



Operating System

Chapter 9 Post Deployment Monitoring of Domain Controllers

Deployment and Operations Guide

Abstract

This chapter outlines how to monitor Active Directory and FRS to ensure your environment is functioning properly.

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 OR IMPLIED, 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.

© 2000 Microsoft Corporation. All rights reserved.

Microsoft, Windows, and Active Directory are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries/regions.

CONTENTS

INTRODUCTION	1
Resource Requirements	1
What You Will Need	1
What You Should Know	1
MONITORING CONSIDERATIONS	2
USING THE QUALITY ASSURANCE SCRIPTS TO MONITOR ACTIVE DIRECTORY AND FRS.....	3
The Quality Assurance Scripts	3
QA_Check.cmd	3
QA_Parse.vbs	3
CheckServers.cmd	3
CheckServers.vbs	3
Process for Using the Quality Assurance Scripts	4
Log Files Generated by QA_Check.cmd	4
GENERAL DOMAIN CONTROLLER MONITORING	7
Processor Utilization	7
Available Disk Space	7
Monitoring Domain Controller Performance	7
NTDS Object Counters	8
Useful NTDS Counters for Monitoring Active Directory	8
Database Object Counters	9
Useful Counters for Monitoring the Active Directory Database	10
Installing the Database Performance Object	11
Installing the Database Performance Object	11
Viewing Database Performance Object Counters	11
Monitoring FRS Performance	11
MONITORING ACTIVE DIRECTORY REPLICATION	13
What to Monitor	13
Using Netdiag.exe to Monitor Network Connectivity and DNS	13
Using Repadmin.exe to Monitor Active Directory Replication	14
/showreps	14
/showconn	14
Using Dcdiag.exe to Monitor Active Directory Replication	15
Using Replmon.exe to Monitor Active Directory Replication	15
MONITORING FRS REPLICATION	17
Examining the FRS Log Files	17
Configuring the FRS Log Files	18
Analyzing the FRS Log Files	18
Contents of the FRS Log Files	19
Using FRSUTL to Monitor FRS Replication	21
Scripts for Monitoring FRS Replication	22

Monitoring FRS Replication with Connstat.cmd	22
Monitoring FRS Replication with Frscheck.cmd	26
SUMMARY	27

INTRODUCTION

This chapter discusses the areas you should monitor in your Microsoft® Active Directory™ directory service branch office environment.

Resource Requirements

You will need to have operations staff to perform the ongoing monitoring of your branch office domain controllers.

What You Will Need

You will need operations staff who will be responsible for the monitoring and troubleshooting of Active Directory and File Replication service (FRS).

What You Should Know

To perform monitoring on your domain controllers, you will need an administrator user account and password.

MONITORING CONSIDERATIONS

After completing your deployment of Active Directory in your branch office environment, it is very important to continue to perform quality assurance checks on your domain controllers. Doing so will allow you to detect any potential problems before they have a chance to cause a significant impact on your environment and your users' ability to access network resources.

Quality assurance checks should be performed on your domain controllers on a regular basis. These checks should be performed at least once a day. If your branch office domain controllers are only replicating with the bridgehead servers in the hub site once per day, the daily quality assurance check should be performed after the replication interval. This will allow you to verify that replication was successful for the day, detect any issues that may have occurred, and allow you to correct any issues before the next replication interval. If the quality assurance check is performed before the daily replication interval, you may not become aware of any problems for up to 24 hours, which could allow the problem to have a larger impact on your environment.

There are three main areas that should be examined as part of your ongoing domain controller quality assurance:

- General domain controller monitoring
- Active Directory replication monitoring
- FRS replication monitoring

Each of these is covered in detail in this chapter. However, before looking at the details of these we will examine the quality assurance scripts included with this branch office guide.

USING THE QUALITY ASSURANCE SCRIPTS TO MONITOR ACTIVE DIRECTORY AND FRS

This Active Directory branch office guide includes a set of quality assurance scripts that can be used to perform a daily quality assurance check on your branch office environment. These scripts should be scheduled to run daily on your domain controllers—both the branch office domain controllers and your bridgehead servers in the hub site.

The Quality Assurance Scripts

There are four scripts that make up the quality assurance process. Each of these is described below, along with their interaction with each other.

QA_Check.cmd

This is the main script of the quality assurance process. It records the current state of a domain controller when it is run. This script uses a variety of Microsoft Windows® 2000 Resource Kit utilities and other scripts to obtain information about the domain controller, Active Directory replication, and FRS replication. For more details on what this script does, see the below section entitled “Checks Performed by QA_Check.cmd.”

Note: The QA_Check.cmd script uses the Ntfrsutl.exe utility from the Microsoft Windows 2000 Resource Kit to obtain FRS information. To obtain this FRS information, the script must be run with an administrator account if you are not running it as part of the scheduled quality assurance check.

QA_Parse.vbs

This script is called by the QA_Check.cmd script and parses the data files created by the utilities and scripts run as part of the quality assurance process. The script parses the data files to locate errors and potential issues, which are then written to a summary file. This summary file is then copied by the QA_Check.cmd script to a central server so that you have a single location to examine the state of your domain controllers.

CheckServers.cmd

This script is used on the central server that has the summary files from each domain controller to provide a status report on the health of the domain controllers in your environment. This script outputs a file that contains three lists of domain controllers:

- Domain controllers that are healthy and did not report any errors
- Domain controllers that reported errors and require further investigation
- Domain controllers that did not report and should be investigated

CheckServers.vbs

This script is used by CheckServer.cmd to perform the parsing of the summary files from your domain controllers and determine which list a domain controller should be placed into.

Process for Using the Quality Assurance Scripts

There are four steps to using the quality assurance scripts included with this guide. The steps are:

1. Schedule QA_Check.cmd to run on every domain controller in your environment. This should be scheduled to run every day, after any replication intervals.

QA_Check.cmd:

- Copies any files in C:\ADResults*computername* to C:\ADResults*computername*\old. Doing so provides a history of the state of the domain controller.
- Runs the Microsoft Windows 2000 Resource Kit utilities and other scripts. This generates a series of data files, one for each utility and script, and stores the data files in C:\ADResults.

Note: The data files stored in C:\ADResults are over written each time the QA_Check.cmd script is run.

- Runs the QA_Parse.vbs script to generate a summary report for the domain controller. The summary report uses the domain controller's computer name and the current date and month for the file name, for example: BO1DC-30-11.txt. This file is stored in C:\ADResults*computername*.
 - On the central server, copies the previous summary file to the \\<server>\QAShare*computername*\old folder.
 - Copies the new summary file to the central server specified in the QA_Check.cmd script. The file is copied to the \\<server>\QAShare*computername* folder.
2. Schedule CheckServers.cmd to run on the central server. This script should be scheduled to run every day, after all of the domain controllers are scheduled to run the QA_Check.cmd file. The CheckServers.cmd script:
 - Copies C:\QAShare\Serverreport.txt to C:\QAShare\old
 - Runs CheckServers.vbs to parse the summary files each domain controller copied to the central server, generating C:\QAShare\Serverreport.txt.
 3. Use Notepad to examine the contents of the C:\QAShare\Serverreport.txt file to determine if any domain controllers have reported errors or did not report in the last quality assurance cycle.
 4. If a domain controller reported errors, or did not report, you will need to investigate and resolve any errors. To start investigating errors, open the summary file under the C:\QAShare*computername* folder. From there, you can then examine the detailed files on the domain controller itself (discussed in more detail in the next section).

Log Files Generated by QA_Check.cmd

The QA_Check.cmd script uses utilities from the Microsoft Windows 2000 Resource Kit and several other scripts to record the state of a domain controller.

QA_Check.cmd generates a large amount of data. This is why the script creates the summary file using QA_Parse.vbs. However, if a domain controller is reporting errors, you will need to examine the data files generated by the script. This section

lists each of the data files that are generated and the tools used to generate the files.

QA_Check.cmd creates the following files in the C:\ADResults folder on each domain controller on which it is run:

File	Contents
Dcdiag.txt	Output of running Dcdiag.exe to perform domain controller diagnostic checks.
Netdiag.txt	Output of running Netdiag.exe to check the network configuration and health of the domain controller. When running Netdiag.exe, the Lightweight Directory Access Protocol (LDAP) tests are skipped as they can place a large load on the network when there is a large number of domain controllers.
GPOstat.txt	Output of running Gpostat.vbs to verify that each Group Policy object is in sync.
Ntfrs_ds.txt	Output of running Ntfrsutl.exe ds to list the FRS view of the DS.
Ntfrs_sets.txt	Output of running Ntfrsutl.exe sets to list the active replica sets.
Ntfrs_inlog.txt	Output of running Ntfrsutl.exe inlog to enumerate the FRS inlog.
Ntfrs_outlog.txt	Output of running Ntfrsutl.exe outlog to enumerate the FRS outlog.
Ntfrs_version.txt	Output of running Ntfrsutl.exe version to list the application programming interface (API) and service versions.
Ntfrs_reg.txt	Output of running Regdmp.exe to output the contents of the HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NtFrs\Parameters registry key.
Ntfrs_sysvol.txt	Output of running Dir %Systemroot%\sysvol /s to store a list of the contents of the SYSVOL folder.
Frsconstat.txt	Output of running Connstat.cmd to summarize the FRS connection state. For more details on what this script does, see "Monitoring FRS Replication with Connstat.cmd" later in this chapter.
Ntfrs_errscan.txt	Output of running Findstr to search the %windir%\debug\ntfrs_*.log files for "error", "invalid", "fail", "abort", and "warn".
Ntfrs_parse.txt	Output of running Findstr to search %windir%\debug\ntfrs_0005.log for "error", "invalid", "fail", "abort", and "warn".
Ntfrs_parse2.txt	Output of running Findstr to search %windir%\debug\ntfrs_0005.log for "ERROR - EXCEPTION (000006ba): WStatus: RPC_S_SERVER_UNAVAILABLE", "ERROR - STAGING AREA FULL", "ERROR - DISK_FULL",

	"ERROR_DISK_FULL", "ERROR - EXCEPTION EPT_S_NOT_REGISTERED", "has no inbound server", "has no outbound server", "DS: Multiple connections from", "WARNING: Setting FrsVsn - Current system Time has moved backwards from value in config record", and "JRNL_WRAP_ERROR".
Ds_showreps.txt	Output of running Repadmin /showreps to list the replication partners for the domain controller.
Ds_showconn.txt	Output of running Repadmin /showconn to list the connection objects for the domain controller.
Services.txt	Output of running Net Start to list the services that are running on the domain controller.

If the summary file for a domain controller indicates that it reported errors, the above log files should be examined to determine the specifics of the error that was reported. The additional information in these log files will aid you in troubleshooting the error. In addition to examining the above files, you should also examine the event logs in Event Viewer to see if any events were logged that are related to the error.

GENERAL DOMAIN CONTROLLER MONITORING

In addition to monitoring Active Directory and FRS on your domain controllers, it is also important to perform some more general monitoring of your domain controllers. Two important areas to monitor on your domain controllers are processor utilization and available disk space.

The performance counters discussed in this section should always be monitored on your bridgehead domain controllers. If you are experiencing problems with a branch office domain controller, you should monitor these counters to assist you in locating the source of the problem.

Processor Utilization

Monitoring the processor utilization on your domain controllers will allow you to determine if your domain controllers are being overloaded by logon or, on bridgehead servers, by replication. This will also allow you to verify that you are meeting your service level agreements.

To monitor a domain controller's processor utilization you can use System Monitor or Performance Logs and Alerts to monitor the Processor\ % Processor Time counter.

Available Disk Space

Monitoring available disk space is important as problems can arise with your domain controllers if the partition storing any of the following run out of available disk space:

- Active Directory database files
- Active Directory log files
- SYSVOL folder

By default, these are all stored in either C:\WINNT\NTDS or C:\WINNT\SYSVOL.

To monitor the free disk space on the partition containing your Active Directory database and log files and the SYSVOL folder use System Monitor or Performance Logs and Alerts to monitor the LogicalDisk\ Free Megabytes counter.

Monitoring Domain Controller Performance

In addition to monitoring the processor utilization and free disk space, you can also monitor domain controller performance by tracking performance counters in Performance Logs and Alerts and then viewing the results in System Monitor. For example, if you want to monitor whether a server is regularly receiving and applying directory replication updates, you can select one or more counters from the NTDS performance object, and then view the current activity in System Monitor.

Use the counters of the following two performance objects to monitor domain controller performance:

- NTDS object counters
- Database object counters

Note: Before you can use the Database performance object, you must install it manually. Instructions are provided later in this module.

The NTDS and Database counters should all show some activity when monitored over a period of time. However, the amount of activity will greatly depend on your environment. Factors that will affect the activity include the number of branch office domain controllers, number of clients, how often replication is scheduled, the number of directory changes that occur, and so on.

NTDS Object Counters

NTDS performance object counters enable you to monitor the performance of Active Directory. The NTDS performance object includes counters that provide information about Active Directory replication activity between domain controllers, LDAP, and authentication.

Useful NTDS Counters for Monitoring Active Directory

The following table describes useful counters for monitoring Active Directory.

Object\ Counter	Description	Guideline
NTDS\ DRA Inbound Bytes Total/sec	Indicates the total number of bytes (per second) received through replication. It is the sum of the number of bytes of uncompressed data and compressed data.	This counter should show activity over time. If it does not, it usually indicates that the network is slowing replication.
NTDS\ DRA Inbound Object Updates Remaining in Packet	Indicates the number of object updates received in the current directory replication update packet that have not yet been applied to the local server. This counter indicates that the monitored server is receiving changes, but is taking a long time applying them to the database.	This counter should be as low as possible. If it is not, it usually indicates that server hardware is slowing replication.
NTDS\ DRA Outbound Bytes Total/sec	Indicates the total number of bytes sent per second. This is the sum of the number of bytes of uncompressed data and compressed data.	This counter should show activity over time. If it does not, it usually indicates that either server hardware or network problems are slowing replication.
NTDS\ DRA Pending Replication Synchronizations	Indicates the number of directory synchronizations that are queued for this server that are not yet processed. This counter helps determine the	This counter should be as low as possible. If it is not, it usually indicates that server hardware is slowing replication.

	replication backlog—the higher the counter, the larger the backlog.	
NTDS\ DS Threads in Use	Indicates the current number of threads in use by the directory service.	This counter should show activity over time. If it does not, it usually indicates that network problems are hindering client requests.
NTDS\ Kerberos Authentications/sec	Indicates the number of Kerberos authentications (per second) serviced by the domain controller.	This counter should show activity over time. If it does not and the clients use Windows 2000, it usually indicates that network problems are occurring.
NTDS\ LDAP Bind Time	Indicates the time (in milliseconds) required for the completion of the last successful LDAP binding.	This counter should be as low as possible. If it is not, it usually indicates that hardware or network-related problems are occurring.
NTDS\ LDAP Client Sessions	Indicates the number of sessions of connected LDAP clients.	This counter should show activity over time. If it does not, it usually indicates that network-related problems are occurring.
NTDS\ LDAP Searches/sec	Indicates the number of search operations (per second) performed by LDAP clients.	This counter should show activity over time. If it does not, it usually indicates that network problems are hindering client requests.
NTDS\ LDAP Successful Binds/sec	Indicates the number of LDAP bindings (per second) that occurred successfully.	This counter should show activity over time. If it does not, it usually indicates that network-related problems are occurring.
NTDS\ NTLM Authentications	Indicates the number of NTLM authentications (per second) serviced by the domain controller.	This counter should show activity over time. If it does not and the clients use Windows 98 or Windows NT®, it usually indicates that network-related problems are occurring.

Database Object Counters

Database performance object counters enable you to monitor the Active Directory database at an advanced level. These counters provide information regarding the performance of the database cache, database files, and database tables. You can use some of these counters to determine whether you need more hard disks to

store additional Active Directory data.

Useful Counters for Monitoring the Active Directory Database

The following table describes useful counters for analyzing the Active Directory database.

Object\ Counter	Description	Guideline
Database\ Cache % Hit	Indicates the percentage of page requests for the database file that were fulfilled by the database cache without causing a file operation.	This counter should show activity over time. If it does not, it usually indicates that the server does not have enough free physical memory and you should consider adding more memory.
Database\ Cache Page Fault Stalls/sec	Indicates the number of page faults (per second) that cannot be serviced because there are no pages available for allocation from the database cache.	This counter should be zero. If it is not, it usually indicates that the server needs more memory.
Database\ Cache Page Faults/sec	Indicates the number of page requests (per second) for the database file that require the database cache manager to allocate a new page from the database cache.	This counter should be as low as possible. If it is not, it usually indicates that the server needs more memory.
Database\ File Operations Pending	Indicates the number of reads and writes issued by the database cache manager to the database file or files that the operating system is currently processing.	This counter should be as low as possible. If it is not, it usually indicates that the server needs more memory or processing power.
Database\ File Operations/sec	Indicates the number of reads and writes (per second) issued by the database cache manager to the database file or files.	This counter should be as low as possible. If it is not, it usually indicates that the server needs more memory.
Database\ Log Record Stalls/sec	Indicates the number of instances (per second) that a log record cannot be added to the log buffers because the buffers are full.	This counter should be as close to zero as possible. If it is not, it usually indicates that the server needs more memory and that the size of the log buffer may have become a bottleneck.
Database\ Log Threads Waiting	Indicates the number of threads waiting for data to be written to the log so that an	This counter should be as low as possible. If it is not, it usually indicates that the

	update of the database can be completed.	server needs more memory or a faster hard disk.
Database\ Table Open Cache Hits/sec	Indicates the number of database tables opened (per second) by using cached schema information.	This counter should be as high as possible. If it is not, it usually indicates that the server needs more memory.

Installing the Database Performance Object

The Database performance object monitors the Extensible Storage Engine (ESENT), which is the transacted database system that stores all Active Directory objects.

Installing the Database Performance Object

Because the Database performance object is not installed by default, you must use the performance dynamic-link library (DLL), `Esentprf.dll`, to install it.

To install the Database performance object:

1. Copy the performance DLL from `%SystemRoot%\System32\esentprf.dll` to a different directory. For example, create a directory named `C:\Performance`, and then copy the DLL and paste it in the new directory.
2. Run `Regedt32.exe` (or `Regedit.exe`), and then create the following registry subkeys if they do not already exist:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ESENT
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ESENT\Performance
```

3. Make sure that, under the Performance subkey, the registry values are set as follows:

```
Open : REG_SZ : OpenPerformanceData
Collect : REG_SZ : CollectPerformanceData
Close : REG_SZ : ClosePerformanceData
Library : REG_SZ : C:\Performance\esentprf.dll
```

4. Change the directory to `x:\Winnt\System32`, where `x` is the letter of the drive where Windows 2000 is installed.
5. To load the counter information into the registry, run:

```
Lodctr.exe Esentperf.ini
```

Viewing Database Performance Object Counters

After you have installed the Database performance object, you can view the counters by restarting System Monitor.

Monitoring FRS Performance

In addition to the Active Directory performance counters discussed above, there are also performance counters for monitoring FRS performance. The FRS performance counters are in two performance objects:

- **FileReplicaConn.** Contains counters for monitoring the performance of replica connections to Distributed file system (Dfs) roots.

- **FileReplicaSet.** Contains counters to monitor the performance of replica sets.

Within the FileReplicaSet performance objects, the following counters should be monitored to ensure that FRS is functioning properly. Each of the following counters should display activity when monitored over time.

Counter	Description
Change Orders Received	Indicates the number of change notifications received from inbound partners.
Change Orders Sent	Indicates the number of change notifications sent out to outbound partners.
File Installed	Indicates the number of replicated files installed locally.
Packets Received	Indicates the amount of data received locally. These packets can be change notifications, file data, or other command packets.
Packets Sent	Indicates the amount of data sent. These packets can be change notifications, file data, or other command packets.
USN Records Accepted	Indicates the number of records that are accepted for replication. Replication is triggered by entries written to the NTFS change journal. FRS reads each file close record from the journal and determines whether to replicate the file. An accepted record generates a change order, which is then sent out. A high value on this counter (about one every five seconds) indicates a lot of replication traffic, which can cause replication latency.

In addition to the counters in the above table, the following two FileReplicaSet counters should be monitored on a regular basis to ensure that a lack of disk space will not cause FRS issues:

- **KB of Staging Space Free.** Indicates the amount of free space in the staging directory used by FRS to temporarily store files before they are replicated.
- **KB of Staging Space in Use.** Indicates the amount of space in the staging directory currently in use. If the staging directory runs out of space, replication stops.

MONITORING ACTIVE DIRECTORY REPLICATION

An important aspect of any Active Directory deployment that is often overlooked is ongoing monitoring of the environment. Ongoing monitoring will allow you to detect any issues that may arise in your environment and correct them, hopefully before they impact your environment or users.

If ongoing monitoring is not performed on a regular basis, a problem could arise with a domain controller and you would be unaware of the issue until it started to impact users. By the time user problems are reported and you identify the cause of the issue, the problem could be having a larger impact on your environment than if it had been detected with an ongoing monitoring process.

What to Monitor

When monitoring Active Directory, the key areas to monitor include:

Area to Monitor	Utilities for Monitoring
DNS and Network configuration	Netdiag.exe
Connection objects	Repadmin.exe and Replmon.exe
Replication	Dcdiag.exe, Repadmin.exe, and Replmon.exe

If problems occur in any of the above areas, there will be an impact on Active Directory.

The following sections discuss the utilities that can be used to monitor these areas.

Using Netdiag.exe to Monitor Network Connectivity and DNS

The Netdiag.exe diagnostic tool can be used to isolate network and connectivity problems. Netdiag.exe performs a series of tests to determine the state of the network client and whether it is functional, as well as verifies DNS name registrations. These tests and the network status information Netdiag.exe provides can be used to identify and isolate network problems.

The recommended command line and switches for running Netdiag.exe to monitor your Active Directory environment is:

```
netdiag /v
```

The /v switch provides verbose output.

This command is included in the QA_Check.cmd script with the addition of the /skip:LDAP switch. The LDAP tests performed by Netdiag.exe generate a large amount of network traffic as it attempts to contact every domain controller in your environment. Since the QA_Check.cmd script runs daily on every domain controller, performing the LDAP test would place too large of a load on a branch office network with slow links. However, Netdiag.exe should be run at least once a week without the /skip:LDAP switch to ensure that there are no LDAP issues.

You should troubleshoot any errors found by Netdiag.exe, using the information

provided by the Netdiag.exe output to help isolate the problem. For more information on specific errors, see Chapter 11, "Troubleshooting Guidelines for Branch Office Environments," of this guide.

Using Repadmin.exe to Monitor Active Directory Replication
Repadmin.exe, also referred to as the Replication Diagnostics tool, can be used to diagnose replication problems. Repadmin.exe allows administrators to view a domain controller's perspective of the replication topology. In addition, Repadmin.exe can be used to force replication events between domain controllers and to view both the replication metadata and up-to-dateness vectors.

There are two Repadmin.exe switches that should be used on a regular basis on each domain controller: /showreps and /showconn. The QA_Check.cmd script runs Repadmin.exe with both of these switches.

/showreps

This switch displays the replication partners, both inbound and outbound, for each naming context that is on the specified domain controller. Examining the replication partners will allow you to determine if the domain controller has the correct connection objects. For each replication partner, /showreps also displays the last time replication was attempted and whether or not the attempt was successful, for example:

```
DC=branches,DC=corp,DC=hay-buv,DC=com
HUB\HUBDC1 via RPC
  objectGuid: fe641acc-3d4e-48a9-ada6-209e5329feef
  Last attempt @ 2000-12-02 07:09.44 was successful.
```

/showconn

This switch displays the connection objects on the current domain controller. Examining the connection objects will allow you to determine if the domain controller is configured to replicate with the correct bridgehead servers in the hub site. In addition, the information returned can be used to verify that the connection is enabled, the transport being used, when the connection object was created, and when it was last changed. For example:

```
CN=Staging, CN=Sites, CN=Configuration, DC=corp, DC=hay-buv, DC=com:
POS\HUBDC1 to POS
  enabledConnection: TRUE
  fromServer: HUB\HUBDC1
  TransportType: IP
  whenChanged: 20001130071554.0Z
  whenCreated: 20001130070741.0Z
```

If you think a domain controller does not have the correct replication partners, you should examine the Mkdsx.dat file to determine who the replication partners should be. You should then use Active Directory Sites and Services to examine the incorrect connection objects to see if you can determine how they were created. Rerunning the Mkdsx script will remove any connection objects that should not be part of your topology and recreate any valid connection objects that may be missing.

Using Dcdiag.exe to Monitor Active Directory Replication
Dcdiag.exe is a utility that can be used to analyze the state of a domain controller and report any problems. Dcdiag.exe performs a series of tests to verify different areas of the system, these tests include:

- Connectivity
- Replication
- Topology Integrity
- Check NC Head Security Descriptors
- Check Net Logon Rights
- Locator Get Domain Controller
- Intersite Health
- Check Roles
- Trust Verification

There are three switches that should be used with Dcdiag:

- **/v**. Provides verbose results
- **/f:LogFile**. Redirects output to the specified log file
- **/ferr:ErrLog**. Redirects fatal error output to a separate log file

When using the /v to provide verbose results, the output from Dcdiag will provide a great deal of information, which makes troubleshooting any errors that are found easier. In addition, Dcdiag provides for each test a summary line at the end of the information about the test that indicates whether or not the test passed or failed. An example of the output from a Dcdiag test is:

```
Starting test: RidManager
* Available RID Pool for the Domain is 6603 to 1073741823
* hubdc1.branches.corp.hay-buv.com is the RID Master
* DsBind with RID Master was successful
* rIDAllocationPool is 4603 to 5102
* rIDNextRID: 4605
* rIDPreviousAllocationPool is 4603 to 5102
..... POS passed test RidManager
```

The QA_Check.cmd script uses the following command line to execute Dcdiag.exe:

```
Dcdiag /s:%computername% /v /f:C:\ADResults\Dcdiag.txt
/ferr:C:\ADResults\Dcdiagerr.txt
```

You should troubleshoot any errors found by Dcdiag, using the information provided by the Dcdiag output to help isolate the problem. For more information on specific errors, See Chapter 11, "Troubleshooting Guidelines for Branch Office Environments," of this guide.

Using Replmon.exe to Monitor Active Directory Replication
Replmon.exe can be used to view the low-level status of Active Directory replication, force synchronization between domain controllers, view the topology in a graphical format, and monitor the status and performance of domain controller replication through a graphical interface.

Because Replmon.exe has a graphical interface, it is not used by the QA_Check.cmd script to monitor domain controllers. However, some administrators

may want to use Replmon.exe when examining the cause of any errors reported in the summary file generated by QA_Check.cmd. Two useful features of Replmon.exe for this purpose are:

- **Generate Status Report.** This option generates a status report for the monitored server that includes: a list of the directory partitions for the server, the status of the replication partners for each of the directory partitions, detail on which domain controllers the monitored server notifies when changes have been recorded, the status of any Group Policy objects, the domain controllers which hold the operations master roles, a snapshot of the performance counters on the computer, and the registry configuration of the server.
- **Show Replication Topologies.** This option displays a graphical view of the intra-site topology and can be used to display the properties of the server and any intra-site and inter-site connections that exist for that server.

MONITORING FRS REPLICATION

FRS is a multithreaded replication engine used to replicate files between different computers simultaneously. When you add, remove, or modify the contents of the SYSVOL folder on a domain controller, those changes are replicated by FRS to the SYSVOL folders on all other domain controllers in the domain.

FRS uses the same connection objects as Active Directory when replicating SYSVOL content. Therefore, it uses the same schedule as Active Directory for intersite replication.

It is very important to monitor FRS replication and ensure that it is functioning in your environment. If FRS is having problems replicating to domain controllers and you are using Group Policy, Group Policy changes will not replicate to the domain controllers that are experiencing replication problems.

Unfortunately there are not very many tools for monitoring FRS replication. There are three methods that can be used to monitor FRS replication:

1. A pragmatic approach is to copy a “tag file” to the SYSVOL share. After the next replication interval for a domain controller’s replication partners, you can check the SYSVOL share of the replication partners to see if the “tag file” replicated successfully.
2. Examine the FRS log files for errors. The log files generated by FRS are a comprehensive way to follow the actions performed and any problems encountered by FRS.
3. Use Ntfrsutl.exe from the Microsoft Windows 2000 Resource Kit to view FRS information.

The QA_Check.cmd script uses methods 2 and 3 to monitor the FRS service for problems. Both of these methods are described in more detail below.

Examining the FRS Log Files

FRS creates text-based log files in the `%systemroot%\Debug` folder to help debug FRS problems. To observe a particular event, you should take a snapshot of the log files as close to the occurrence of the event as possible. Save the log files in a different location so they can be examined afterward.

By default, the FRS log files store transaction and event detail in sequentially numbered files: Ntfrs_0001 through Ntfrs_0005. Transactions and events are written to the log files with the highest version number in existence at that time. The Ntfrsapi.log file contains events that take place during promotion and demotion—namely, creating the FRS subkeys in the FRS registry key.

To capture a random or intermittent event, you might want to expand the number of FRS log files. For example, you can increase the number of log files to 50 and then archive the files when they become full. This may help to accumulate the history necessary to respond to overnight queries from users and locate a problem.

Depending on the problem that is being investigated, it might be necessary to review FRS logs on both the inbound and outbound replicas. If so, it is important to

ensure that the system clocks are synchronized between the two servers so that events can be correlated between replication partners.

Finally, the recovery setting for the FRS service in Service Control Manager (SCM) can be critical to locating and keeping important log events on the system. If the service is asserting, but SCM is configured to automatically start FRS upon error, enough log traffic might be generated to cause events in Ntfrs_0005.log to decrement and be deleted from the drive. Stop the service on both the inbound and outbound replicas close to the time when an error occurs, and then copy the logs to another location for analysis.

Configuring the FRS Log Files

The characteristics of the log files are determined by the values of several registry entries in the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtFrs\Parameters subkey. The following registry entries can be used to configure how the FRS log files are generated:

- **Debug Log Files.** Specifies the number of log files to create. After the number of log files has been filled, the lowest log version is deleted and the remaining log file names are decremented by 1 to make room for a new log file. Note, if you change this value in the registry, you must edit the QA_Check.cmd file and change the number five in all references to Ntfrs_0005.log to the number specified for this value.
- **Debug Log Severity.** Specifies the level of detail in the FRS log file. The level of detail can range from 0 to 5, with 5 providing the most detail. If this value is not present, the default is 2.
- **Debug Maximum Log Messages.** Specifies the log file size, in lines. The default value of 10,000 lines in a log files that are approximately 1 MB in size.

To change the quantity, size, or level of detail of FRS log files, edit the values of the registry entries. Before you increase either the size or quantity of log files, make sure sufficient disk space is available.

When changing these registry entries, you should stop the FRS service, edit the registry, and then restart the FRS service for the changes to be used immediately .

Analyzing the FRS Log Files

The first step to resolving problems using the FRS log files is to make sure the Debug Log Severity entry in the registry is set high enough to capture the events needed to identify the problem.

Next, identify errors, warning messages, and milestone events in the log files. A good practice is to start at the bottom of the last log file and work your way up. Focus on keywords such as "install," "success," and "fail." If you do not find the error that you are looking for, start at the bottom of the previous log (Ntfrs_0005.log, then Ntfrs_0004.log, and so on). Use the Find or Findstr commands to isolate errors in the log files as follows:

```
find /in "error warn fail" ntfrs*. * >err.tmp
```

Note: The QA_Check.cmd script contains several Findstr commands to examine the FRS log files for certain messages. Examine the QA_Check.cmd script for further examples of messages to search the FRS log file for.

Depending on the context, some errors, such as "jet attach db – 1811. Db not found", can be ignored because the Ntfrs.jdb file does not exist the first time FRS starts. Until the service creates the file, you will see this error immediately after Dcpromo or when you delete the Ntfrs.jdb file manually. Sharing violations, designated by the SHARING_VIOLATION message, occur when a user or process has a lock on a file. Because FRS tracks only closed files, locked files and directories do not replicate.

If you find failure errors in the FRS log files, examine the thread number and follow up all events in the log that have matching thread identifiers until you see the associated change order. To determine why a file on Server A has not replicated to a second or third replica, locate the ":: COG" number in the Ntfrs_00n.log files on the originating server. Search for the same globally unique identifier (GUID) in the logs on the second and third replicas. For the ":: COG" entries to appear in the FRS log files, the Debug Log Severity registry value must be set to three or higher.

Contents of the FRS Log Files

There are a variety of identifiers that appear in the FRS log files. The following table lists the important identifiers and provides a description in order to help you interpret the contents of the FRS log files. In some cases, only the first DWORD of a GUID is displayed in the log file.

Identifier	Description
:T:	Identifying String.
CoG:	Change Order GUID - Uniquely identifies a create/delete/rename/modify action for a file.
CxtG:	Connection GUID - Identifies the connection object in the topology connecting an upstream computer to the computer that delivered this change order.
[] - RemCo	Identifies a remote change order.
[] - RemCo, Abort	Identifies a remote change order that was aborted.
[] - LclCo	Identifies a local change order.
[] LclCo, Abort	Identifies a local change order that was aborted.
Name:	File name.
EventTime:	Time on the originating member, at which the change was performed.
Ver:	Version number of the file. Increases by one each time a local change order is created.
FileG:	File GUID - Uniquely identifies the file or directory and is used as the

	NTFS object ID on the file or directory. The corresponding file/directory on each replica member have the same File GUID.
FID:	File ID - The NTFS volume-specific file ID (also known as File Reference Number).
ParentG:	Parent GUID - The GUID of the parent directory that contains this file or directory.
Size:	The approximate size of the file or directory, noted in hexadecimal.
OrigG:	Originator GUID - The GUID associated with the member of the replica set that originated this update.
Attr:	File Attributes - The attribute flags for the file or directory.
LocnCmd:	Location Command - One of the following: Create, Delete, NoCmd, MoveDir; indicating that the file is being created, deleted, updated, or is changing parent directories.
State:	The change order state - One of the following: IBCO_STAGING_RETRY, IBCO_FETCH_RETRY, IBCO_INSTALL_RETRY, IBCO_COMMIT_STARTED; indicating that the change order is being retried later because of insufficient staging space, inability to complete the fetch of the staging file, or inability to install the change to the file. Finished change orders have a state of IBCO_COMMIT_STARTED.
ReplicaName:	The name of the replica set containing this file or directory.
CoFlags:	Change Order Flags: Abort – Set when CO is being aborted. VVAct – Set when VV activate request is made. Content – Valid content command. Locn – Valid location command. LclCo – CO is locally generated. Retry – CO needs to retry. InstallInc – Local install not completed. Refresh – CO is an upstream-originated file refresh request. OofOrd – Don't check/update version vector. NewFile – If CO fails, delete IDTable entry. DirectedCo – This CO is directed to a single connection. DemandRef – CO is a downstream demand for refresh. VVjoinToOri – CO is from vvjoin to originator. MorphGen – CO generated as part of name morph resolution. MoveinGen – This CO was generated as part of a sub-dir MOVEIN. OidReset – All CO did was reset OID back to FRS-defined value. CmpresStage – The stage file for this CO is compressed.
UsnReason:	Flags set in the NTFS change log describing modifications to the file. Close – Change log close record.

	<p>Create – File or directory was created.</p> <p>Delete – File or directory was deleted.</p> <p>RenNew – File or directory was renamed.</p> <p>DatOvrWrt – Main file data stream was overwritten.</p> <p>DatExt – Main file data stream was extended.</p> <p>DatTrunc – Main file data stream was truncated.</p> <p>Info – Basic info change (attrib, last write time, etc.).</p> <p>Oid – Object ID change.</p> <p>StreamNam – Alternate data stream name change.</p> <p>StrmOvrWrt – Alternate data stream was overwritten.</p> <p>StrmExt – Alternate data stream was extended.</p> <p>StrmTrunc – Alternate data stream was truncated.</p> <p>EACHg – Extended file attribute was changed.</p> <p>Security – File access permissions changed.</p> <p>IndexableChg – File change requires re-indexing.</p> <p>HLink – Hard link change.</p> <p>CompressChg – File compression attribute changed.</p> <p>EncryptChg – File encryption changed.</p> <p>Reparse – Reparse point changed.</p>
--	---

Using FRUSUTL to Monitor FRS Replication

The Ntfrsutl.exe tool can be used to do the following:

- Show the ID table, inbound log, or outbound log for a computer hosting FRS.
- Examine memory usage by FRS.
- Show the FRS configuration in Active Directory.
- List the active replica sets in a domain.
- List the API and version number for FRS.
- Poll immediately, quickly, or slowly for changes to the FRS configuration

Note: To access some of the information reported by Ntfrsutl, the user running the utility must be logged on as an administrator.

The following switches are useful for monitoring FRS with Ntfrsutl:

- Ds. Lists the FRS service's view of the directory service
- Sets. Lists the active replica sets
- Inlog. Lists the FRS service's inbound log
- Outlog. Lists the FRS service's outbound log
- Version. Lists the API and service versions for FRS

The QA_Check.cmd script runs Ntfrsutl with each of the above switches, creating a separate log file for each switch.

Scripts for Monitoring FRS Replication

In addition to the QA_Check.cmd script included with this guide, there are two scripts that can be used to monitor FRS. These scripts are Connstat.cmd and Frscheck.cmd.

Monitoring FRS Replication with Connstat.cmd

The Connstat.cmd script processes the output of the "ntfrsutl sets" command to generate a summary of the FRS connections for a given domain controller. The report created by Connstat.cmd consists of three parts: the header, the state of the inbound connections, and the state of the outbound connections. This script is called by the QA_Check.cmd script, which redirects its output to C:\ADResults\Frsconstat.txt.

Report Header

The report header contains the following information:

```
Processing file C:\ADResults\ntfrs_sets.txt    Modify Time: Sat Dec  2
07:17:22 2000
  Replica: DOMAIN SYSTEM VOLUME (SYSVOL SHARE) (b0513a54-b248-492b-
96f475d9fec62804)
    Member: STAGING           ServiceState: 3 (ACTIVE)
OutLogSeqNum: 319    OutlogCleanup: 319    Delta: 0
```

The first line prints the input file name followed by the time it was last modified. It is recommended that you include the name of the server as part of the filename. The next line shows the replica set name and GUID. The third line shows the member name (which for DFS replica sets is often a GUID), the state of the FRS service, the current outbound log sequence number, the outbound log sequence number where the next cleanup pass will begin, and the difference between these two numbers. This last value provides an approximate count of the number of change orders currently present in the outbound log. It is approximate because in some cases intervening change orders may have been deleted.

The FRS service state is one of the following:

State	Meaning
REPLICA_STATE_ALLOCATED	Replica set is in an initializing state
REPLICA_STATE_INITIALIZING	Replica set is in an initializing state
REPLICA_STATE_STARTING	Starting the replica set
REPLICA_STATE_ACTIVE	Replica set is now active
REPLICA_STATE_STOPPED	Replica set is now stopped
REPLICA_STATE_ERROR	Replica set is stopped due to an error
REPLICA_STATE_JRNL_WRAP_ERROR	Replica set is stopped due to data loss in the NTFS journal
REPLICA_STATE_REPLICA_DELETED	Replica set is marked as deleted

Inbound Connections

The section on the inbound connections is next. It appears as follows:

Partner	I/O	State	Rev	LastJoinTime
<Jrnl Cxtion>	In	Joined	0	
BRANCHES\HUBDC1\$	In	Joined	3	

The first column, Partner, is the name of the connection's partner. It consists of the domain name followed by the server name. The I/O column describes the connection as either inbound or outbound. The State column is the current state of this connection. The Rev column is the minor rev level of the partner's communication protocol. LastJoinTime is the time that this member last joined with the corresponding partner.

The partner name "<Jrnl Cxtion>" refers to the local FRS journal, which serves as another inbound partner.

The Connection State can be one of the following:

Connection State	Meaning
INIT	Newly allocated
UNJOINED	Not joined to partner
START	Inbound partner has requested join
STARTING	Starting the join
SCANNING	Scanning the inbound log
SENDJOIN	Scan complete, send join request to partner
WAITJOIN	Sent request, waiting for partner's reply
JOINED	Joined with partner
UNJOINING	Draining remote change orders through retry
DELETED	Connection has been deleted

A "connection" is initially created in the INIT state and then goes to the UNJOINED state. From there, when the schedule allows, it goes to the STARTING state when a StartJoin request is sent to the inbound log subsystem. When inlog starts the request it advances the state to SCANNING. When it has scanned the inbound log for the replica set and has requeued any change orders from this inbound partner's connection it advances the state to SENDJOIN. The Replica control subsystem then picks it up as part of its retry path, does a one-time INIT, sends the JOIN request to the inbound partner, and advances the state to WAITJOIN. Once the join request is completed, the state goes to JOINED if it succeeded or to UNJOINED if it failed.

A quick method to determine if a partner understands compression is the Rev level. A Rev level 3 connection does not generate or understand compressed staging files. A Rev level 4 connection generates compressed staging files for other Rev level 4 partners and uncompressed staging files for down rev partners. A Rev level 4 partner understands both compressed and uncompressed staging files from its inbound partners.

To participate in a valid FRS replica set each member **MUST** have at least one inbound partner.

Outbound Connections

The section on outbound connections is next. It appears as follows:

```
Member: STAGING ServiceState: 3 (ACTIVE)
OutLogSeqNum: 319 OutlogCleanup: 319 Delta: 0

Send Cleanup Cos
Partner I/O State Rev LastJoinTime
OLog State Leadx Delta Trailx Delta LMT Out Last VVJoin

BRANCHES\HUBDC1$ Out Joined -vv 3
OLP_AT_QUOTA 422352 146 422225 7552 127 7 Sat Dec 2 2000
07:17:22
```

The leftmost portion up through the LastJoinTime is mostly the same as described above for the Inbound connections. The only difference is the appearance of the tag "-vv" in the State column. If this tag is present, it means that this connection is in the middle of doing a Version Vector-based join operation. Typically this is done when the outbound partner is first added to the replica set or if it is in the process of doing a non-authoritative restore (a D2). Once this initial sync is complete, the connection leaves the VVJoin state. This only applies to outbound connections.

The rest of the report contains the outbound log-related state for this connection. The OLog state describes the state of outbound log processing for this connection. It is one of the following:

OLog State	Meaning
OLP_UNJOINED	The partner is not joined
OLP_ELIGIBLE	The partner can accept further change orders (Joined and change orders [COs] Out < Max)
OLP_STANDBY	The partner is ready to join the eligible list
OLP_AT_QUOTA	The partner is at max quota for outstanding change orders
OLP_INACTIVE	The partner is not accepting change orders

Note: OLP_AT_QUOTA is a normal condition when FRS is actively replicating because it always tries to keep eight change orders outstanding (see COs Out, later in this chapter) on each joined outbound connection. However, the combination of OLP_AT_QUOTA and an LMT value of 127 may indicate a problem.

Leadx column is the index into the outbound log for the next change order to be processed for this connection. Send Delta is the difference between OutLogSeqNum (index of the most recent change order in the outlog) and the Leadx value for this connection. So this is the number of change orders that remain to be processed by this connection. A Send Delta of zero means that this connection is fully synchronized at this time. Unless there is very little activity in the replica set or the connection schedule is "always on" you are unlikely find connections with a Send Delta of zero.

Trailx column is the index into the outbound log of the oldest unacknowledged outbound change order. Each change order sent to the partner for this connection must eventually be acknowledged. The change orders are not necessarily acknowledged in the order they were sent. There is a 128-bit sliding ack window

used to track which change orders have been acknowledged. This means that up to 127 subsequent change orders can be sent following the oldest unacknowledged change order before the 128-bit AckVector wraps, forcing FRS to stop sending change orders to this particular partner. Cleanup Delta is the difference between the Trailx value for this connection and OutLogCleanup index, the latter being the minimum Trailx value for all outbound connections. So Cleanup Delta is the count of change orders that have now been processed by this connection and can be retired from the outlog. However, since all outbound connections share the same outbound log, the cleanup phase must stop at the Trailing Index of the slowest (or farthest behind) connection. Until an unjoined partner either (1) joins with this member to resume replication, or (2) its connection object to this member in the directory service is deleted, no further outlog cleanup operations (which will free the related staging file space) can be performed. So those partners having connections with small cleanup deltas should be examined carefully to verify their operational status.

The LMT column is the difference between the values in the Leadx and Trailx columns. This maximum value is limited by the size of the AckVector described above. A value of 127 in this column indicates that this connection may be hung. This is caused by FRS on the partner failing to properly acknowledge a change order. It could also be a normal condition caused by the transmission of a very large file followed by change orders for very small files. Once the transmission of the large file is completed, the outbound partner will acknowledge the change order and the condition will clear itself. At this point you can either wait for a while and see if the condition changes or study the FRS debug logs on the outbound partner to see if it is making progress fetching staging file data. All servers should be running Service Pack 1 or later to avoid at least one known case where a FRS connection can hang. If you conclude that the connection is hung, then first try to delete and recreate the related connection object in the directory service. If the condition recurs, then you will have to run a non-authoritative restore (D2) on the outbound partner.

The COs Out column is the number of active unacknowledged change orders pending at the partner. By default, FRS allows up to eight change orders to be outstanding at a time on each outbound connection. This is controlled by a registry parameter, which you may want to increase in an environment with high latency communication links. Note that FRS on the partner will try to initiate fetch requests for staging files for all eight change orders concurrently, so increasing this registry parameter will increase the load on the upstream member. The registry key is

"Max Num Outbound COs Per Connection".

It is a DWORD with a maximum value of 100. A COs Out value of 1 combined with an LMT value of 127 is very likely a hung connection (if not a very large file).

The Last VVJoin column is the date/time that the last time a Version Vector-based join was done on the connection.

Running CONNSTAT.CMD

Connstat.cmd is a perl script with a cmd wrapper. You need to have Active Perl

from the Microsoft Windows 2000 Resource Kit installed and in your path to run it.

The usage information is:

```
connstat [-sort=xxx] datafile

-sort=send    -- sort outbound connections by the send delta
-sort=clean   -- sort outbound connections by the cleanup delta
-sort=name    -- sort outbound connections by the server name (default)
-sort=lmf     -- sort outbound connections by the leading minus trailing
              index value.
-sort=lastjointime -- sort outbound connections by the last join time.
-sort=lastvvjoin  -- sort outbound connections by the last version
              vector join time.
```

The output report is written to stdout. The *-sort* parameter lets you sort the outbound connection section by the data in the related column.

Monitoring FRS Replication with *Frscheck.cmd*

The *Frscheck.cmd* script captures most of the FRS-related state for a machine. It uses *Nfrsutl.exe* to gather the FRS version, sets, directory services, configtable, inlog, and outlog information. It gathers the registry parameters and it scans the FRS debug logs for error information.

Note: To access some of the information reported by *Nfrsutl* the user running the utility must be logged on as an administrator.

The usage for *Frscheck.cmd* is:

```
frscheck result_dir [target_computername]
    result_dir is created if it does not exist.
    Target_ComputerName is optional. Default is current computer.
    It can be a netbios name with no leading \\ or a full dns name,
    xxx.yyy.zzz.com
```

Frscheck.cmd requires that delayed environment variable expansion be enabled at the command prompt. You can enable or disable delayed environment variable expansion for a particular invocation of *Cmd.exe* with the */v:ON* or */v:OFF* switch. You can enable or disable completion for all invocations of *Cmd.exe* on a computer by setting the following *REG_DWORD* value to either 1 or 0 in the registry by using *Regedt32.exe*:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Command Processor\DelayedExpansion
```

SUMMARY

This chapter has focused on the scripts and other tools available for use in the monitoring of Active Directory and FRS. Information on the scripts and their output was provided, along with detailed information on the Microsoft Windows 2000 Resource Kit tools was provided. In addition, information on third party tools was provided.