intArgIter = 0

Do While intArgIter <= Wscript.arguments.Count - 1

    Select Case Left(LCase(Wscript.arguments.Item(intArgIter)),4)
        Case "/int"
            bInteractive = True
            intArgIter   = intArgIter + 1

        Case "/sp"

            If Not blnGetArg("Crypto Provider", strSelectedCSP, intArgIter) Then
                intParseCmdLine = CONST_ERROR
                Exit Function ' intParseCmdLine
            End If

            intArgIter = intArgIter + 1

        Case "/sk"

            If Not blnGetArg("Key length", iSelectedKeySize, intArgIter) Then
```

```
            intParseCmdLine = CONST_ERROR
            Exit Function ' intParseCmdLine
        End If

        intArgIter = intArgIter + 1

    Case "/sa"

        If Not blnGetArg("Hash algorithm",strSelectedHashAlg, intArgIter) Then
            intParseCmdLine = CONST_ERROR
            Exit Function ' intParseCmdLine
        End If

        intArgIter = intArgIter + 1

    Case "/sn"

        If Not blnGetArg("CA Name", strCAName, intArgIter) Then
            intParseCmdLine = CONST_ERROR
            Exit Function ' intParseCmdLine
        End If

        intArgIter = intArgIter + 1

    Case "/dn"

        If Not blnGetArg("DN Suffix", strDNSuffix, intArgIter) Then
            intParseCmdLine = CONST_ERROR
            Exit Function ' intParseCmdLine
        End If

        intArgIter = intArgIter + 1
```

```
Case "/sr"

    If Not blnGetArg("Root CA", strRootCAName, intArgIter) Then
        intParseCmdLine = CONST_ERROR
        Exit Function ' intParseCmdLine
    End If

    intArgIter = intArgIter + 1

Case "/or"

    If Not blnGetArg("Request File", strRequestFile, intArgIter) Then
        intParseCmdLine = CONST_ERROR
        Exit Function ' intParseCmdLine
    End If

    intArgIter = intArgIter + 1

Case "/iw"

    If bIsCore = False Then
        bWebPages = True
    End If

    intArgIter = intArgIter + 1

Case "/ie"

    If (eCARole <> NO_INSTALL_CA) Then
        intParseCmdLine = CONST_ERROR
        Exit Function ' intParseCmdLine
    End If

    intParseCmdLine = CONST_PROCEED
```

```
    bInstallService = True
    eCARole         = ENTERPRISE_ROOTCA
    intArgIter      = intArgIter + 1

Case "/is"

    If (eCARole <> NO_INSTALL_CA) Then
        intParseCmdLine = CONST_ERROR
        Exit Function ' intParseCmdLine
    End If

    intParseCmdLine = CONST_PROCEED
    bInstallService = True
    eCARole         = STANDALONE_ROOTCA
    intArgIter      = intArgIter + 1

Case "/if"

    If (eCARole <> NO_INSTALL_CA) Then
        intParseCmdLine = CONST_ERROR
        Exit Function ' intParseCmdLine
    End If

    intParseCmdLine = CONST_PROCEED
    bInstallService = True
    eCARole         = ENTERPRISE_SUBCA
    intArgIter      = intArgIter + 1

Case "/it"

    If (eCARole <> NO_INSTALL_CA) Then
        intParseCmdLine = CONST_ERROR
        Exit Function ' intParseCmdLine
    End If
```

```
    intParseCmdLine = CONST_PROCEED
    bInstallService = True
    eCARole         = STANDALONE_SUBCA
    intArgIter      = intArgIter + 1

Case "/uc"

    If (eCARole <> NO_INSTALL_CA) And (eCARole <> UNINSTALL_CA)  Then
        intParseCmdLine = CONST_ERROR
        Exit Function ' intParseCmdLine
    End If

    bInstallService = False
    bWebPages      = False
    bInstall        = False
    eCARole         = UNINSTALL_CA
    intParseCmdLine = CONST_PROCEED
    intArgIter      = intArgIter + 1

Case "/uw"

    If (eCARole <> NO_INSTALL_CA) And (eCARole <> UNINSTALL_CA) Then
        intParseCmdLine = CONST_ERROR
        Exit Function ' intParseCmdLine
    End If

    bWebPages      = False
    bInstall        = False
    eCARole         = UNINSTALL_WEB_PAGES
    intParseCmdLine = CONST_PROCEED
    intArgIter      = intArgIter + 1

Case "/rk"
```

```
                bReuseKey  = True
                intArgIter = intArgIter + 1

            Case "/rc"
                bReuseCert = True
                intArgIter = intArgIter + 1

            Case "/rcd"
                bReuseCert = True
                bReuseDB   = True
                intArgIter = intArgIter + 1

                'Deprecated switches kept to prevent automation from failing
            Case "/sl"
                intArgIter = intArgIter + 2
            Case "/sc"
                intArgIter = intArgIter + 2
            Case "/si"
                intArgIter = intArgIter + 2

            Case Else

                Call OutputLine(ECHOMINIMAL, "Invalid or misplaced parameter: " &
Wscript.arguments.Item(intArgIter))
                Call OutputLine(ECHOMINIMAL, "Please check the input and try again")
                Call OutputLine(ECHOMINIMAL, "or invoke with " & Chr(39) & "/?" & Chr(39)
& " for help with the syntax.")
                Wscript.Quit

        End Select

    Loop '** intArgIter <= Wscript.arguments.Count - 1

    intParseCmdLine = CONST_PROCEED
```

```
End Function

'**********************************************************************
'*
'*  Function blnGetArg()
'*
'*  Purpose: Helper to intParseCmdLine()
'*
'*  Usage:
'*
'*     Case "/s"
'*       blnGetArg ("server name", strServer, intArgIter)
'*
'**********************************************************************

Private Function blnGetArg (ByVal StrVarName, _
    ByRef strVar, _
    ByRef intArgIter)

    blnGetArg = False 'failure, changed to True upon successful completion
    Err.Clear

    If Len(Wscript.Arguments(intArgIter)) > 3 Then

        If Mid(Wscript.Arguments(intArgIter),4,1) = ":" Then

            If Len(Wscript.Arguments(intArgIter)) > 4 Then
                strVar    = Right(Wscript.Arguments(intArgIter), _
                Len(Wscript.Arguments(intArgIter)) - 4)
                blnGetArg = True
                Exit Function
            Else
                intArgIter = intArgIter + 1
```

```
                If intArgIter > (Wscript.Arguments.Count - 1) Then
                    Call OutputLine(ECHOMINIMAL, "Parameter Missing: " & StrVarName &
"."
                    Call OutputLine(ECHOMINIMAL, "Invalid " & StrVarName & ".")
                    Call OutputLine(ECHOMINIMAL, "Please check the input and try again.")
                    Exit Function
                End If

                strVar = Wscript.Arguments.Item(intArgIter)

                If Err.Number Then
                    Call OutputLine(ECHOMINIMAL, "Error: " & Err.Number & ": " &
Err.Description & ".")
                    Call OutputLine(ECHOMINIMAL, "Invalid " & StrVarName & ".")
                    Call OutputLine(ECHOMINIMAL, "Please check the input and try again.")
                    Exit Function
                End If

                If InStr(strVar, "/") Then
                    Call OutputLine(ECHOMINIMAL, "Invalid " & StrVarName)
                    Call OutputLine(ECHOMINIMAL, "Invalid Parameter was:" & StrVar)
                    Call OutputLine(ECHOMINIMAL, "Please check the input and try again.")
                    Exit Function
                End If

                blnGetArg = True 'success
            End If

        Else
            strVar    = Right(Wscript.Arguments(intArgIter), _
            Len(Wscript.Arguments(intArgIter)) - 4)
            blnGetArg = True 'success
            Exit Function
```

```
        End If

    Else
        intArgIter = intArgIter + 1

        If intArgIter > (Wscript.Arguments.Count - 1) Then
            Call OutputLine(ECHOMINIMAL, "Parameter Missing: " & StrVarName & ".")
            Call OutputLine(ECHOMINIMAL, "Invalid " & StrVarName & ".")
            Call OutputLine(ECHOMINIMAL, "Please check the input and try again.")
            Exit Function
        End If

        strVar = Wscript.Arguments.Item(intArgIter)

        If Err.Number Then
            Call OutputLine(ECHOMINIMAL, "Error: " & Err.Number & ": " & Err.Description
& ".")
            Call OutputLine(ECHOMINIMAL, "Invalid " & StrVarName & ".")
            Call OutputLine(ECHOMINIMAL, "Please check the input and try again.")
            Exit Function
        End If

        If InStr(strVar, "/") Then
            Call OutputLine(ECHOMINIMAL, "Invalid " & StrVarName)
            Call OutputLine(ECHOMINIMAL, "Invalid Parameter was:" & StrVar)
            Call OutputLine(ECHOMINIMAL, "Please check the input and try again.")
            Exit Function
        End If

        blnGetArg = True 'success
    End If

End Function
```

```
'*********************************************************************
'*
'* Function CreateLogFile()
'*
'* Purpose: Create the local log file of all of the script output
'*
'* Input:  strLogFileName
'*
'*********************************************************************
Function CreateLogFile(ByVal strLogFileName)
    Dim FileSystem
    Set FileSystem = CreateObject("Scripting.FileSystemObject")

    'Get the actual path
    Dim strFileName
    strFileName = FileSystem.GetAbsolutePathName(strLogFileName)

    Call WScript.StdOut.WriteLine ("Log file = " & strFileName)

    On Error Resume Next

    ' just append to

    If FileSystem.FileExists(strFileName) Then
        'Open Existing log
        Set CreateLogFile = FileSystem.OpenTextFile(strFileName, FILE_FLAG_APPEND, True)
    Else
        'Open new log
        Set CreateLogFile = FileSystem.CreateTextFile(strFileName, True)
    End If

    Set FileSystem = Nothing

    If Err.Number <> 0 Then
```

```
        Call WScript.StdErr.WriteLine ("Error creating the log file " & strFileName)
        Call WScript.StdErr.WriteLine ("Error " & Err.Number & " - " & Err.Description)
        Call WScript.Quit (1)
    End If

    On Error Goto 0
End Function ' CreateLogFile

'*********************************************************************
'*
'* Function SetProvider()
'*
'* Purpose:
'*
'* Input:  ProviderString
'*         HashAlg
'*         KeyLen
'*
'*********************************************************************
Function SetProvider(ByRef oCASetup, ByVal ProviderString, ByVal HashAlg, ByVal KeyLen)
    Call OutputLine(ECHOMINIMAL, _
    "SetProvider called with " & _
    Chr(34) & ProviderString & Chr(34) & ", " & _
    Chr(34) & HashAlg & Chr(34) & ", " & _
    Chr(34) & KeyLen & Chr(34))

    'Declare variable to store KeyInfo object
    Dim oCAKeyInfo
    Dim retVal

    retVal = False

    Call OutputLine(ECHOMINIMAL, "SetProvider: Creating oCAKeyInfo by calling
oCASetup.GetCASetupProperty(SETUPPROP_CAKEYINFORMATION )")
```

```
    ' Create CA KeyInfo object
    Set oCAKeyInfo = oCASetup.GetCASetupProperty(SETUPPROP_CAKEYINFORMATION)

    If ("" <> ProviderString) Then
        Call OutputLine(ECHOMINIMAL, "SetProvider: Changing oCAKeyInfo.ProviderName to "
& ProviderString)
        oCAKeyInfo.ProviderName = ProviderString
    End If

    ' Only modify key length if it was specified

    If ("" <> KeyLen) Then
        Call OutputLine(ECHOMINIMAL, "SetProvider: Changing oCAKeyInfo.Length to " &
KeyLen)
        oCAKeyInfo.Length = KeyLen
    End If

    ' Only modify hash algorithm if it was specified

    If ("" <> HashAlg) Then
        Call OutputLine(ECHOMINIMAL, "SetProvider: Changing oCAKeyInfo.HashAlgorithm to "
& HashAlg)
        oCAKeyInfo.HashAlgorithm = HashAlg
    End If

    Call OutputLine(ECHOMINIMAL, "SetProvider: Calling
oCASetup.SetCASetupProperty(SETUPPROP_CAKEYINFORMATION, oCAKeyInfo) ")

    On Error Resume Next
    Call Err.Clear()

    ' Set the keyInfo property
    Call oCASetup.SetCASetupProperty(SETUPPROP_CAKEYINFORMATION, oCAKeyInfo)
```

```
    If (Err.Number <> 0) Then
        Call OutputLine(ECHOMINIMAL, "SetProvider1: Error " & Err.Number & ": " &
Err.Description)
        Call OutputLine(ECHOMINIMAL, "Error Source: " & Err.Source)
        'Exit the script
        Call WScript.Quit (1)
    End If ' error occurred

    SetProvider = True
End Function 'SetProvider

'*********************************************************************
'*
'* Function InstallAndVerifyCA()
'*
'* Purpose: Run setup on CA object with specified parameters
'*
'* Input:  CAType
'*         CAService
'*         WebPages
'*
'*********************************************************************'
Function InstallAndVerifyCA(ByVal CAType, ByVal CAService, ByVal WebPages)
    Dim LocalCAConfig
    Dim CADBPath

    ' Default to failed
    InstallAndVerifyCA = False

    On Error Resume Next

    Call Err.Clear()

    Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: InitializeDefaults")
```

```
    Call OutputLine(ECHOMINIMAL, "CAService: " & CAService)
    Call OutputLine(ECHOMINIMAL, "WebPages: " & WebPages)

    Err.Number = 0

    ' Call this function with an error handling wrapper, or VBScript equivalent
    Call g_oCASetup.InitializeDefaults(CAService, WebPages)

    If (0 <> Err.Number) Then

        If (5 = Err.Number) Then
            ' uninstall

            If(bRecursed          = False) Then
                bRecursed          = True
                Call UninstallCA(False)
                InstallAndVerifyCA = InstallAndVerifyCA( CAType, CAService, WebPages)
                Exit Function
            End If

        End If 'error is already installed

        Call PrintErrorInfo("CA already installed and cannot uninstall", Err)
        Call OutputLine(ECHOMINIMAL, "")
        Exit Function 'InstallAndVerifyCA
    End If 'error occurred

'CA Service setup section
If (CAService = True) then
'Specify CA role
Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: SetCASetupProperty - CAType = " &
CAType)
Call g_oCASetup.SetCASetupProperty(SETUPPROP_CATYPE,  CAType)
```

```
If (0 <> Err.Number) And (VALUE_OUT_OF_RANGE <> Err.Number) Then
Call PrintErrorInfo("InstallAndVerifyCA3:unable to set SETUPPROP_CATYPE!", Err)
Exit Function 'InstallAndVerifyCA
End If 'not a domain admin and error occurred


If (VALUE_OUT_OF_RANGE = Err.Number) Then
Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: Error! Must be a domain administrator
to create an enterprise CA")
Exit Function 'InstallAndVerifyCA
End If ' not a domain admin

Call Err.Clear()

if (bInteractive <> FALSE) then
Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: SetCASetupProperty - Interactive = " &
bInteractive)
Call g_oCASetup.SetCASetupProperty(SETUPPROP_INTERACTIVE,  bInteractive)

If (0 <> Err.Number) Then
Call PrintErrorInfo("InstallAndVerifyCA:unable to set Interactive!", Err)
Call OutputLine(ECHOMINIMAL, "")
Exit Function 'InstallAndVerifyCA
End If
end if

If (False <> bReuseKey) Or (False <> bReuseCert) Then

If (False = SetupKeyReuse(bReuseKey, bReuseCert, strCAName)) Then
Call PrintErrorInfo("InstallAndVerifyCA: SetupKeyReuse failed.", Err)
Exit Function
End If

Else
```

```
If "" <> strCAName then
Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: SetCADistinguishedName")
'CAName, ignore UTF8, overwrite existing key, overwrite CA in DS
Dim strCAFullDN
strCAFullDN = "CN=" & strCAName
If "" <> strDNSuffix then strCAFullDN = strCAFullDN & "," & strDNSuffix

Call g_oCASetup.SetCADistinguishedName(strCAFullDN, True, True, True)
'Display errors

If (g_oCASetup.CAErrorId <> 0) Then
Call PrintErrorInfo("InstallAndVerifyCA:SetCADistinguishedName failed. ", Err)
End If

End If

End If

Call Err.Clear()

If (CAType <> ENTERPRISE_ROOTCA) And (CAType <> STANDALONE_ROOTCA) And (bReuseCert <>
True) Then
If (strRequestFile = "") Then
Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: SetParentCAInformation")
'Set parent CA information if it is a subordinate
Call g_oCASetup.SetParentCAInformation(strRootCAName)

If (0 <> Err.Number) And (ROOT_CA_NOT_FOUND <> Err.Number) Then
Call PrintErrorInfo("InstallAndVerifyCA:unable to set ParentCAInformation!", Err)
Call OutputLine(ECHOMINIMAL, "")
Exit Function 'InstallAndVerifyCA
End If ' root ca not found

If (ROOT_CA_NOT_FOUND = Err.Number) Then
```

```
Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: Root CA  could not be found!")
Exit Function 'InstallAndVerifyCA
End If ' root ca not found
Else
Call g_oCASetup.SetCASetupProperty(SETUPPROP_REQUESTFILE, strRequestFile)
End If
End If ' not root

If (bReuseCert = False) Then
Dim bProviderSet
bProviderSet = SetProvider(g_oCASetup, strSelectedCSP, strSelectedHashAlg,
iSelectedKeySize)

If (False = bProviderSet) Then
Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA:unable to set key properties!")
Exit Function 'InstallAndVerifyCA
End If 'error occurred
End If

Call Err.Clear()
End If

If (True = WebPages) And (CAType = NO_INSTALL_CA) Then
        Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: SetWebCAInformation")
        'Set web CA information if it is a web pages
        Call g_oCASetup.SetWebCAInformation(strRootCAName)

        If (0 <> Err.Number) Then

            If ( REG_VALUE_NOT_FOUND <> Err.Number) Then
                Call PrintErrorInfo("InstallAndVerifyCA:unable to set
SetWebCAInformation!", Err)
                Call OutputLine(ECHOMINIMAL, "")
            Else
```

```
                Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: install failed,
registry key not present!")
            End If

            Exit Function 'InstallAndVerifyCA
        End If ' error

    End If ' web pages should be installed

    Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: Setting Key Properties")

    Call Err.Clear()

    Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: calling SetProvider")

    'Dim KeyLenVar
    'KeyLenVar = ProviderKeyLength(strSelectedCSP)

    'If ("" <> KeyLenVar) Then
    '  iSelectedKeySize = KeyLenVar
    'End If

    Call Err.Clear()

    Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: BeforeInstall!")

    Call g_oCASetup.Install()

    If (0 <> Err.Number) Then

        If ( REG_VALUE_NOT_FOUND <> Err.Number) Then
            Call PrintErrorInfo("InstallAndVerifyCA:Install failed!", Err)
            Call OutputLine(ECHOMINIMAL, "")
        Else
```

```
            Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: install failed, registry
key not present!")
            Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: " & Err.Number & ": " &
Err.Description)
        End If

        Exit Function 'InstallAndVerifyCA
    End If 'error occurred

    Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: AfterInstall!")

    On Error GoTo 0

    Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: calling GetLocalCAConfig")

    LocalCAConfig = GetLocalCAConfig()

    If (LocalCAConfig = "") Then
        Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: CA registry entry not
present!")
        Exit Function 'InstallAndVerifyCA
    End If ' getlocalcaconfig failed

    WScript.Sleep 30000

    If (CAService = True) Then

        If (0 <> PingCA(LocalCAConfig)) Then
            Call OutputLine(ECHOMINIMAL, "InstallAndVerifyCA: Service not started!")
            Exit Function 'InstallAndVerifyCA
        End If ' can't ping service

    End If ' ca set to install as a service
```

```
    InstallAndVerifyCA = True
End Function 'InstallAndVerifyCA

'*********************************************************************
'*
'* Function UninstallCA()
'*
'* Purpose: Uninstall all of the CA server components or optionally just the pages
'*
'* Input:
'*
'*********************************************************************'
Function UninstallCA(ByVal WebPagesOnly)
    Dim LocalCAConfig

    Call OutputLine(ECHOMINIMAL, "UninstallCA: calling GetLocalCAConfig")

    ' Get the current location of the server
    LocalCAConfig = GetLocalCAConfig()

if (WebPagesOnly = False) Then
If ("" = LocalCAConfig) Then
Call OutputLine(ECHOMINIMAL, "UninstallCA: CA not installed!")
UninstallCA = True
Exit Function 'UninstallCA
End If ' getlocalcaconfig failed
End If

    Call OutputLine(ECHOMINIMAL, "UninstallCA: calling .PreUninstall")

    ' Clean up the web pages
    On Error Resume Next
    Call g_oCASetup.PreUninstall(WebPagesOnly)
```

```
    If Err.Number <> 0 Then
        Call PrintErrorInfo("UninstallCA: ", Err)
    End If

    Call OutputLine(ECHOMINIMAL, "UninstallCA: calling .PostUninstall")

    Call g_oCASetup.PostUninstall()

    Call OutputLine(ECHOMINIMAL, "UninstallCA: calling .GetLocalCAConfig")

    ' Check registry to see if CA is still installed
    LocalCAConfig = GetLocalCAConfig()

    If ("" = LocalCAConfig) Then
        'Not installed!
        Call OutputLine(ECHOMINIMAL, "UninstallCA: Uninstall completed successfully!")
        UninstallCA = True
        Exit Function 'UninstallCA
    End If 'getlocalcaconfig failed

    Call OutputLine(ECHOMINIMAL, "UninstallCA: calling PingCA")



    ' Try pinging the CA

    If (0 <> PingCA("")) Then
        UninstallCA = True
        Exit Function 'UninstallCA
    End If ' can't ping service

    ' Default to error
    UninstallCA = False
End Function 'UninstallCA
```

```
'************************************************************************
'*
'* Function GetLocalCAConfig()
'*
'* Purpose: Determine role of CA if installed
'*
'* Input:
'*
'************************************************************************'
Function GetLocalCAConfig()
    Dim WshShell
    Dim ActiveConfig
    Dim CAName
    Dim CAServer

    On Error Resume Next

    Set WshShell = WScript.CreateObject("WScript.Shell")
    ActiveConfig =
WshShell.RegRead("HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configurat
ion\Active")

    If (Err.Number <> 0) Then

        If (REG_VALUE_NOT_FOUND <> Err.Number) Then
            GetLocalCAConfig = ""
            Call PrintErrorInfo("GetLocalCAConfig: ", Err)
            Exit Function 'GetLocalCAConfig
        Else ' reg value not found
            GetLocalCAConfig = ""
            Call OutputLine(ECHOMINIMAL, "GetLocalCAConfig: CA Not Installed!")
            Call OutputLine(ECHOMINIMAL, "")
            Exit Function 'GetLocalCAConfig
        End If ' reg value found
```

```
    End If ' error occurred

    Call OutputLine(ECHOMINIMAL," Reading
HKLM\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\" & ActiveConfig &
"\CommonName")
    CAName =
WshShell.RegRead("HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configurat
ion\" & ActiveConfig & "\CommonName")
    Call OutputLine(ECHOMINIMAL, "CAName: " & CAName)

    Call OutputLine(ECHOMINIMAL," Reading
HKLM\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\" & ActiveConfig &
"\CAServerName")
    CAServer =
WshShell.RegRead("HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configurat
ion\" & ActiveConfig & "\CAServerName")
    Call OutputLine(ECHOMINIMAL, "CAServer: " & CAServer)

    ' Cleanup
    Set WShShell = Nothing

    'Set Return value
    GetLocalCAConfig = CAServer & "\" & CAName
End Function 'GetLocalCAConfig

'*********************************************************************
'*
'* Function PingCA()
'*
'* Purpose: Use CertUtil to ping the CA
'*
'* Input:
'*
'*********************************************************************'
```

```
Function PingCA(ByVal CAConfig)
    Dim WshShell
    Dim command
    Dim RunRet

    Set WshShell = WScript.CreateObject("WScript.Shell")

    If ("" <> CAConfig) Then
        command = "certutil -config " & CAConfig & " -ping"
    Else 'caconfig param null
        command = "certutil -ping"
    End If ' caconfig param passed

    RunRet       = WshShell.Run(command, 1, False)

    Set WShShell = Nothing
    PingCA       = RunRet
End Function ' PingCA

'*********************************************************************
'*
'* Function SetUpKeyReuse()
'*
'* Purpose:
'*
'* Input:
'*
'*********************************************************************'
Function SetUpKeyReuse(ByVal bReuseKey, ByVal bReuseCert, ByVal KeyName)

    Dim oCAKeyInfo
    Dim oExistingCerts
    Dim CertInfo
```

```
    On Error Resume Next

    Set oCAKeyInfo     = g_oCASetup.GetCASetupProperty(SETUPPROP_CAKEYINFORMATION)
    Set oExistingCerts = g_oCASetup.GetExistingCACertificates()

    Call OutputLine(ECHOMINIMAL,"Searching Existing Machine Keys")

    For Each CertInfo in oExistingCerts
        wscript.echo "Existing Cert: " & certinfo.ContainerName

        If (KeyName = certinfo.ContainerName) Then
            wscript.echo "Found cert!"
            oCAKeyInfo.Existing      = True
            If (Err.Number <> 0) Then Call PrintErrorInfo("SetUpKeyReuse:
oCAKeyInfo.Existing", Err)
            oCAKeyInfo.ContainerName = CertInfo.ContainerName
            If (Err.Number <> 0) Then Call PrintErrorInfo("SetUpKeyReuse:
oCAKeyInfo.ContainerName", Err)
            oCAKeyInfo.HashAlgorithm = CertInfo.HashAlgorithm
            If (Err.Number <> 0) Then Call PrintErrorInfo("SetUpKeyReuse:
oCAKeyInfo.HashAlgorithm", Err)
            oCAKeyInfo.Length        = CertInfo.Length
            If (Err.Number <> 0) Then Call PrintErrorInfo("SetUpKeyReuse:
oCAKeyInfo.Length", Err)
            oCAKeyInfo.ProviderName  = CertInfo.ProviderName
            If (Err.Number <> 0) Then Call PrintErrorInfo("SetUpKeyReuse:
oCAKeyInfo.ProviderName", Err)

            If (bReuseCert = True) Then
                oCAKeyInfo.ExistingCACertificate = CertInfo.ExistingCACertificate
                If (Err.Number <> 0) Then Call PrintErrorInfo("SetUpKeyReuse:
oCAKeyInfo.ExistingCACertificate", Err)
            End If

            Call g_oCASetup.SetCASetupProperty(SETUPPROP_CAKEYINFORMATION, oCAKeyInfo)
```

```
            If (Err.Number <> 0) Then Call PrintErrorInfo("SetUpKeyReuse:
g_oCASetup.SetCASetupProperty(1, oCAKeyInfo)", Err)
            wscript.echo g_oCASetup.GetCASetupProperty(SETUPPROP_CANAME)
            wscript.echo g_oCASetup.GetCASetupProperty(SETUPPROP_CADSSUFFIX)
        End If

    Next

    SetupKeyReuse = True

End Function ' SetKeyReuse
```

See also

- Active Directory Certificate Services Migration Guide
- AD CS Migration: Preparing to Migrate
- AD CS Migration: Migrating the Certification Authority
- AD CS Migration: Verifying the Migration
- AD CS Migration: Post-Migration Tasks