


Microsoft
Windows Server 2003

Operating System

Understanding How Domain Rename Works

Microsoft Corporation

Published: April 2003

Abstract

This document provides the rationale and technical background for understanding the effects of a domain rename operation in a Windows Server 2003 forest.

For the preparation instructions and step-by-step procedures for performing a domain rename operation in your enterprise, see "Step-by-Step Guide to Implementing Domain Rename."

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. **MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.**

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2003 Microsoft Corporation. All rights reserved.

Microsoft, Windows Server 2003 Datacenter Server, Windows Server 2003 Enterprise Server, and Windows Server 2003 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Table of Contents

Table of Contents	3
Introduction to Domain Rename.....	5
Constraints to Restructuring Domains in a Windows 2000 Forest	5
Constraints to Restructuring Domains in a Windows Server 2003 Forest.....	6
When to Use Domain Rename	6
Simple Rename without Repositioning	7
Rename with Repositioning in the Same Tree.....	8
Rename with Creation of a New Tree Root	10
Rename with Repositioning to a Different Tree	12
Reusing a Domain Name	13
Rules, Conditions, and Requirements for Performing the Domain Rename Operation	14
Rules for a Well-Formed Forest.....	14
Domain Rename Conditions and Effects on Service	15
How Domain Rename Works.....	16
Overview of the Domain Rename Process	16
The Domain Rename Tool.....	16
The Domain Rename State File	17
Domain Controller States	17
General Steps in the Domain Rename Process.....	17
How Specifying the Target Forest Structure Works.....	18
Current Domain Names – Generating the Forest Description File.....	18
Target Domain Names – Editing the Forest Description File.....	20
How Domain Rename Instructions are Transferred to Active Directory	21
Domain Rename Script and State File	21
How DCs are Prepared for Domain Rename	22
Establishing a DNS Alias for a New Domain Name	22
Establishing Service Principal Names for a New Domain Name	25
Actions Performed by Rendom in Response to the /upload Command.....	25
How DC Readiness is Verified	27
Actions Performed in Response to the /prepare Command.....	27
How Domain Rename Instructions are Executed.....	28
Single-User Mode.....	28
Update Transaction	28

Actions Performed in Response to the /execute Command	29
Determining Domain Rename Completion	30
How Group Policy is Reconciled Following Domain Rename.....	30
How Old Domain Names are Removed Following Domain Rename.....	31

Introduction to Domain Rename

Microsoft® Windows® .Server 2003 Standard Edition, Microsoft® Windows® Server 2003 Enterprise Edition, and Microsoft® Windows® Server 2003 Datacenter Edition provide the capability to rename domains in an Active Directory forest after the forest structure is in place. This functionality is not available in Microsoft® Windows® 2000 Server family. The structure of an Active Directory forest is the result of the order in which you create domains and the hierarchical names of those domains. Beginning with the forest root domain, all child domains derive their distinguished names and default DNS names from the forest root domain name. The same is true of every additional tree in the forest. The way to change the hierarchical structure of an existing domain tree is to rename the domains. For example, you can rename a child domain to have a different parent, or rename a child domain to be a new tree-root domain. In each case, you reposition an existing domain to create a different domain-tree structure. Alternatively, you can rename domains without affecting the structure. For example, if you rename a root domain, the names of all child domains below it are also changed, but you have not created a different domain-tree structure.

In Windows Server 2003, the goal of the domain rename functionality is to ensure a supported method to rename domains when necessary; *it is not intended to make domain rename a routine operation*. Thus, although renaming domains is possible in Windows Server 2003, the process is complex and should not be undertaken lightly.

Constraints to Restructuring Domains in a Windows 2000 Forest

The restructuring capabilities in a Windows Server 2003 forest provide solutions to some of the problems that are not addressed in Windows 2000 Server family. In a Windows 2000 forest, renaming domains is essentially not possible after the forest structure is in place without moving domain contents or recreating them. The constraints associated with making domain name changes or domain-tree restructuring in Windows 2000 Active Directory are prohibitive.

In a Windows 2000 forest, you *cannot*:

- Change the DNS name or the NetBIOS name of a domain. Although you cannot rename a domain, you can achieve the same results by moving its contents into a new domain that has the name you want the existing domain to have. (Active Directory Object Manager (MoveTree) in the Windows 2000 Server family Support Tools can be used to move directory objects between domains.)
- Move a domain within a forest in a single operation. As above, you can clone items in and move items from a domain, but you cannot move the entire domain itself within a forest.
- Split a domain into two domains in a single operation. To split a domain, you must create a new domain and then move appropriate users and resources from the existing domain into the new domain.

- Merge two domains into a single domain in a single operation. To merge domains, you must move all the contents from one of the domains into the other and then demote all domain controllers in the empty domain and decommission it.

Thus, in a Windows 2000 forest, significant administrative overhead is associated with performing such manual move operations to achieve the domain-tree restructuring or renaming one or more domains.

Constraints to Restructuring Domains in a Windows Server 2003 Forest

Windows Server 2003 Standard Edition, Windows Server 2003 Enterprise Edition, and Windows Server 2003 Datacenter Edition provide tools with which you can safely rename domains to restructure a Windows Server 2003 forest. When making a decision about whether to restructure an existing Windows Server 2003 forest, be sure to consider what you *cannot* do with forest restructuring. Although a Windows Server 2003 forest has forest restructuring capability, certain types of structural changes *are not supported*.

In a Windows Server 2003 forest, you *cannot*:

- Change which domain is the forest root domain. Changing the DNS or the NetBIOS name of the forest root domain, or both, is supported.
- Drop domains from the forest or add domains to the forest. The number of domains in the forest before and after the rename/restructure operation must remain the same.
- Rename a domain with the same name that another domain gave up in a single forest restructure operation.

When to Use Domain Rename

The ability to rename a domain provides you with the flexibility to make important name changes and forest structural changes as the needs of your organization change. Using domain rename, you can change not only the name of a domain, but you can change the structure of the domain hierarchy such that the parent of a domain can be changed or a domain residing in one domain tree can be moved to another domain tree. The domain rename functionality can accommodate situations involving company acquisitions, mergers, or name changes; but it is not designed to accommodate forest mergers or moving domains between forests.

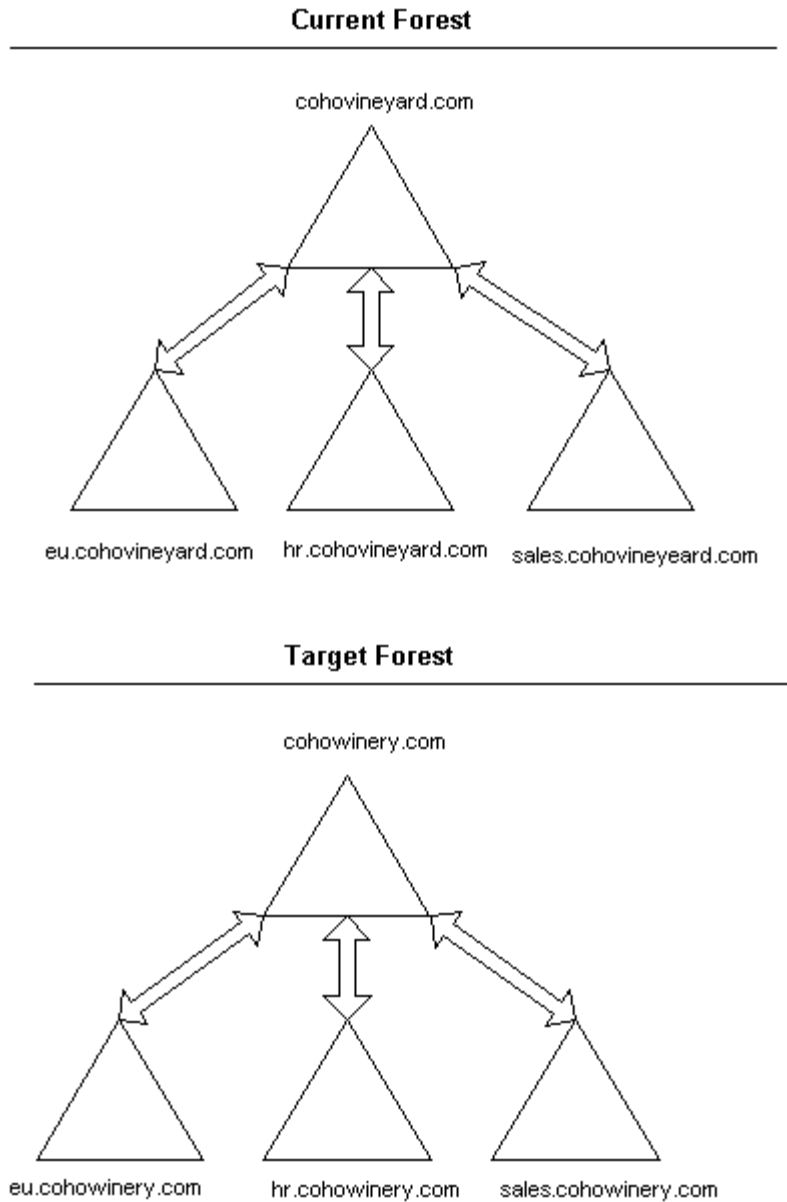
By using the domain rename operation, you can make several kinds of changes to an existing Windows Server 2003 forest, including:

- Simple rename without repositioning any domains in the forest structure.
- Create a new domain-tree structure by repositioning domains within a tree.
- Create new trees.

Simple Rename without Repositioning

You can rename domains without restructuring the forest in terms of the parent-child relationships between existing domains. For example, the existing `cohovineyard.com` forest has four domains — `cohovineyard.com` (root), `eu.cohovineyard.com`, `hr.cohovineyard.com`, and `sales.cohovineyard.com`. Now suppose that the company decides to expand into wine bottling and distribution and needs to change the name from Coho Vineyard to Coho Winery, requiring Active Directory domain names to reflect the new company name. As shown in Figure 1, the target forest still has four domains, with the following names: `cohowinery.com` (root), `eu.cohowinery.com`, `hr.cohowinery.com`, and `sales.cohowinery.com`. By renaming the forest root domain, you create the condition where you must rename all child domains in the tree to preserve the original forest structure, as shown in Figure 1.

Figure 1 Domain rename of four domains without repositioning domains

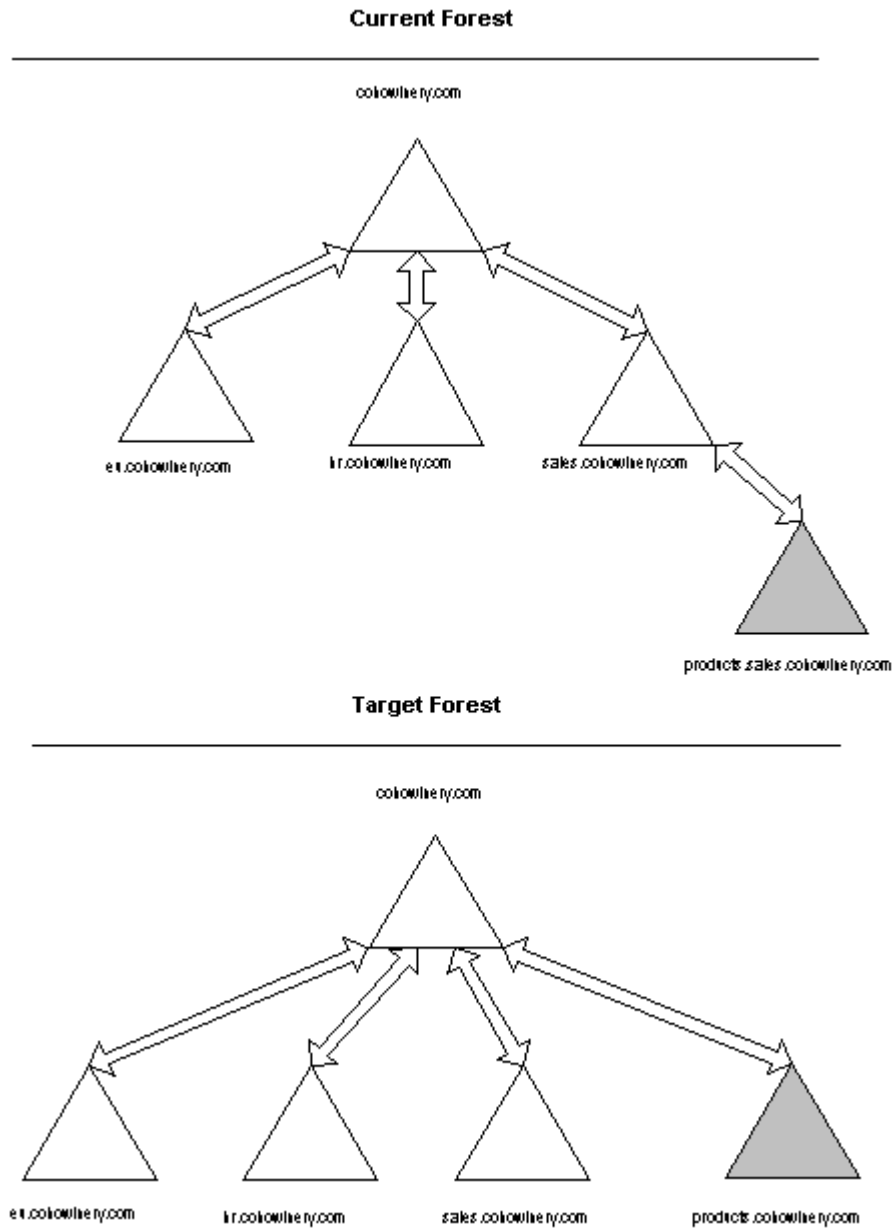


Rename with Repositioning in the Same Tree

You can change the structure of the domain tree by renaming a child domain to appear in a different location in the tree. For example, in the cohowinery.com forest, the products.sales.cohowinery.com domain is currently a child of the sales.cohowinery.com domain,

placing it two levels below the forest root domain. If internal reorganization results in the products division no longer being a subdivision of the sales organization, the company might want to change the domain structure to put the products organization at the same level as the sales organization. Figure 2 shows how changing the parent of products.sales.cohowinery.com results in a restructured domain tree.

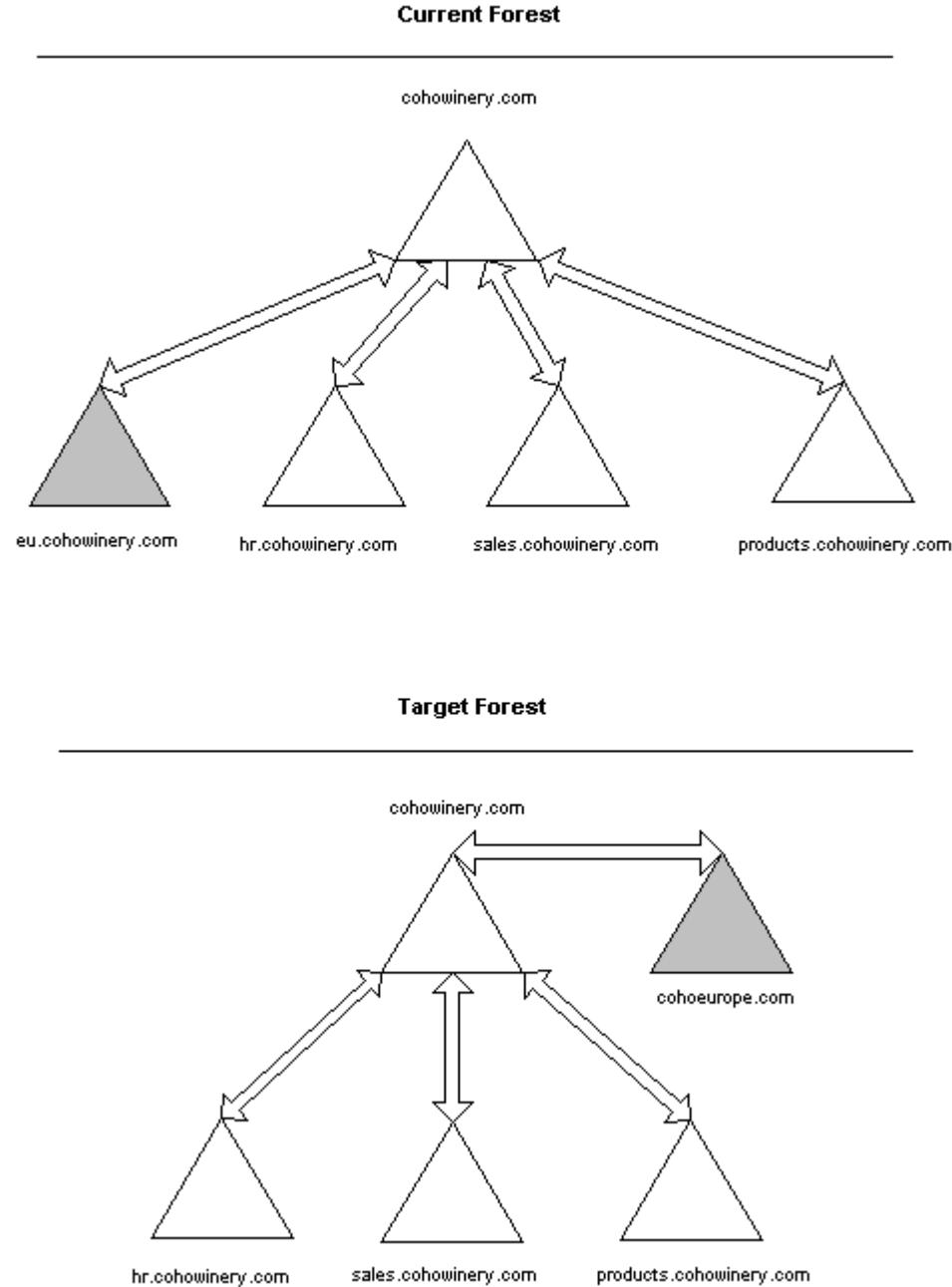
Figure 2 Domain rename to change the parent of a child domain



Rename with Creation of a New Tree Root

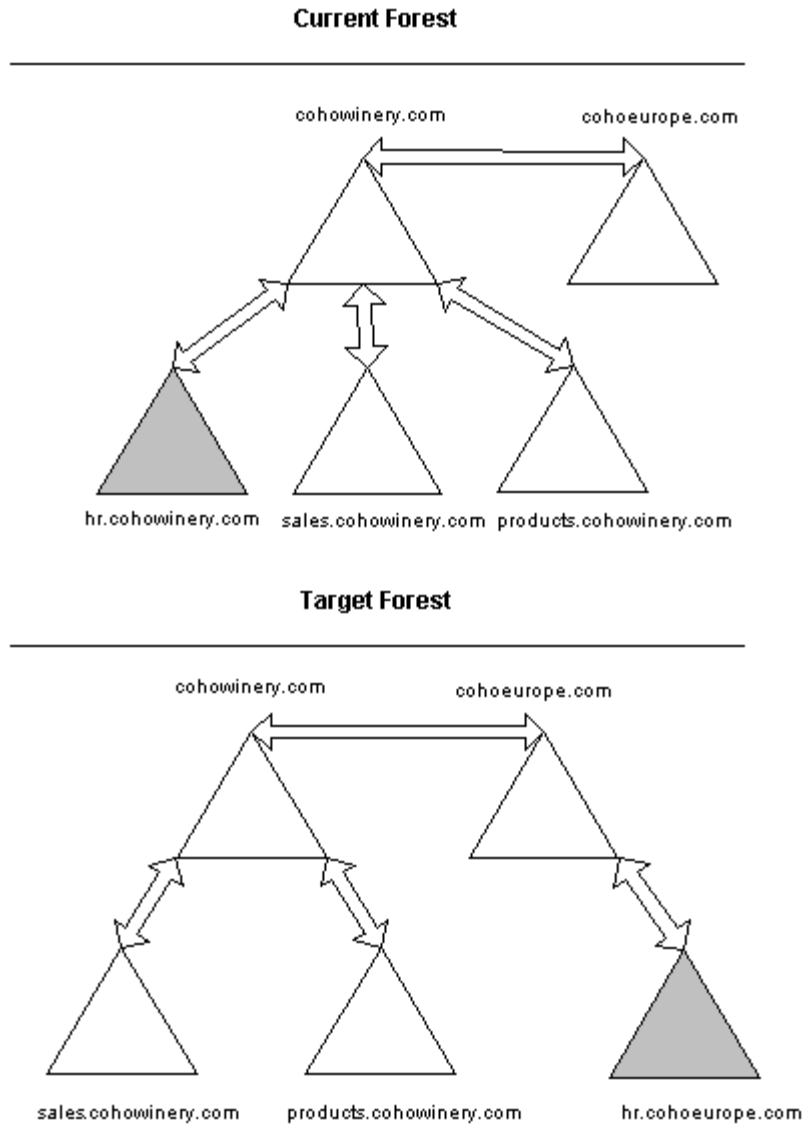
Restructuring a forest allows you to move a domain (except the forest root domain) anywhere within the forest in which it resides, including the ability to move a domain so that it becomes the root of its own domain tree. For example, in the cohowinery.com forest, the European branch of the organization, named eu.cohowinery.com, is a child of the forest root domain. Management within the organization has determined that the European division's internal domain name should better reflect its Internet DNS name, cohoeurope.com. In the desired forest structure shown in Figure 3, the company would like to move and rename the eu.cohowinery.com domain so that it becomes its own tree-root domain named cohoeurope.com.

Figure 3 Domain rename to create a new tree root



Rename with Repositioning to a Different Tree

By renaming domains, you can effectively move a child domain to a different parent, even if the parent is in a different tree. For example, in the current forest structure, the human resources (hr) domain is a child of cohowinery.com. This domain has domain controllers in the United States. However, changes in your organization have prompted the need for the human resources department to move its location to Europe. In your desired forest structure, you would like to move the hr.cohowinery.com domain so that it becomes a child of the domain cohoeurope.com, residing in another domain tree. As shown in Figure 4, to accomplish this relocation, you rename the hr.cohowinery.com domain to hr.cohoeurope.com.

Figure 4 Domain rename to move a domain to a different tree

Reusing a Domain Name

As described earlier in "Constraints to Restructuring Domains in a Windows Server 2003 Forest," the domain rename operation cannot rename two or more domains such that one domain gives up its name and another domain assumes the same name in a single forest restructuring operation. For example, using the Current Forest configuration in Figure 4, earlier in this document, you cannot restructure the current forest so that the cohoeurope.com domain is named

something else and the hr.cohowinery.com domain assumes the name cohoeurope.com in a single restructuring operation.

However, you can accomplish the desired result by first running the domain rename procedure to rename the cohoeurope.com domain. When you are absolutely sure that the first rename process is completed, you can then perform the domain rename procedure again so that hr.cohowinery.com assumes the domain name cohoeurope.com.

Rules, Conditions, and Requirements for Performing the Domain Rename Operation

It is extremely important to remember that the goal of the domain rename functionality provided for a Windows Server 2003 forest is to ensure that there is a *supported* method to rename/restructure domains in a deployed Active Directory forest when necessary (within the constraints described earlier). The intent is not to make domain rename a *routine* operation. The underlying reason for this caution is that the domain rename procedure is complex and requires a great deal of care in planning and execution. Further, the time required to go through a complete domain rename operation is directly proportional to the size of the deployed Active Directory forest in terms of the number of domains, domain controllers, and member computers.

Rules for a Well-Formed Forest

The forest restructuring capability in a Windows Server 2003 forest supports any set of changes to the DNS and NetBIOS names of the domains of a forest such that the resulting forest is well-formed.

In a well-formed forest, the following conditions must be true.

- The DNS names of the comprised domains form one or more trees.
- The forest root domain is the root of one of these trees.
- An application directory partition cannot have a domain directory partition as a child.

Domain Rename Conditions and Effects on Service

Before undertaking a domain rename operation, it is imperative that you fully understand the following conditions and effects that are inherent in the process and that you are willing and able to fully accommodate them:

- Domain rename operation is not supported in an Active Directory forest that has Exchange 2000 deployed in it. The domain rename tool will detect this fact and refuse to proceed.
- Headless management is helpful. Each domain controller in the forest will be individually contacted to put the required changes corresponding to the domain rename into effect; the update will *not* spread across the forest through Active Directory replication. This condition *does not* imply that each domain controller in the forest has to be visited physically by an administrator. However, if you want to rename a domain in a large forest, it is highly recommended that you implement headless management of the domain controllers in the forest. In the event that some domain controllers are found to be unresponsive during the domain rename procedure, headless management will greatly improve your ability to troubleshoot the problem.
- The forest will be out of service for a short period of time. Forest service is interrupted during the time it takes for each domain controller to perform the directory database updates necessary for the domain rename and then reboot. The time period is proportional to the number of domain controllers in the forest and is preferable to the alternative of having the forest in odd "in-between" states for a much longer period of time just to avoid the relatively short service interruption.
- All domain controllers must either successfully complete the rename operation or be eliminated from the forest. The domain rename will take effect even if it proves impossible to update some domain controllers in the forest. For the domain rename operation to be deemed complete, *every* domain controller in the forest needs to be contacted and updated. If you choose to declare your domain rename operation complete without having updated some number of domain controllers because they were impossible to contact, then you must remove all such uncontacted DCs from the forest.
- Each member computer joined to a renamed domain must be rebooted twice after all domain controllers have been updated. Machines running Windows NT 4.0 will need to *unjoin* and then *rejoin* the renamed domain instead of rebooting.
- If you want DNS host names of domain controllers to match the new domain name, you must perform domain controller rename procedures following domain rename. The DNS host names of the domain controllers are not changed automatically by the domain rename operation to reflect the new domain name. In other words, the primary DNS suffix of the domain controller will not match the new domain DNS name after the domain has been renamed. Having the host name of a domain controller decoupled from its domain name has no impact on forest service. However, renaming a domain controller requires a separate multi-step procedure after the domain rename operation is complete.

- The DNS suffix of member computer host names in a domain that is being renamed might not match the DNS name of the domain for a period of time. By default, the DNS suffix portion of the computer name is updated automatically when the domain to which the computer is joined changes (as it does when you rename a domain). In general, the period of time during which the DNS name of the domain does not match the DNS suffix of computer names is proportional to the number of machines in the domain. In some cases, you might want to keep the computer names from being updated automatically.

How Domain Rename Works

Before attempting a domain rename, you will benefit from a thorough understanding of what happens during the process so that you know what to expect. This document is organized so that you read everything about the procedure before actually following the instructions to perform the procedure. *Do not try to follow any steps until you have read this entire document.*

A domain rename will affect every domain controller (DC) in the forest. The procedure is a multi-step process that requires a general understanding of the actions that occur at each step as well as the updates made to the directory and their side effects. This section provides the details you will need to understand how the domain rename process works and what happens within Active Directory, DNS, and with integral features such as Group Policy and security.

Overview of the Domain Rename Process

The domain rename process involves making basic changes independently at each DC in a forest. You set up an administrative computer from which you can issue commands that are executed remotely at each DC in the forest. Through these commands, the directory database at each DC in the forest is updated individually to effect the necessary changes for renaming domains; that is, the updates that rename domains do not spread across the forest through Active Directory replication.

Domain rename is implemented in a monitored, step-by-step process that ensures that every DC in the forest completes its changes one step at a time — that is, the next step in the process cannot occur until the current step has been completed at every DC in the forest.

The Domain Rename Tool

Rendom.exe is the command-line utility for renaming domains in Windows Server 2003 forests. Rendom is used to carry out the multiple steps in the domain rename procedure. You precede the domain rename process by using Rendom to prepare a list of domains in the forest. You begin the domain rename process by using Rendom to generate a script that contains the instructions for renaming domains in the forest. You use Rendom again to verify that all DCs are adequately prepared to make the necessary updates to rename the domains. Finally, you use Rendom to execute the actual domain rename instructions on every DC. Following the domain rename

procedure, you use Rendom to remove all metadata written to the directory by the domain rename operation.

The Domain Rename State File

As a result of the first command you issue to begin the domain rename process, Rendom generates an XML-structured text file called a *state file*, which contains the list of all DCs in the forest. As DCs progress through the various steps in the procedure, Rendom updates the state file to track the state of each DC relative to the completion of the domain rename process.

As you perform each step in the domain rename operation, Rendom automatically updates the state file. By monitoring the states of completion of each DC in the state file, you receive the information you need to issue the next Rendom command in the sequence.

Domain Controller States

Rendom records four states of completion for each DC in the state file.

- *Initial*: Each DC that is reachable during the rename procedure starts out from the *Initial* state.
- *Prepared*: When the domain rename instructions written by Rendom have been verified by a DC independently, it advances to the *Prepared* state.
- *Final*: From the Prepared state, a DC advances to one of two final states. The domain rename process stops when every DC in the forest has reached either of the following states:
 - *Done*, signifying that the domain rename is complete at that DC.
 - *Error*, implying that some irrecoverable error has occurred and further progress on the domain rename is deemed impossible at that DC.

The steps in the domain rename procedure that attempt to take a DC from the Initial state to the Prepared state and from the Prepared state to a final state can be executed only after *every* DC in the forest has reached the required state. A step can be executed multiple times for any DCs that are not reachable in an initial attempt. Each such additional execution of the same step attempts to contact only those DCs that have not achieved the required state.

For more information about the contents of the state file, see "Domain Rename Script and State File" later in this document.

General Steps in the Domain Rename Process

The following set of steps is a very high-level representation of what happens during the domain rename process. A more detailed discussion of each step is provided in subsequent topics, beginning with the topic "How Specifying the Target Forest Structure Works." The steps described in this topic are not provided as a procedure that you can follow, but rather as a discussion to give you an idea of how the operation works. *Do not try to follow these steps.* A

fully documented set of procedures, including all the tasks you will perform, is provided in a separate document titled "Understanding and Implementing Domain Rename."

The general steps in the domain rename procedure can be summarized as follows:

1. Specify the new forest structure represented by the set of changed domain names in the forest.
2. Generate domain rename instructions encoded as a special script based on the specified new forest structure and transfer it to every DC in the forest.
3. Verify the validity of the domain rename instructions (script) at every DC and its readiness to execute those instructions.
4. Execute the domain rename instructions at every DC in the forest. This is the step at which a brief interruption in the forest service may occur.
5. Fix up Group Policy metadata in the directory so that policies can continue to be applied following the renaming of domains.
6. Clean up all domain rename related metadata written to the directory such that it is ready for another round of domain rename operation if needed.

How Specifying the Target Forest Structure Works

After you have carefully planned your domain rename, you begin the domain rename process by using `Random` to make a list of the domain names in the current forest structure, including application directory partition names. After `Random` creates this list, you create the new forest structure by editing the names in the list. In accordance with the DNS hierarchical naming scheme, the structure of the forest is implicit in the set of DNS names for the domains and application directory partitions that make up the forest. In addition to specifying DNS name changes for domains and application directory partitions, the NetBIOS name of any domain can also be changed. Changes to the DNS and NetBIOS names of the domains and the DNS names of the application directory partitions that constitute the forest are supported, subject to the constraints outlined in "Constraints to Restructuring Domains in a Windows Server 2003 Forest" earlier in this document.

Current Domain Names — Generating the Forest Description File

The `random /list` command generates the current forest description and writes it to an output file (default name *domainlist.xml*) using an XML-encoded structure. This file contains a list of all domains and application directory partitions in the forest, along with their corresponding DNS and NetBIOS names (application directory partitions do not have NetBIOS names). Each domain and application directory partition is also identified by a globally unique identifier (GUID), which does not change with a rename. To simplify specifying the new forest structure, `Random` gathers and compiles the current forest structure automatically such that the new forest structure can be overlaid on top of it.

Figure 5 shows an example forest description file generated for a forest that has three domains named cohovineyard.com (the forest root domain), sales.cohovineyard.com, and hr.sales.cohovineyard.com, as well as four application directory partitions named DomainDnsZones.hr.sales.cohovineyard.com, DomainDnsZones.sales.cohovineyard.com, DomainDnsZones.cohovineyard.com, and ForestDnsZones.cohovineyard.com that are used by the DNS service. In Figure 5, nonvariable values are designated in bold text.

Figure 5 Forest description file (domainlist.xml) from the `random /list` command

```
<Forest>
  <Domain>
    <!-- PartitionType:Application -->
    <GUID>78438a56-f4a7-383a-5c82-fe05a76ed464</GUID>
    <DNSname>DomainDnsZones.hr.sales.cohovineyard.com</DNSname>
    <NetBiosName></NetBiosName>
    <DcName></DcName>
  </Domain>
  <Domain>
    <GUID>89cf8ae3-f4a3-453b-ac5c-cb05a76bca40</GUID>
    <DNSname>hr.sales.cohovineyard.com</DNSname>
    <NetBiosName>HR</NetBiosName>
    <DcName></DcName>
  </Domain>
  <Domain>
    <!-- PartitionType:Application -->
    <GUID>b9748a56-c4a7-385a-5d84-7490e76ba484</GUID>
    <DNSname>DomainDnsZones.sales.cohovineyard.com</DNSname>
    <NetBiosName></NetBiosName>
    <DcName></DcName>
  </Domain>
  <Domain>
    <GUID>89238a56-f3a1-343b-bc5c-cb05a76bc341</GUID>
    <DNSname>sales.cohovineyard.com</DNSname>
    <NetBiosName>SALES</NetBiosName>
    <DcName></DcName>
  </Domain>
  <Domain>
    <!-- PartitionType:Application -->
    <GUID>ea658a56-f4a7-383a-5c82-cb05a76bdf35</GUID>
    <DNSname>DomainDnsZones.cohovineyard.com</DNSname>
    <NetBiosName></NetBiosName>
    <DcName></DcName>
  </Domain>
  <Domain>
    <!-- PartitionType:Application -->
    <GUID>ea658a56-f4a7-383a-5c82-cb05a76bd461</GUID>
    <DNSname>ForestDnsZones.cohovineyard.com</DNSname>
    <NetBiosName></NetBiosName>
    <DcName></DcName>
  </Domain>
</Forest>
```

```

<Domain>
  <! - ForestRoot -->
  <GUID>34fg8ae3-f4a3-453b-ac5c-3ce5a76bca42</GUID>
  <DNSname>cohovineyard.com</DNSname>
  <NetBiosName>COHOVINEYARD</NetBiosName>
  <DcName></DcName>
</Domain>
</Forest>

```

Target Domain Names – Editing the Forest Description File

The current forest description file that is generated by the **random /list** command is a text file that you can edit to express the target forest structure as new domain names. To express the new structure, you simply modify the **<DNSname>** and **<NetBiosName>** fields to contain the new names where needed.



Note

The domain GUID value in the **<GUID>** field represents a fixed name for a domain and cannot be modified. The GUID provides a permanent identity by which a renamed domain can be identified in the new forest structure.

For example, using the description file in Figure 5, if you wanted to change the DNS name of the hr.sales.cohovineyard.com domain to payroll.sales.cohovineyard.com and change its NetBIOS name from HR to PAYROLL, you would replace the variable values in the appropriate lines in the forest description file as follows (nonvariable values are designated in bold text):

- Current DNS and NetBIOS names:

```
<DNSname>hr.sales.cohovineyard.com</DNSname>
```

```
<NetBiosName>HR</NetBiosName>
```

- Modified DNS and NetBIOS names (the two lines above should be modified as follows):

```
<DNSname>payroll.sales.cohovineyard.com</DNSname>
```

```
<NetBiosName>PAYROLL</NetBiosName>
```

Based on the specified new forest structure, Random reads information for each domain from the directory and then generates a set of directory update instructions. By default, the information collected by Random will be retrieved from any available DC in each domain. However, you can optionally specify a particular DC in each domain from which to retrieve the information required to generate the domain rename update instructions. To use this option, simply add the DNS host name of the desired DC in the **<DcName></DcName>** field of the forest description file.

After editing the forest description file as described above, you can use the **random /showforest** command to display the new forest structure contained in the file (this command does not have

any impact on the directory itself). The user-friendly format uses indentations to reflect the domain hierarchy within the forest.

How Domain Rename Instructions are Transferred to Active Directory

After you create the new structure by editing the forest description file, the next step in the domain rename process involves translating the new forest structure into a sequence of directory update instructions, to be executed individually on each DC in the forest. This translation occurs when you issue the **rendom /upload** command, and the resultant domain rename instructions are uploaded to the configuration directory partition at the DC that currently is the domain naming master for the forest. The instructions are then replicated to all other DCs in the forest through normal replication of the configuration directory partition. Only after these rename instructions have replicated to *every* DC in the forest and the required conditions have been verified at each DC can each DC proceed with executing the rename instructions.

The following topics describe details of the changes produced by the /upload command. The actual sequence of actions that occur in response to issuing the command are described in "Actions Performed by Rendom in Response to the /upload Command" later in this document.

Domain Rename Script and State File

To transform the current forest structure into the target structure, Rendom translates the new forest description to a set of update instructions that the DC uses to update its replica directory partitions to make the name changes effective. The **rendom /upload** command generates the required domain rename instructions, which are encoded in a special script format that is private to the implementation. The upload command also generates the state file that is used to track the progress of the domain rename operation.



Note

The actions performed by the upload command and the resultant changes made to the directory are in preparation for the rename operation. No domain name changes occur at this step.

Domain Rename Script for Update Instructions

The script that is generated by the **rendom /upload** command is private to the domain rename procedure and does not create directory changes per se. Rather, it provides instructions for DCs to execute in response to Rendom commands. The script is an XML-encoded document that has three components:

- *Test*, a read transaction to validate that the directory replica at the DC is in an appropriate state to perform an action.

- *Action*, an update transaction to be performed on the directory database at the DC.
- *Signature*, a cryptographic hash that proves to the DC that the script was prepared by the Rendom utility (hence, it is authentic and correct) and not manually by someone acting as an administrator.

State File for Tracking

While translating the forest description to the update instruction script and transferring it to the directory, Rendom also generates the state file (default name *DClist.xml*). This file is structured as an XML-text document to facilitate tracking the state of each DC as it progresses through the various steps of the domain rename process. This file contains an entry for every DC in the forest. Each DC is identified by its DNS host name, along with fields for its current state and the last error encountered on the DC while processing the domain rename instructions.

How DCs are Prepared for Domain Rename

When you run the **rendom /upload** command, certain changes occur on the domain naming master in preparation for the actual execution of domain rename. On the domain naming master, the XML-encoded script containing the domain rename instructions is written to the single-valued, octet-string attribute *msDS-UpdateScript* on the Partitions container object (cn=partitions,cn=configuration,dc=*ForestRootDomain*) in the configuration directory partition. The Partitions container can be updated only on the DC that is the domain naming master for the forest, so the *msDS-UpdateScript* attribute is necessarily changed on the DC that holds the domain naming master role. From this source DC, the script stored in the *msDS-UpdateScript* attribute replicates to all DCs in the forest through normal replication of the configuration directory partition.

Establishing a DNS Alias for a New Domain Name

In addition to the *msDS-UpdateScript* value being written to the Partitions container, the new DNS name of each domain being renamed is also written by Rendom to the single-valued, Unicode-string attribute *msDS-DnsRootAlias* on the cross-reference object (class *crossRef*) corresponding to that domain. Again, because cross-reference objects are stored in the Partitions container and the Partitions container can be updated only on the domain naming master, the *msDS-DnsRootAlias* attribute can be changed only on the DC that holds the domain naming master role. From this source DC, the DNS name in the *msDS-DnsRootAlias* attribute replicates to all DCs in the forest through normal replication of the configuration directory partition.

DC Locator Mechanism

DNS is required for the location of DCs by Active Directory clients. DC Locator is an algorithm that runs in the context of the NetLogon service. It finds the best domain controller for a request, based on network proximity, by using the resource records that are registered by domain controllers in DNS. The NetLogon service on every Active Directory domain controller dynamically registers service (SRV) resource records in DNS at startup, which allow domain

controllers to be located by service type and protocol. Additionally, every domain controller also registers a set of A (host) resource records in DNS for use by LDAP clients that do not support DNS SRV resource records (DC Locator does not use these records), and a CNAME (alias) record for use by Active Directory replication.

According to the client request, the DC Locator can use various specified criteria that map to values in the SRV resource records to locate DCs with specific roles or capabilities. For example, DC Locator can find a domain controller that has a full, writable replica of a domain directory partition, a domain replica in a specified site, or a domain controller that is a global catalog server.

For information about DC Locator and the DNS resource records registered by an Active Directory domain controller at startup, see "Locating Domain Controllers" in the *Directory Services Guide* of the *Windows Server 2003 Family Resource Kit*.

Publishing Two Sets of Locator SRV Resource Records in DNS

The directory service is extended in Windows Server 2003 forests to use the *msDS-DnsRootAlias* attribute to support the concept of a *DNS alias* for a domain. Presence of this attribute prompts the NetLogon service running on the domain controller to register the DNS domain name value of this attribute in DNS.

Thus, with the addition of the *msDS-DnsRootAlias* attribute, NetLogon has the ability to register not one, but two domain names in DNS. The identities of the real (original) domain name and the alias (target) domain name are established by publishing two sets of resource records in DNS, as follows:

- Publish the real domain name in DNS. Each domain normally has a single DNS name. The domain-specific DNS records published by NetLogon are derived from the *dnsRoot* attribute on the cross-reference object for the domain, which holds the real DNS name of the domain (as opposed to an alias). The forest-specific DNS records are derived from the cross-reference object for the forest root domain, which holds the real name of the forest root. The NetLogon service running on a DC also responds to DC Locator pings for the domain name, and performs secure channel setup between a machine that is joined to the domain and the DC on which NetLogon runs.
- Publish the alias domain name in DNS. Soon after the *msDS-DnsRootAlias* attribute value on a cross-reference object is set on a DC, either by an originating write or by a replicated write, the NetLogon service on the DC additionally publishes a parallel set of DNS DC Locator records for the domain *alias* name that is held in the *msDS-DnsRootAlias* attribute. The domain-specific DNS records are derived from the *msDS-DnsRootAlias* attribute on the cross-reference object for the domain, and the forest-specific DNS records are derived from the *msDS-DnsRootAlias* attribute on the cross-reference object for the forest root domain. Further, on DCs for a domain whose *msDS-DnsRootAlias* is set, NetLogon responds to DC Locator pings for the DNS alias as if *msDS-DnsRootAlias* were the actual domain name. Also, secure channel setup from a domain member that believes it is connecting to a DC for a domain named in the *msDS-DnsRootAlias* attribute succeeds as if it were the real domain name.

Note that in the published SRV resource record corresponding to the alias domain name the owner field reflects the new domain name whereas the hostname field reflects the actual DNS name of the DC. For example, if the domain `cohovineyard.com` is being renamed to `cohowinery.com` then the following two SRV resource records for the LDAP service would be published by the NetLogon service on a DC named “dc01” in that domain:

```
_ldap._tcp.cohovineyard.com SRV 0 0 389 dc01.cohovineyard.com
_ldap._tcp.cohowinery.com SRV 0 0 389 dc01.cohovineyard.com
```

Observe that while the owner field in the second SRV record above corresponds to the new name to which the domain will be renamed, the hostname field reflects the true DNS name of the DC.

Pre-Publishing CNAME Records for Replication

The set of domain-specific resource records published in DNS by the NetLogon service running on a DC includes a DNS CNAME (alias) record for use by Active Directory replication. The owner name of the CNAME record is provided by:

DsaGuid._msdcs.DnsForestName.

which allows one DC to locate a replication partner DC in the forest. To find its partner, the DC requires knowledge of only the GUID of the directory system agent (DSA) object for that DC (as represented by *DsaGuid* in the CNAME record). The DSA GUID is the GUID of the NTDS Settings object (class *nTDSDSA*). Its value is stored in the *objectGUID* attribute of the NTDS Settings object, which is a child of the domain controller server object. These objects reside in the Sites container in the configuration directory partition. If the forest root domain is being renamed, it is essential that this CNAME record is *pre-published* in DNS so that replication continues to work after a DC is updated to change the forest root domain name. If the DNS CNAME records for the new forest root domain name are not pre-published for each DC, and records for the old forest root name are being replicated among DNS servers that use Active Directory to store DNS zones, then replication is interrupted due to a cyclic condition. The error generated by this condition is "cannot replicate because the CNAME record cannot be read from the local DNS replica; the CNAME record is not present in the local DNS replica because it is not replicating." Pre-publication of the DNS records serves to shorten the period during which the directory service is unavailable during the forest restructuring.

Note that in the pre-published CNAME resource record corresponding to the new forest name the owner field reflects the new forest name whereas the alias field reflects the actual DNS name of the DC. For example, if the forest `cohovineyard.com` is being renamed to `cohowinery.com` then the following two CNAME resource records would be published by the NetLogon service on a DC named “dc01” in that domain:

```
<DSA_guid>.cohovineyard.com IN CNAME dc01.cohovineyard.com
<DSA_guid>.cohowinery.com IN CNAME dc01.cohovineyard.com
```

Observe that while the owner field in the second CNAME record above corresponds to the new name to which the forest will be renamed, the alias field reflects the true DNS name of the DC.

Before the domain rename process can continue to the next step, the pre-published CNAME records corresponding to the value in *msDS-DnsRootAlias* must replicate to all DNS servers that are authoritative for those records.

Establishing Service Principal Names for a New Domain Name

The DSA on every DC writes a set of Service Principal Names (SPNs) to the *servicePrincipalName* attribute on the DC computer object in the domain directory partition. Among other things, SPNs are used for mutual authentication between DCs during Active Directory replication. The specific SPN used for mutual authentication between replication partners has the following three-part format:

E3514235-4B06-11D1-AB04-00C04FC2DCD2/DsaGuid/DnsForestName

where the first part is the Active Directory replication Remote Procedure Call (RPC) interface GUID, the second part is the GUID of the DSA object for the DC, and the third part is the DNS name of the forest root domain.

Pre-Publishing Two Sets of Service Principal Names

Soon after the value for the *msDS-DnsRootAlias* attribute on a cross-reference object is set on a DC, either by an originating write or by a replicated write, the DSA on the DC rewrites the SPNs on the DC computer object so that each SPN that includes the domain name (or the forest root name) is present in two versions — one for the actual domain (or forest root) name and one for the alias held in the *msDS-DnsRootAlias* attribute of the domain (or forest root) cross-reference object. The SPN values corresponding to the domain name alias require pre-publication for the same reason the DNS CNAME records do; that is, a cyclic condition that produces the error "cannot replicate because the SPN needed for mutual authentication cannot be read from the directory replica; the required SPN value is not present in the directory replica because it is not replicating." Pre-publication of the SPNs serves to shorten the period during which the directory service is unavailable during the forest restructuring.

Before the domain rename process can continue to the next step, the pre-published SPNs that correspond to the value in *msDS-DnsRootAlias* must replicate to all DCs in a domain as well as to all global catalog servers in the forest.

Actions Performed by Rendom in Response to the /upload Command

Rendom performs the following sequence of actions in response to the **rendom /upload** command:

- Connects to the DC that holds the domain naming master role for the forest and validates the forest description against the current state of the forest. If any of these validity checks fails, the command fails now. The following requirements are verified:
 - Each existing domain is part of the new forest.

- o The new forest is well formed.
- o The new forest does not re-assign domain names that are being relinquished as part of the current domain rename operation.
- Still connected to the domain naming master, retrieves all information needed to compute the list of rename instructions for updating the configuration and schema directory partitions.
- Connects to a randomly selected DC (or the one specified in the `<DcName></DcName>` field of the domain entry in the forest description file) for each domain, one by one. Retrieves all information needed to compute the list of rename instructions for updating each domain.
- Computes the full list of domain rename instructions for updating the entire forest (creates the script) and computes a signature to include in the script such that the authenticity of the script can be proven to a DC.
- Still connected to the domain naming master, writes the resulting script to the `msDS-UpdateScript` attribute of the Partitions container.
- Still connected to the domain naming master, writes the `msDS-DnsRootAlias` attribute of all cross-reference objects for domains that are being renamed.
- On the control station (the computer from which the `rendom` commands are issued), writes a new state file to track the progress of every DC in the forest. The state file contains an entry for every DC in the forest with each DC entry marked to be in the “Initial” state.

**Note**

Because the `msDS-UpdateScript` and `msDS-DnsRootAlias` attributes are first written to the directory on the DC holding the domain naming master role and then replicated to the remaining DCs in the forest, if the domain naming master is unavailable during the **rendom /upload** operation, the process cannot continue.

Side Effect of /upload Command on Forest Configuration

Certain types of changes in the forest made after the domain rename operation has begun can affect the outcome of the rename operation and cause it to fail in a way such that some of the DCs can never complete the rename process. In order to prevent this situation from arising, the forest configuration is frozen with respect to these types of changes once the rename operation has begun. Successful execution of the **rendom /upload** command results in the forest configuration being frozen with respect to these types of changes. The following operations cannot be performed in a forest after the `/upload` command has successfully completed:

- The addition or removal of a domain or application partition,
- The addition/removal of a DC into/from an existing domain,
- The addition or removal of trusts including cross-forest trusts. (Note that attributes of an existing trust can be changed during this frozen configuration. For example, a unidirectional trust can be converted to a bidirectional trust.)

The forest configuration remains frozen as long as the *msDS-UpdateScript* attribute of the Partitions container has a value that indicates a domain rename operation in progress. The forest configuration becomes unfrozen at the conclusion of the domain rename operation by executing the **rendom /end** command.

How DC Readiness is Verified

This step of the domain rename process verifies that the directory database at each DC in the forest is adequately prepared to perform the directory modifications dictated by the script in the *msDS-UpdateScript* attribute. In response to the **rendom /prepare** command, the necessary verification is performed on each DC by executing the *test* component of the script in *msDS-UpdateScript* as a read-only transaction on the local directory database. The test checks for the following conditions:

- The correct script in *msDS-UpdateScript*, having replicated to this DC.
- Any trust relationships that are required by the new forest structure.
- Pre-published SPNs for all DCs.
- Name conflicts due to administrative errors since the time the script in *msDS-UpdateScript* was created. For example, an administrator might have mistakenly removed a trust required in the new forest, or created a computer account whose SamAccountName equals the new name of some trusted domain, creating a name conflict with an interdomain trust account.

Additionally, the test includes an authorization check on each DC to determine whether or not the user running the **rendom /prepare** command is authorized to execute the domain rename instructions contained in *msDS-UpdateScript*. The authorization check consists of verifying that the user has *write* permission on the *msDS-UpdateScript* attribute on the Partitions container. If the user is not authorized, the command fails with an error.

Actions Performed in Response to the /prepare Command

The following sequence of actions occurs in response to the **rendom /prepare** command:

- Rendom issues a special RPC (for internal use only) to every DC in the forest in turn, requesting authorization and verification of readiness as encoded in the test component of the script in *msDS-UpdateScript*. The RPC is issued to multiple DCs at a time with a high degree of concurrency, while ensuring that resource limits on the machine executing the command are not exceeded. The RPC request and response are signed and sealed for integrity and privacy.
- In response to the RPC, each DC performs the authorization check, ensures the authenticity of the script in *msDS-UpdateScript* by validating the signature, and performs verification of the test component of the domain rename script before responding. If any of these checks fails on a DC, the RPC returns an error for that DC.

- Rendom updates the state file with the state of each DC that was successfully contacted and passed verification. The state of each successfully verified DC is updated from the Initial state to the “Prepared” state. The Prepared state indicates that the DC has authorized the execution of the restructure and that the contents of its directory database are consistent with the rename instructions contained in *msDS-UpdateScript*. If a DC cannot be contacted or if it fails any of the checks, its corresponding state in the state file remains as Initial with an appropriate error code and message to indicate the cause of failure.

How Domain Rename Instructions are Executed

In the final step of the domain rename process, the directory database at each DC in the forest is updated individually to make the new forest structure effective. This process does not rely on Active Directory replication. Rather, the required modifications to the directory database on each DC is performed locally by executing the *action* component of the script in *msDS-UpdateScript* as a single update transaction. The action component of the script is the actual update of the domain name. The **rendom /execute** command is used to perform this final update step.



Note

The actions performed at this step make the actual domain name changes effective at each DC. This step causes a brief interruption in service. Prior to this point in the process, forest service has not been disrupted.

Single-User Mode

To perform the action component of the script in *msDS-UpdateScript*, the directory service on each DC enters a special mode called *single-user mode*. In single-user mode, Active Directory refuses service to all normal clients of the directory service, including LDAP, MAPI, replication, SAM, Kerberos, and other directory service RPCs. In this mode, only the directory service itself can read and write the local directory database, using a single thread. The single-user mode is needed because the domain rename updates that the directory service performs invalidate its internal data structures. After the directory service performs these updates in single-user mode, the DC reboots. Following the reboot, the internal data structures are rebuilt to a consistent state.

Update Transaction

All the directory updates specified by the action component of the script in *msDS-UpdateScript* are performed within an Extensible Storage Engine (ESE) database transaction. If no errors occur, the transaction is committed; otherwise the entire transaction is rolled back. Domain controllers that have committed the update transaction in single-user mode and then rebooted can be considered as having completed the domain rename.

Enter Single-User Mode

As part of a successful update transaction, the non-replicated, single-valued integer attribute *msDS-ReplicationEpoch* on the NTDS Settings object for the DC is updated to a new value by incrementing its current value. If two DCs have different *msDS-ReplicationEpoch* values, no directory replication RPC interaction is allowed between them. In addition to replication, nested group membership evaluation and global catalog lookups are also discontinued. Because the domain rename procedure involves making these updates independently at all DCs in the forest, it is impossible to modify these DCs simultaneously. The goal of the *msDS-ReplicationEpoch* attribute is to minimize potentially complex interactions, including replication, between DCs that have completed the domain rename and those DCs that have not yet completed the domain rename.

Switch Real and Alias DNS Names

Further, as part of a successful update transaction, the values of the *dnsRoot* and *msDS-DnsRootAlias* attributes on the cross-reference objects of renamed domains are interchanged such that the new domain name, formerly stored in *msDS-DnsRootAlias*, becomes the effective domain name stored in *dnsRoot*. The old domain name is saved as a DNS alias until being removed by a subsequent step.

Actions Performed in Response to the /execute Command

The following sequence of actions occurs in response to the **rendom /execute** command:

- Rendom checks to ensure that *all* DCs in the state file are marked with a current state of Prepared. If the state of any DC is not Prepared, Rendom reports an error and the process cannot continue.
- When all DCs in the state file are in the Prepared state, Rendom issues a special RPC (for internal use only) to every DC in the forest requesting execution of the directory updates encoded in the action component of the script in *msDS-UpdateScript*. The RPC is issued to multiple DCs at a time with a high degree of concurrency, while ensuring that resource limits on the machine executing the command are not exceeded. The RPC request and response are signed and sealed for integrity and privacy.
- In response to the RPC, each DC first performs the verification of the test component of the domain rename script in *msDS-UpdateScript* (just as in the previous step to check for DC readiness). The DC then performs the action component of the script in an update transaction, as described in "Update Transaction" earlier in this document. If any of the checks fails on a DC or if the update transaction cannot be successfully committed, the RPC returns an error for that specific DC.

- Rendom updates the state file with the state of each DC that was successfully contacted. For each DC that successfully completed the update, Rendom changes the state from “Prepared” to the final state “Done.” If a DC cannot be contacted or if it fails any of the checks, its corresponding state in the state file remains Prepared with an appropriate error code and message to indicate the cause of failure. If the RPC returns with an error that is deemed irrecoverable, then the corresponding state for the DC is updated to the final state of Error with an appropriate error code and message to indicate the cause of failure.

Determining Domain Rename Completion

In a large forest, some number of DCs might prove impossible to contact during the domain rename process. These DCs never reach the final state of Done. You must decide how long to continue to try to reach DCs that have been unreachable and to retry failed update attempts on DCs that have reached the Error state. When further progress in the forest is not deemed possible, then declare the domain rename process to be complete. All DCs that did not reach the final Done state because they were either unreachable or finished in the Error state must be demoted (run Configure Your Server to remove Active Directory) or removed from service if demotion is not possible. For the forest to function without problems following the domain rename, only DCs that have reached the Done state can exist in the forest.

How Group Policy is Reconciled Following Domain Rename

When the DNS name of a domain changes, any references to Group Policy Objects (GPOs) within the renamed domain through Group Policy links (the *gpLink* attribute) on sites, domains, and organizational units is rendered invalid because they are based on the old domain name. Furthermore, the optional attribute *gpcFileSysPath* on a GPO that holds a uniform naming convention (UNC) path to a Group Policy templates folder located in the sysvol volume of the renamed domain will also be rendered invalid because the path uses the old domain DNS name. To correct the severed Group Policy links and the invalid UNC paths in GPOs within the renamed domain, you can use the Group Policy fix-up tool *gpfixup.exe* to refresh the Group Policy links and the UNC paths in GPOs based on the new domain name.

The Group Policy fix-up tool should be run once for every renamed domain soon after the actual domain rename operation has been completed and before another domain rename operation is performed.

**Note**

The fix-up tool *gpfixup* refreshes all intradomain GPO references/links (that is, where the link and the target GPO are within the same domain) in the renamed domain. However, cross-domain references to GPOs in the renamed domain, where the link is in a different domain from the domain containing the GPO, will not be automatically rebuilt by this tool. For them to work, these cross-domain links will need to be repaired manually by deleting the old Group Policy links and re-establishing new links.

How Old Domain Names are Removed Following Domain Rename

Following completion of the domain rename process for a forest, you use the **rendom /clean** command to remove the old domain names from Active Directory. This cleanup step removes all values of *msDS-DnsRootAlias* from the domain naming master, and removal of this value is replicated to all domain controllers in the forest. Because this attribute holds the old domain name after the domain rename is completed, replication of the removal of the *msDS-DnsRootAlias* attribute value to all DCs in the forest prompts NetLogon on the DC to deregister the DNS locator resource records for the old domain name. In the course of this deregistration, the DNS CNAME resource records, which are used by Active Directory replication, are also removed for the old domain name.

The cleanup procedure also removes the *msDS-UpdateScript* attribute value from the Partitions container on the domain naming master.

After you have run the **rendom /clean** command successfully, the new forest is ready for another forest restructuring operation.