realtimepublishers.com™

Tips and Tricks Guide™ To

Active Directory Troubleshooting

NETPRO
The Directory Experts

Don Jones

Note to Reader: This book presents tips and tricks for Active Directory troubleshooting topics. For ease of use and for cross referencing, the questions are numbered.

Copyright Statement

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com

Q.16: How does the Knowledge Consistency Checker work?

A: Active Directory (AD) supports two distinct types of replication: *intrasite,* which covers all domain controllers within a site, and *intersite,* which covers replication between different sites. Intrasite replication is managed pretty much automatically by the AD Knowledge Consistency Checker (KCC). The KCC is responsible for ensuring that each domain controller within the site participates in replication in a timely fashion. The KCC also plays a role in generating the replication topology between sites. Although the KCC generally works flawlessly, it's a good idea to understand exactly how it works so that you can troubleshoot it when problems arise.

> ✎ The KCC is physically implemented as a component of a service that runs on all domain controllers.

Automatic Intrasite Replication Topology

One of the KCC's most important tasks is to generate the intrasite *replication topology,* a sort of logical map that decides which domain controllers will replicate with each other. AD does not use a *fully enmeshed* replication topology in which each domain controller in a site replicates with every other domain controller in the site; such a topology would generate an unacceptably high level of replication traffic, especially in sites with a large number of domain controllers. Instead, the KCC tries to generate a topology that minimizes latency and network utilization while providing fault tolerance.

> ✎ Latency refers to the amount of time it takes a change to replicate from the domain controller on which the change was made to all other domain controllers. A high degree of latency can create inconsistent results for users and can even be the source of potential security problems, so reducing latency is a big priority for the KCC.

By default, the KCC tries to ensure that each domain controller in the site has at least two replication partners. That way, if one becomes unavailable, replication can still continue. Additionally, the KCC tries to ensure that no single domain controller is more than three partners away from any other domain controller. This configuration reduces latency because any change to AD would require no more than three replication cycles. Generally, the KCC creates a *bidirectional ring* topology, as Figure 16.1 shows. In this pattern, each domain controller replicates with the domain controllers to its left and right (two partners apiece).

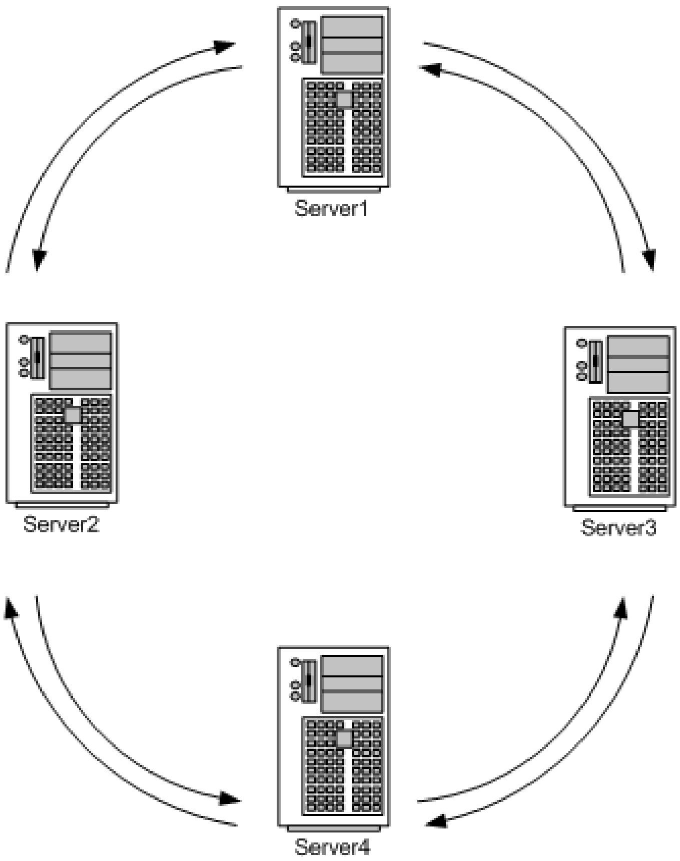


Figure 16.1: Bidirectional ring topology.

In a site with many domain controllers, a ring topology can quickly violate the no-more-than-three-hops rule, so the KCC will generate shortcuts across the ring to reduce the number of hops between domain controllers. Figure 16.2 shows a ring topology with a larger number of domain controllers in which shortcuts are used to reduce hops. Note that some domain controllers are therefore required to carry more than the minimum two connections. However, to reduce domain controller overhead, no domain controller is required to have more than three replication partners.

Server5, which would have been four hops away from Server1 in a bidirectional ring without shortcuts, is reduced to a single hop by the shortcut. Note that the two-way arrows indicate two individual connection objects apiece; I've simplified them into a single arrow for this drawing.

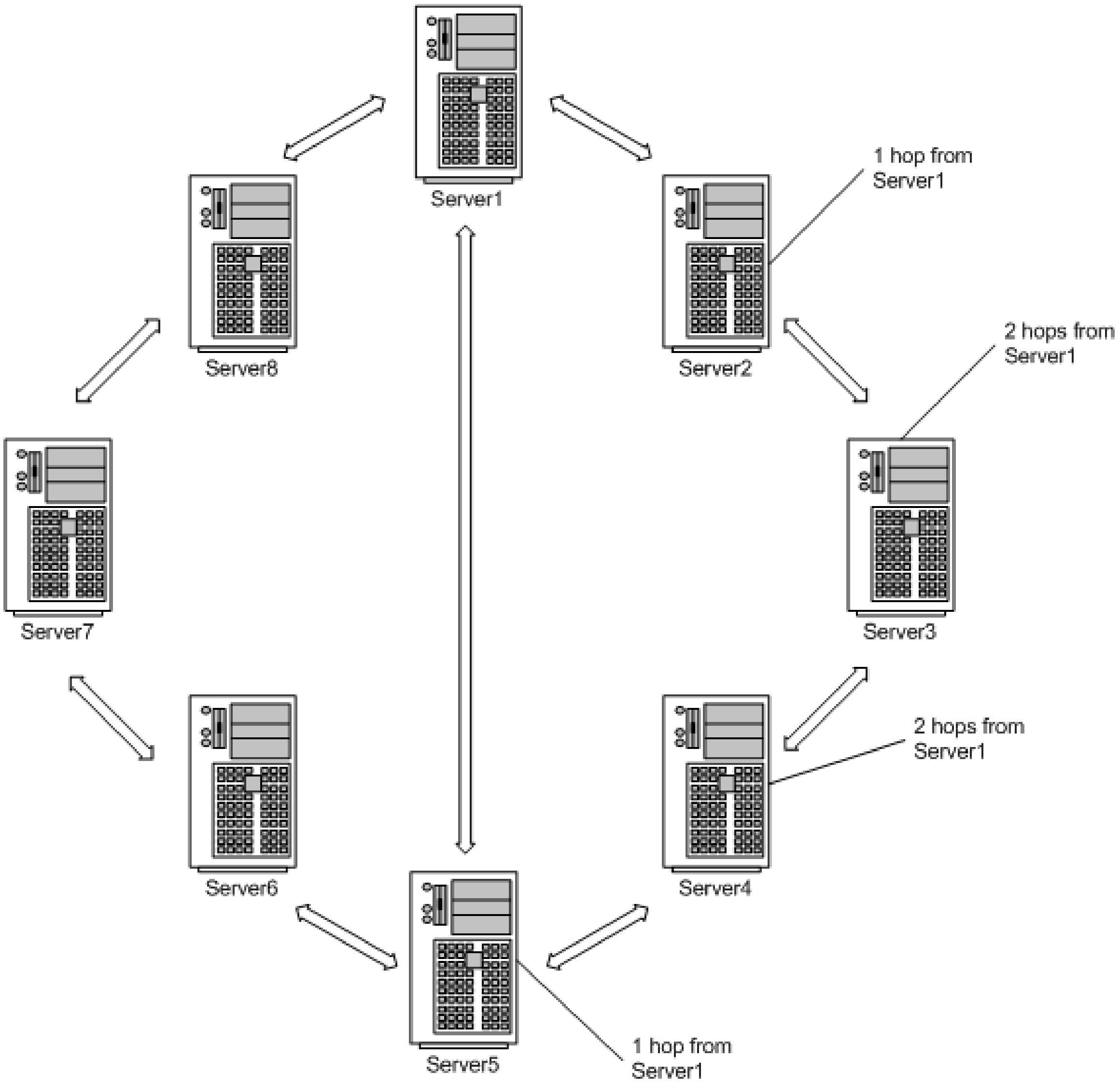


Figure 16.2: Bidirectional ring topology with shortcuts.

As I've mentioned, this topology is automatically generated by the KCC. Actually, the KCC builds several different, independent replication topologies: One for each AD partition. This setup results in a schema replication topology, a configuration topology, a domain topology, and an application topology. Although the various topologies will often be similar, they might not be identical. Note, however, that the KCC never creates duplicate connection objects.

For example, suppose the KCC generates an identical topology for both the schema and configuration partitions. Only one entire site of connection objects will be created for both topologies. If the KCC generates a different topology for the domain partition, then it might create different connection objects. The KCC will never, for example, create two objects from Server1 to Server2; if different topologies require such a connection, they will all share a single connection.

📖 For more information about how replication works, see Question 19.

Automatic Connection Objects

You can't directly affect the KCC's operation. When it creates its replication topology, the result is a set of replication objects. The security on these objects sets the KCC itself as the owner, although members of the Domain Administrators group have permission to modify those objects. As an administrator, you *can* create your own intrasite replication objects. The KCC won't have the ability to modify any objects you create. (For information about AD connection objects, see the sidebar "AD Connection Objects.")

AD Connection Objects

Keep in mind that each connection object represents a one-way, inbound replication path from the domain controller on which the change occurred to the local domain controller. AD replication is *pull-based,* meaning domain controllers request changes from other domain controllers. This concept is important for security: domain controllers do *not* accept *pushed* changes, meaning there's no way for an intruder to send fake replication data around your network and mess up your domain.

When the KCC wants two domain controllers to replicate with each other, it must create two connection objects with each object representing one direction of replication traffic.

You can use the Active Directory Sites and Services console to see the connection objects that exist for each domain controller. Open the console, and expand Sites. Then select the appropriate site (such as Default-First-Site), and expand Servers. Locate the server you're interested in, expand it, and select NTDS Settings. You'll see the connection objects that control replication to other servers within the site. You'll also see the intrasite replication schedule, which is about every 5 minutes by default. Connection objects created by the KCC will have a name of <automatically generated>; connection objects you create will have a descriptive name that you assign.

Replication Security

How does the KCC have permission to perform its job? The AD partitions each grant special permissions to both the Enterprise Domain Controllers group and the built-in Administrators group:

- Replicating Directory Changes—This permission lets replication actually occur.
- Replication Synchronize—This permission lets a synchronization request be issued.
- Manage Replication Topology—This permission allows for the creation and modification of connection objects, which describe the replication topology.

Manual Connection Objects

When the KCC is automatically generating connections, why should you bother? There are a number of reasons. Perhaps the most common reason is to reduce latency. You might, for example, want to have no more than two hops between any two domain controllers to decrease replication latency. You can accomplish this setup by manually creating connection objects. To do so

1. Navigate to the appropriate server in Active Directory Sites and Services, and select the NTDS Settings item. Remember that you're creating an object that represents an inbound connection, so select the server that will be the incoming replication partner.

2. Right-click NTDS Settings, and select New Active Directory Connection from the context menu.
3. Select a domain controller from the list. It should be the domain controller that will be the replication partner. Note that you cannot create a connection from one server to itself.
4. Provide a name for the connection object, and you're done!

Keep in mind a couple of things regarding manually created connections:

- The KCC will never delete your connections. If a domain controller becomes unavailable, breaking a manual connection, then the KCC might automatically generate new automatic connections to compensate.
- If a KCC-generated topology requires a connection between, say, Server1 and Server2, and a manually created connection already exists, the KCC will not generate a new connection. The topology will instead use the manually created connection.

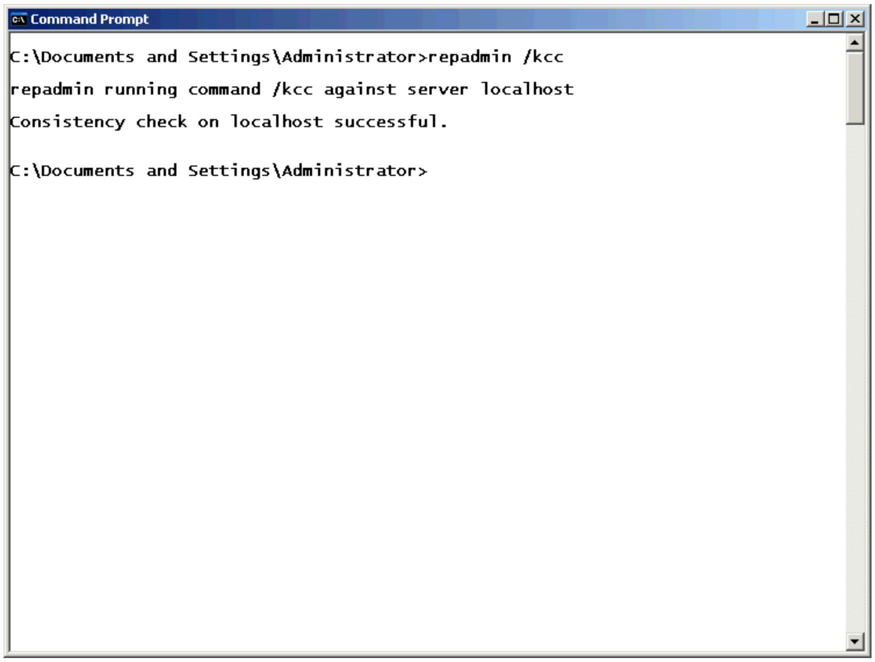
Be very careful when creating manual connections. In fact, unless you plan to regularly monitor your connections, don't create any at all. Manual connection objects can easily create additional domain controller overhead by requiring domain controllers to maintain connections that aren't as efficient. Also, the more connections you create, the fewer the KCC will create (remember that it tries to minimize the number of connections maintained by each domain controller), which can result in an inefficient replication topology.

Controlling the KCC

You can use the Repadmin.exe command-line tool to manipulate some of the KCC's functionality and to check its performance. For example, executing

```
repadmin /kcc servername
```

will force the KCC on the designated server to run and immediately begin recalculating the replication topology. As Figure 16.3 shows, you can omit the server name to run the KCC on the local domain controller.



```
C:\Documents and Settings\Administrator>repadmin /kcc
repadmin running command /kcc against server localhost
Consistency check on localhost successful.

C:\Documents and Settings\Administrator>
```

Figure 16.3: Forcing the KCC to run.

Or you can run

```
      repadmin /showreps servername
```

to show the replication topology as viewed from the designated server. The output of this command displays both inbound and outbound *neighbors,* or replication partners.

> ✐ Inbound neighbors are ones from which the reporting domain controller will request and pull changes. Outbound neighbors are those that will receive change notifications from the reporting domain controller.

Manually drawing the replication topology based on this information can be complex. The Windows Support Tools includes Replmon, a graphical tool that can draw a graphical picture of the replication topology to help you more easily spot topology issues.

The KCC and Intersite Replication

The KCC also generates the topology for intersite replication. However, it cannot do so completely on its own. Instead, it must rely on information that you supply about the intersite networking infrastructure (generally WAN links). You supply this information by accurately configuring AD sites to represent your LANs, and by creating site links that represent the WAN links between LANs.

The KCC on a single domain in each site is designated as the *intersite topology generator* and is responsible for managing that site's connections to other sites. The generator uses a fancy algorithm—called a *least-cost spanning tree* algorithm—to calculate the most efficient way to replicate information across sites.

By default, the intersite topology generator is the first domain controller in the site. However, this can change if that domain controller becomes unavailable for more than an hour; you can check Active Directory Sites and Services to see which server holds the role. You cannot, however, designate a server to hold this role. If the existing generator becomes unavailable, the domain controller with the next-highest AD GUID will become the new generator.

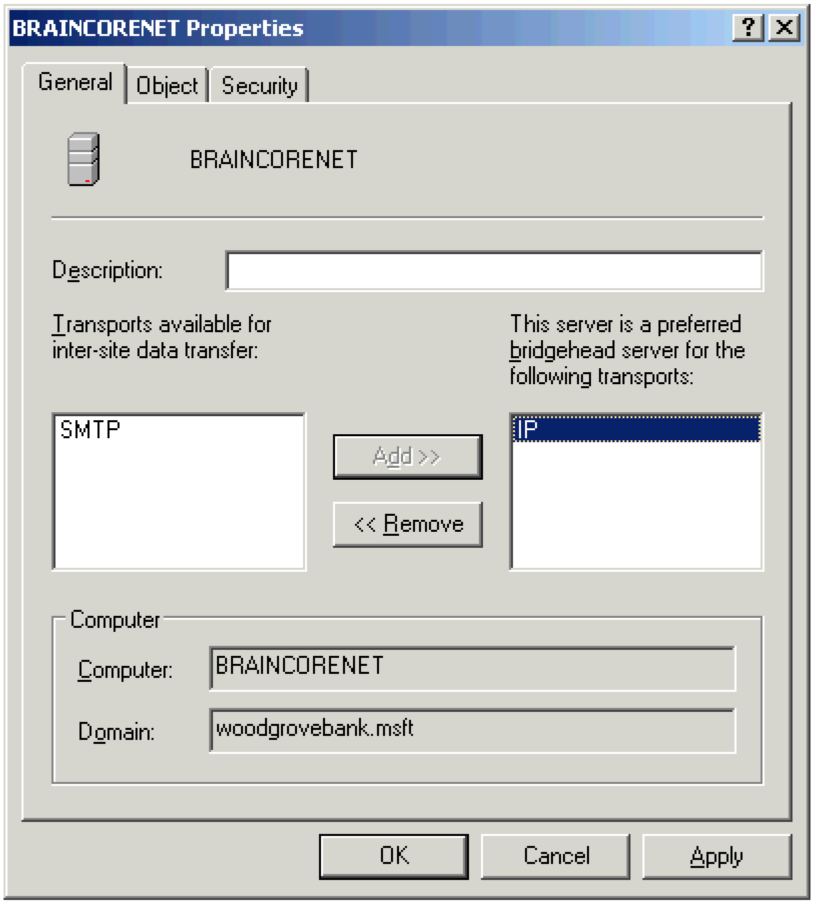
The KCC has to consider several factors when generating the intersite topology. These factors include the availability of site links, which tells the KCC which sites it can reach and which sites can be used to reach other sites. It must also consider the network protocols available on site links, which will usually include IP and/or SMTP. The KCC must also consider the cost of site links. In the event that multiple possible paths are available to a site, the KCC will choose the least expensive option. You configure the cost of site links when you create them.

For intersite replication, the KCC chooses a single domain controller as the *bridgehead* for each domain in the site. The bridgehead is responsible for all replication traffic to a particular site. If the bridgehead fails, the KCC will choose another from the same domain.

Remember that only the KCC on the intersite topology generator deals with intersite replication topologies.

You can also manually designate bridgehead servers by manually creating connection objects between two domain controllers in the same domain but in different sites. The KCC will detect the existing connection object when it generates the intersite topology and use the manually created connection rather than creating a new connection. Manually designating bridgeheads ensures that you can select a server that can support the additional traffic that intersite replication entails. You can also ensure that, if desired, a single domain controller acts as the bridgehead for its domain to all other sites, providing a consolidated bridgehead (and putting all of your replication eggs in one basket, so to speak). If your designated bridgehead fails, the KCC will automatically select a new bridgehead for each site connection.

Finally, as Figure 16.4 shows, you can designate particular domain controllers as *preferred* bridgeheads. Simply right-click the server in Active Directory Sites and Services, and indicate which transport protocols the domain controller is preferred for. The KCC will always choose a preferred server as the bridgehead if one is available.



Description:

Transports available for inter-site data transfer: SMTP

This server is a preferred bridgehead server for the following transports: IP

Computer

Computer: BRAINCORENET

Domain: woodgrovebank.msft

Figure 16.4: Making a domain controller the preferred bridgehead for IP connections.

Automatic Updates

The KCC is configured to automatically recheck the network topology every so often. Doing so allows the KCC to reconfigure the topology to respond to changing network conditions, such as new domain controllers, failed domain controllers, failed WAN links, new site link configurations, and so forth.

Q.17: How can I force Active Directory to replicate changes?

A: Generally, Active Directory (AD) replication works completely automatically and, due to the way its replication topology works, provides replication with very low latency. However, there might be times when—either as a troubleshooting step or as a workaround to a problem—you need to force AD perform replication.

> ✎ Having to force replication is a sign of a problem in most instances. You should attempt to fix the problem so that manual replication isn't necessary.

> 📖 For more information about how AD replication works, see Question 19.

Check the Topology

Before forcing replication, try to perform a quick fix by checking the replication topology. Windows' Support Tools includes Repadmin.exe on domain controllers to help with this. Simply execute

```
repadmin /showreps servername
```

to show the replication partners for a designated server (you can omit *servername* to run the tool against the local computer, if it's a domain controller). If you suspect that one server isn't replicating properly, check its partners. Then verify that each of those partners is functioning and considers the suspect domain controller to be a partner as well.

If the topology seems to be the problem, a quick fix might be to force the Knowledge Consistency Checker (KCC) to regenerate the topology. It could be that you've caught a recent topology problem and that the KCC simply hasn't run yet. Run

```
repadmin /kcc servername
```

to force the KCC on the designated domain controller to regenerate its topology. Follow up with

```
repadmin /showreps servername
```

to see the newly selected replication partners.

For more information about how the KCC generates the replication topology, see Question 16.

For intersite replication issues, determine which domain controller in each affected site (and domain) is acting as the bridgehead server. These will be the only domain controllers in the site with a connection object to a domain controller in another site. If a designated bridgehead domain controller is unavailable or disconnected, the intersite replication will fail. You can check these connections using either repadmin or the Active Directory Sites and Services console.

One potential cause of intersite replication issues is that the intersite topology generator has failed within the last hour, and problems have occurred with the replication topology (such as a designated bridgehead domain controller also failing). AD will correct this problem automatically within about an hour, because it will choose a new topology generator and recalculate the intersite topology.

Why spend all this time worrying about the replication topology? Simple—forcing replication doesn't recalculate the topology. It simply forces AD to replicate using the existing topology; if that topology is flawed, then forcing replication won't solve any problems.

For more information about troubleshooting topology issues, see Question 20.

Forcing Replication

The easiest way to force replication is through the Active Directory Sites and Services console. To do so, open the console, and locate the domain controller that you want to replicate. This domain controller will request changes from its replication partners. Locate the connection over which you want to force replication, right-click the connection, and select Replicate Now.

If the domain controller that you want to replicate doesn't have any valid connection objects, you have a replication topology problem. You can provide a quick fix by manually creating a connection object to a known-working domain controller in the same site (if possible) and domain, and forcing replication over that connection.

📖 For additional methods of forcing replication, refer to the Microsoft article "Initiating Replication Between Active Directory Direct Replication Partners."

Q.18: One of my Windows NT Backup Domain Controllers stopped replicating changes from my Active Directory domain controllers. What do I do?

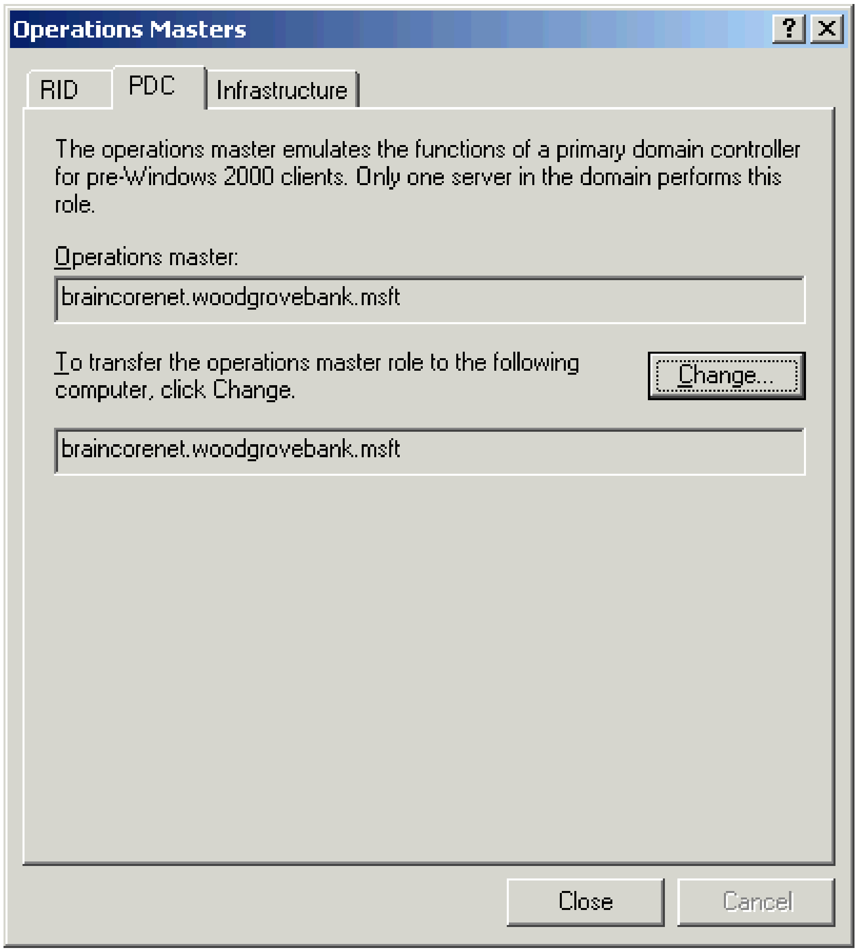
A: For better or for worse, many Active Directory (AD) domains still contain down-level Windows NT Backup Domain Controllers (BDCs). Perhaps your BDCs are needed to support an older application or you've got a mission-critical application running on one. Whatever the case, it's usually important that the BDC continue to replicate changes from your AD domain, just as it used to do from your NT Primary Domain Controller (PDC). Sometimes, however, replication stops working. Fortunately, there are a fairly limited number of causes.

PDC Emulator Failure

Easily the most common cause for BDC replication issues is a failure of some kind in the domain's PDC emulator. The PDC emulator is a Flexible Single Master Operation (FSMO) role that is assigned to one domain controller in the domain.

📖 For more information about how the PDC emulator works, see Question 1.

To troubleshoot this problem, first determine which domain controller holds the PDC emulator role. To do so, open Active Directory Users and Computers, right-click the domain, and select Operations Masters from the context menu. Select the PDC tab, which Figure 18.1 shows, and note the name of the server listed.



The operations master emulates the functions of a primary domain controller for pre-Windows 2000 clients. Only one server in the domain performs this role.

Operations master:

braincorenet.woodgrovebank.msft

To transfer the operations master role to the following computer, click Change.

braincorenet.woodgrovebank.msft

Figure 18.1: Identifying the domain's PDC emulator.

If the designated server is not available on the network or is not responding to ping commands from your NT BDC, you need to resolve that problem. You can either try restarting the PDC emulator to fix it or you might need to seize the PDC emulator role at another domain controller.

For instructions about transferring and seizing FSMO roles, see Question 8.

Make sure that there are no IPSec policies in effect that would prevent an NT computer from communicating with the PDC emulator. For example, applying certain IPSec policies to a Windows 2000 (Win2K) or later computer will prevent communications with down-level clients, which includes NT BDCs.

Other issues can occur when Win2K or later domain controllers don't replicate properly with the PDC emulator; see Microsoft article "Event 1586 Message: The Checkpoint with the PDC Was Unsuccessful" for more information.

Domain Account Failure

A less common cause for BDC replication failure is when the BDC's domain computer account becomes locked out or out of synch. BDCs maintain a domain account in much the same way as users and other computers do. The BDC's accounts must be available and the BDC must know the account password in order for the BDC to participate in the domain.

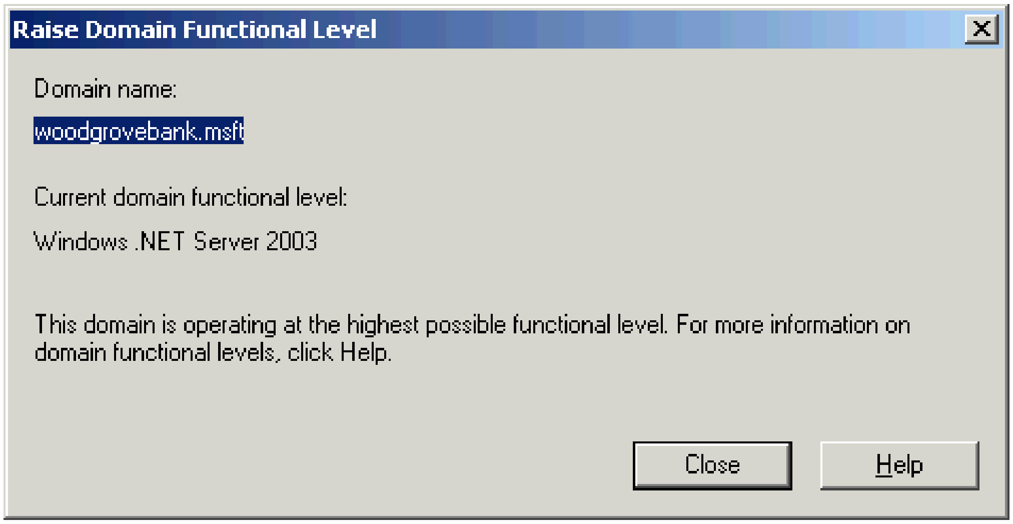
Use Active Directory Users and Computers to ensure that the BDC's account is available and not locked out. Try logging on to the domain from the BDC's console; the BDC must first log on to the domain in order to process any domain user logons. If you can successfully log on to the domain from the BDC console, the BDC's domain account is probably fine. If you cannot or if the BDC's event logs contain domain logon errors, you might need to reset the BDC's domain account, which is a particularly tricky task with a BDC.

> 📖 For more information about how to reset the BDC's domain account, see Question 2.

> 📖 BDC domain accounts might appear as users rather than computers; see the Microsoft article "HOW TO: Create a Computer Object in the Active Directory for a Windows NT 4.0 BDC" for more information.

Domain Mode or Functional Level

Hopefully, your Win2K domain is running in mixed mode or your Windows Server 2003 domain is running in its Win2K mixed mode functional level. In Windows Server 2003, check the functional level by opening Active Directory Users and Computers, right-clicking the domain, and selecting Raise Domain Functional Level from the context menu. The resulting window will show you the current level (see Figure 18.2). Any level other than Win2K mixed mode will result in NT BDCs being unable to replicate. There is no solution for this problem other than decommissioning your NT BDCs or restoring every Win2K (or Windows Server 2003) domain controller from a backup—effectively performing forest-wide disaster recovery.



Raise Domain Functional Level

Domain name:

woodgrovebank.msft

Current domain functional level:

Windows .NET Server 2003

This domain is operating at the highest possible functional level. For more information on domain functional levels, click Help.

Figure 18.2: The domain's functional level.

✎ Generally, Windows Server 2003 and Win2K will warn you if you attempt to change the domain mode or functional level and NT BDCs still exist in the domain. Windows Server 2003, in fact, will attempt to stop you outright. However, if you do change the mode or functional level, it's a one-way operation that can't be undone; any remaining NT BDCs will be useless.

BDC Logs Events 3210, 7023, or 8032

If you inadvertently configure a Win2K domain to restrict anonymous connections, NT BDCs might be unable to locate a domain controller, replicate the domain database, start the Net Logon service, and so forth. Although restricting anonymous access to the domain is a valuable security technique (and is included in Microsoft's Hisecdc.inf security template), it can cause problems for NT BDCs. You can correct the problem with a registry edit or decommission the NT BDC.

🕮 For more information about this problem and its solution, see the Microsoft article "The Net Logon Service of a Windows NT 4.0 BDC Does Not Function In a Windows 2000 Domain". The article "How to Use the RestrictAnonymous Registry Value in Windows 2000" describes more about the RestrictAnonymous registry setting, which is included in the Hisecdc.inf security template.

Q.19: How does Active Directory replication work?

A: Active Directory (AD) is a multi-master directory, meaning each directory services server—referred to as a *domain controller*—contains a fully readable and writable copy of the directory services database. Because all domain controllers can accept changes to the database, some method is needed to replicate those changes to other domain controllers, ensuring a consistent database across all domain controllers. This scheme is referred to as AD replication.

AD replication can be broken down into four basic operational components:

- *Who,* which is a list of servers that participate in replication and the servers with which they replicate. Referred to as a *replication topology,* this list is generated by a special AD component called the Knowledge Consistency Checker (KCC).

🕮 For information about how the KCC works, see Question 16.

- *What,* which is the information that is being replicated. AD uses attribute-based replication and versioning to determine which information has changed and requires replication.
- *When,* which is a schedule that determines when replication will occur. Separate schedules exist for replication within an AD site and for each link connecting different sites.
- *How,* which defines how the replicated data is physically transported across the network.

Deciding What to Replicate

AD maintains multiple attributes for each object. For example, a user object has attributes such as password, account lockout status, user name, and so forth. Each attribute is versioned independently, letting AD replicate only attribute changes. For example, when a user changes his or her password, only that particular attribute receives a new version number and is replicated to other domain controllers (rather than replicating the entire object).

> ✎ Windows Server 2003 uses *linked value replication* to improve replication efficiency. Security groups are an example of a multivalued attribute with linked values, in which the group's multiple attributes are references to the group's members. In Windows 2000 (Win2K), changes made to a member of a group require the entire group to be replicated. In Windows Server 2003, only the group member that has changed is replicated. This feature is only available when the forest functional level is at Windows Server 2003.

Version Control

Obviously, some form of version control is necessary to determine which domain controller has the most recent version of each attribute. AD uses *update sequence numbers* (USNs) as a form of version number. Each USN is a 64-bit number; whenever a domain controllers changes an attribute, it increments the domain controller's USN. Each domain controller maintains only a single USN.

Each domain controller also maintains a list of USNs that have been received from other domain controllers during replication. When a domain controller informs its replication partners that changes are available for replication, its partners respond with the USN previously received from that domain controller. The sending domain controller can then send only the attributes that have changed since that USN, ensuring that data already replicated isn't replicated again. This technique also allows for domain controllers to remain offline for a period of time; when they replicate the next time, they'll receive every change that's been made since the last time they replicated. Figure 19.1 illustrates this process.

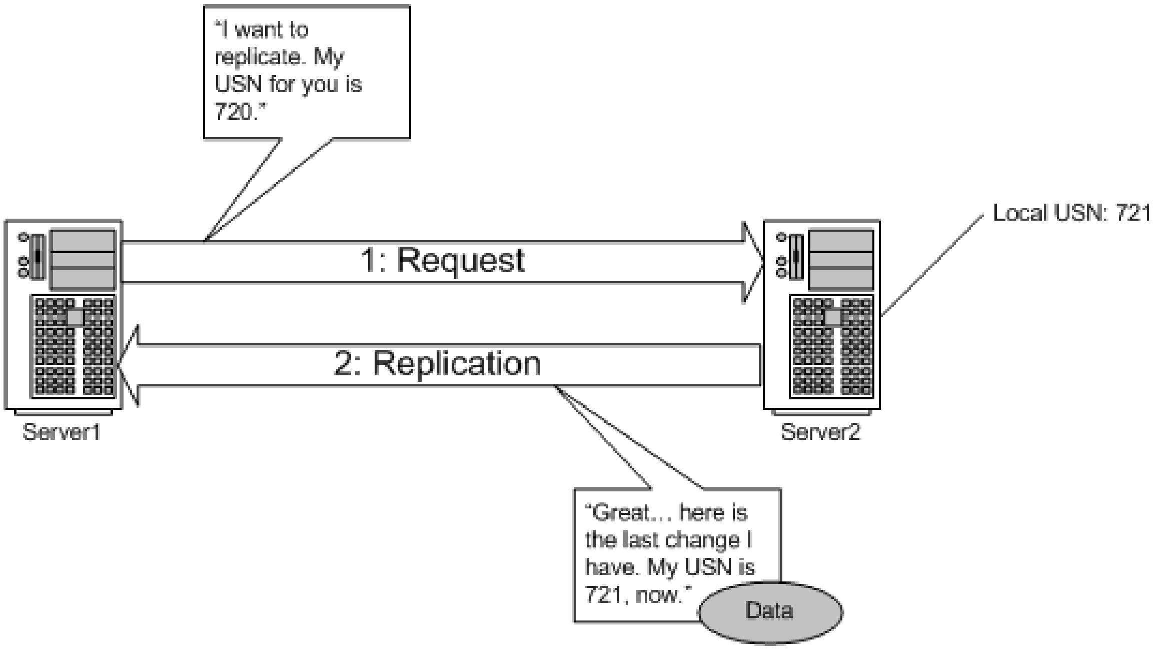

"I want to replicate. My USN for you is 720."
1: Request
2: Replication
Local USN: 721
Server1
Server2
"Great… here is the last change I have. My USN is 721, now."
Data

Figure 19.1: USNs and replication.

Dealing with conflicts is also important because it's possible for a single attribute to be changed on two domain controllers at once (for example, two administrators might attempt to change the same user's password at the same time). A replication *collision* occurs when a single attribute is changed on one domain controller, while a previous change to that attribute is still in the process of replicating. To help resolve collisions, AD maintains a *property version number* for each attribute in the directory. USNs are server-specific, whereas property version numbers are attached to each attribute in the directory and are replicated along with that attribute.

Changing an attribute increments its property version number. *Replicating* an attribute does *not* change its property version number; only "new" changes have that effect. Whenever a domain controller receives a replicated attribute with the *same version number as the local copy,* the domain controller knows a collision has occurred. In other words, another source has changed the attribute but that source hadn't first received a replicated change from a second source. The result is two changes running around the network with the same property version number. When this occurs, the domain controller receiving the change keeps the one with the latest timestamp. This situation is the only instance in which AD replication relies on timestamps and is the reason that Win2K and later includes a time synchronization service. Figure 19.2 shows how a replication collision is handled.

> ✎ It's possible for an attribute to be replicated with a *lower* property version number. This situation can happen when the domain controller making the attribute change has missed two or more replications of that attribute from other sources. AD discards any replicated data with a lower property version number than the local copy.

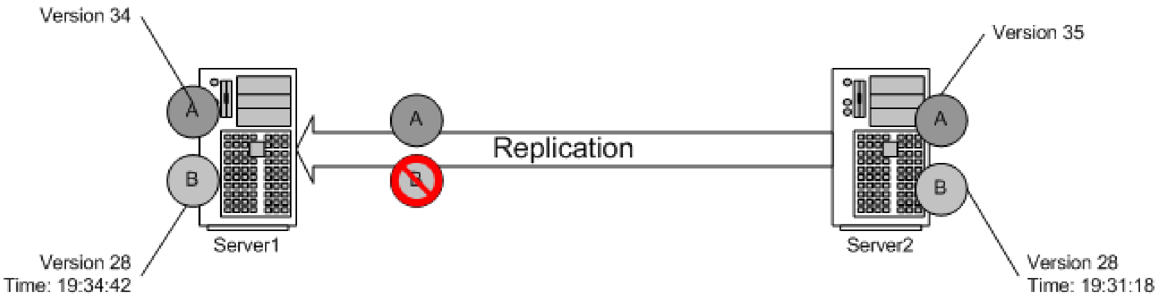


Figure 19.2: Replication collisions.

For more information about how AD handles replication conflicts, see the Microsoft article "How Conflicts Are Resolved in Active Directory Replication."

Propagation Dampening

The AD replication topology allows loops—particularly within a site in which a ring-based topology is the norm. In theory, these loops could result in change being replicated infinitely around the loop. However, AD replication has a built-in *propagation dampening* scheme, which detects and stops looping replication data.

The dampening system uses *up-to-date vectors.* The vector is a list of servers within the site and a USN for each server. The vector at each server thus indicates the highest USN of new changes received from each server within the site. When a new replication cycle starts, the requesting domain controller sends its up-to-date vector to its sending replication partner. That partner, as already described, filters the changes sent to the receiving domain controller so that it only receives changes made after that USN was used. If the requesting domain controller sends a USN that is greater than or equal to the sending domain controller's own USN, then no changes need to be sent, and the replication cycle stops. Figure 19.3 shows the vectors in action.

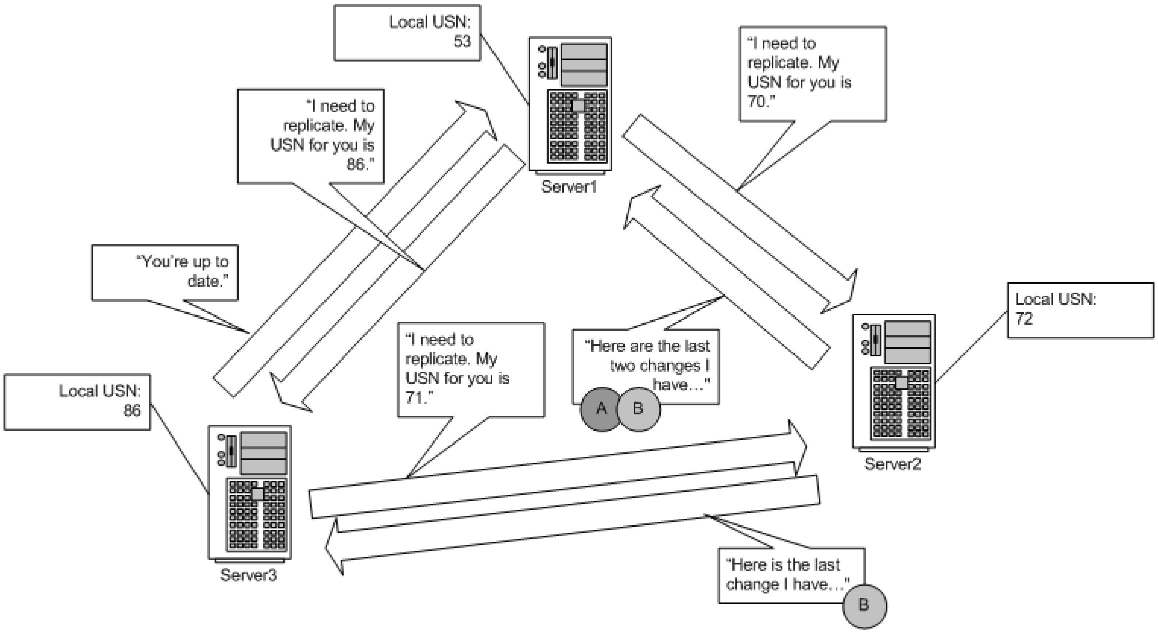

Local USN: 53
"I need to replicate. My USN for you is 86."
Server1
"I need to replicate. My USN for you is 70."
"You're up to date."
Local USN: 72
Local USN: 86
"I need to replicate. My USN for you is 71."
"Here are the last two changes I have..."
A
B
Server2
Server3
"Here is the last change I have..."
B

Figure 19.3: Propagation dampening in AD replication.

When Replication Occurs

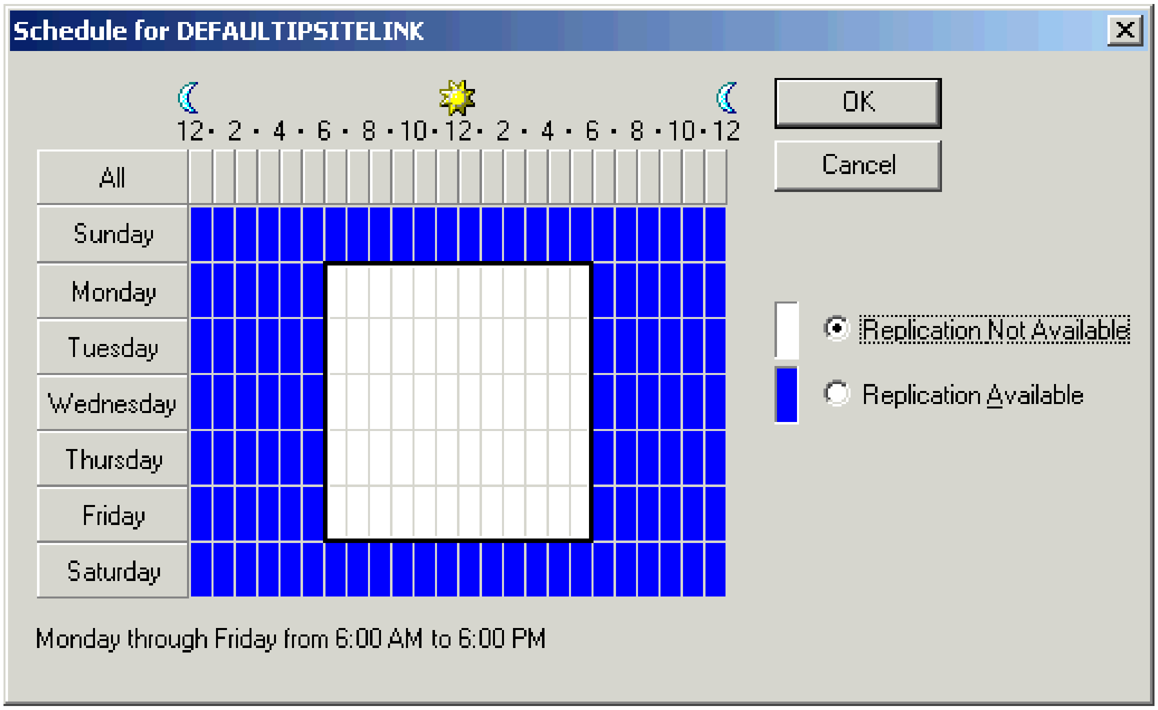
For intrasite replication, each domain controller informs its partners when it has changes available. Those partners then send a request for those changes, providing the domain controller with the USN numbers from their last replication update.

Domain controllers don't always inform their partners of changes as soon as those changes occur. For most changes, domain controllers wait about 5 minutes before sending a notification. This time period allows an administrator to make several changes and have them replicated in one batch, rather than the domain controller sending out independent change notifications for each minor change. However, certain types of security-sensitive changes—such as the lockout status of a user account—are considered high-priority and are always replicated immediately.

> 📖 For more information about high-priority replication triggers, see the Microsoft article "Urgent Replication Triggers in Windows 2000."

> 📖 You can modify the default intrasite replication interval. For details, refer to the Microsoft article "How to Modify the Default Intra-Site Domain Controller Replication Interval."

Intersite replication can occur less frequently, according to a schedule that you specify. This flexibility allows you to configure replication to best utilize your available intersite WAN bandwidth. Intersite replication traffic is compressed somewhat, helping to reduce WAN bandwidth utilization. Figure 19.4 shows how you can alter the schedule for a site link, for example, to prevent replication during business hours.


Schedule for DEFAULTIPSITELINK

● Replication Not Available
○ Replication Available

Monday through Friday from 6:00 AM to 6:00 PM

Figure 19.4: Restricting replication on a site link.

How Replication Travels

Replication can use either IP or Simple Mail Transport Protocol (SMTP) as a transport. IP is the standard intrasite transport and the default intersite transport. Replication actually uses IP to carry remote procedure call (RPC) traffic, uses Kerberos to mutually authenticate both replication partners, and uses Microsoft's CryptoAPI to encrypt the replicated data for maximum security.

The SMTP transport packages replicated data in packets, which allows for disconnected sites that have only mail-based connectivity. In effect, domain controllers email each other replication data. SMTP replication can only be used between sites, and can only be used between domain controllers in different domains (which still replicate certain forest-wide information to one another). SMTP requires the use of an enterprise certificate authority (CA), which allows the domain controllers participating in replication to authenticate one another through a trusted root CA.

Q.20: How can I check the replication topology for problems?

A: Active Directory (AD) replication is dependent entirely upon an accurate *replication topology*, a map of which domain controllers will exchange replicated changes with one another. The topology is generated by the AD Knowledge Consistency Checker (KCC), which generates a topology both for intrasite and intersite replication.

> 🕮 For details about how the KCC works, see Question 16.

Generally, the first symptom of topology problems is when one or more domain controllers, or an entire site, fail to replicate AD changes. You might, for example, notice that user password resets aren't being replicated properly or that new users and groups aren't consistently available throughout the domain. Often your first response to this problem is to use the Active Directory Sites and Services console to force replication; but doing so is useless if the underlying replication topology isn't correct.

> 🕮 For instructions about forcing replication, see Question 17.

Troubleshooting topology issues requires a methodical approach. If possible, start by determining whether you're dealing with an intersite or intrasite topology problem, as you'll need to troubleshoot them separately. To help make that determination, connect to a specific domain controller by using Active Directory Users and Computers. Make a change to AD, such as creating a new user group or organizational unit (OU). Check to see whether the change appears on another domain controller within the same site and within another site. Keep in mind that intrasite replication can take as long as 5 minutes or so to complete under normal conditions. Intersite replication is dependent upon the replication schedule you configured for the site links.

Intersite Replication

When intersite replication seems to be a problem, check the obvious first:

- Make sure your site links are configured with the correct replication schedule. If you have multiple site links between two sites, check the schedule on each. It's possible that one link has failed, and that AD was switched to an alternative link that uses a different schedule.
- Check the network connectivity between the two sites to make sure that your network isn't creating the problem.

Next, figure out which domain controllers are acting as the bridgehead servers in each site. Keep in mind that each domain in each site will have at least one designated bridgehead server. Sites with connections to multiple other sites might have multiple bridgehead servers per domain, with each bridgehead handling connections to another site. You can find the bridgehead servers by looking at the connection objects in Active Directory Sites and Services, and noting which domain controller is configured with connection objects to a domain controller in another site.

If you can't find a connection object on any domain controller in one site to a domain controller in another site and both sites contain domain controllers in the same domain, then you have a topology problem. Troubleshoot the intersite topology generator (ITSG).

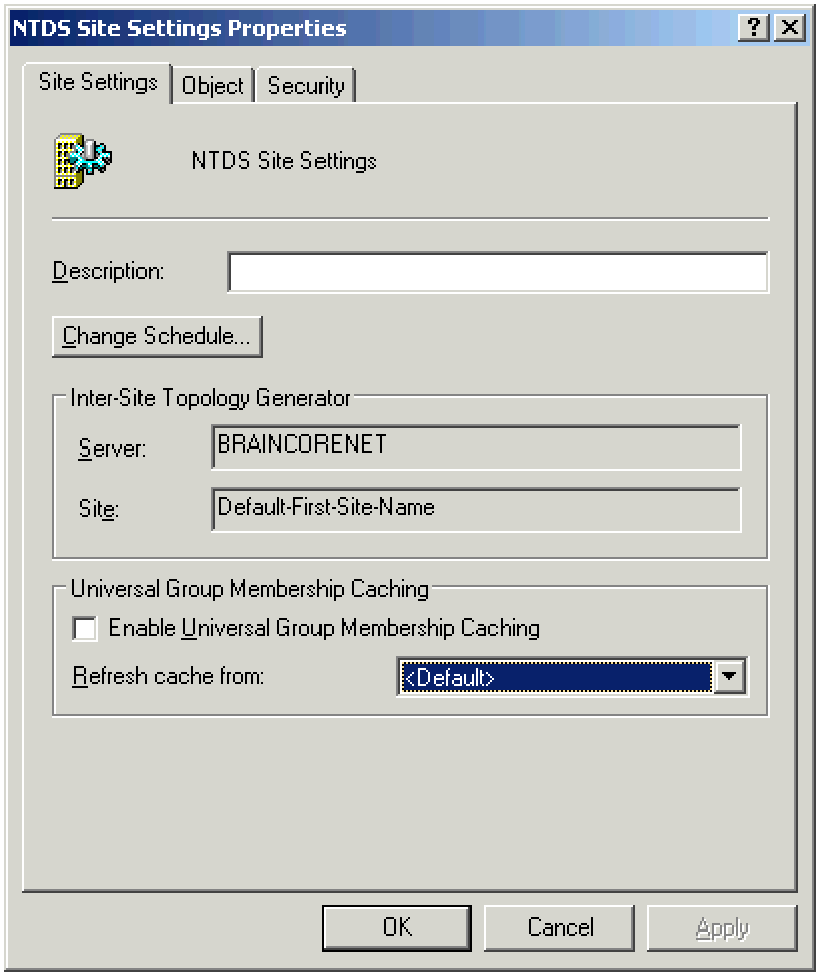
On each bridgehead server, ensure that you can ping the bridgehead servers at the other site(s). If you can't, a network issue is causing the problem and must be resolved.

If network connectivity between the bridgehead servers is OK, it's possible that the bridgehead servers aren't able to handle the replication traffic. Although this situation is rare, you can test the theory by manually creating connection objects to different domain controllers, thus creating new bridgehead connections. Delete the automatically configured bridgehead connection objects. If this action resolves the problem, the former bridgehead domain controllers are at fault and should be examined for functionality and capacity.

If no steps thus far have solved the problem or indicated a possible cause, you might have a serious replication problem. Check the System and Security event logs on your domain controllers for any events that might provide clues to the source of the problem.

ITSG

Each site has a designated ITSG, which generates the topology for that site. You can discover which domain controller is the ITSG by using Active Directory Sites and Services. To do so, open the console, select the site in question, and in the details pane, right-click NTDS Settings. As Figure 20.1 shows, you'll see the name of the server acting as ITSG.


NTDS Site Settings Properties
Site Settings
Object
Security
NTDS Site Settings
Change Schedule...
Inter-Site Topology Generator
Universal Group Membership Caching
OK
Cancel
Apply

Description:

Server: BRAINCORENET

Site: Default-First-Site-Name

☐ Enable Universal Group Membership Caching

Refresh cache from: <Default>

Figure 20.1: Server BRAINCORENET is the ITSG for this site.

After you've located the ITSG, ensure that it's functioning properly (services are all started and you can log on) and connected to the network. Next, force it to regenerate the intersite replication topology by running

```
repadmin /kcc
```

from the server's console. If the new topology doesn't solve your problem, consider taking the domain controller offline or restarting it. AD will choose a new ITSG within about an hour, and the new domain controller might have better luck generating the correct topology.

Intrasite Replication

The intrasite replication is generated by the KCC service running on each domain controller in the site. Intrasite replication generally occurs automatically and constantly throughout each domain in the site.

If a particular domain controller (or domain controllers) within the site don't seem to be replicating properly, check its replication partners. You can do so by running

```
repadmin /showreps
```

at each domain controller's console, or running

```
repadmin /showreps servername
```

from a single console, providing the appropriate *servername*.

Document the incoming and outgoing partners for each involved domain controller, and ensure that the domain controller has proper network connectivity (such as ping) to its replication partners. If network connectivity between any domain controller and one or more of its replication partners isn't available, troubleshoot and resolve that problem.

If network connectivity appears to be OK, try to force each involved domain controller to generate a new replication topology by running

```
repadmin /kcc
```

on each domain controller. This process normally occurs automatically every so often, but it's possible that a new topology problem hasn't yet been corrected by the automatic process.

If a new topology doesn't correct the problem, try restarting each domain controller involved. If that doesn't fix the problem, you have a more serious AD replication issue that does not involve the replication topology; check the System and Security event logs for messages that provide clues as to the source of the problem.