

Chapter 6: Troubleshooting and Managing Group Policy

The final chapter of this book is devoted to keeping your Group Policy infrastructure up and running after you've applied everything I've described in chapters 1 through 5. This chapter will provide you with a "kit" of tools and techniques to keep your Group Policy Objects (GPOs) running and to determine why they're not when something goes wrong. Most of the tools I describe are available in the Windows 2000 Resource Kit, in the Windows 2000 Support Tools, or from Microsoft's Web site.

This chapter will fill in some gaps left when Microsoft released Windows 2000 (Win2K) and Group Policy. Plenty of policy functionality was built into the product, but very few management and troubleshooting tools were provided out-of-the-box. For example, there is currently no way in the product to create, delete, or edit a GPO using a script, although you can write a script to link GPOs to Active Directory (AD) containers. You also can't back up or restore individual GPOs. And there is no single interface or log that you can view to find out about problems when GPOs aren't working as expected.

As of this writing, there are very few third-party products on the market to help you manage your GPO deployments. One tool that *does* exist, and that I'll talk about later in this chapter, is FullArmor's Zero Administration for Windows 2000 (FAZAM 2000), which provides some of the missing capabilities. (A Reduced Functionality Version, called FAZAM 2000 RFV, is available in the Windows 2000 Resource Kit.) As GPOs become a more widely deployed technology in Windows infrastructures, I look forward to seeing other vendors step up to the plate and provide more value.

Examining Common GPO Problems

GPO problems can be divided into two general categories—those that result from "operator error" and those that are caused by a breakdown in Win2K or AD. The former includes things like not realizing that you've disabled a GPO, applying security filters to a GPO in a way that the intended user or computer no longer receives it, and overwriting the effects of one GPO with the settings from another that is processed after the first. These kinds of problems are easily mitigated by having good system documentation, a well-controlled delegation model (discussed in Chapter 5), and change control on your GPO infrastructure. Even though Group Policy doesn't come with a structured version-control mechanism, I described a way to "fudge" it in Chapter 4 by simply adding a version number to the display name of the GPO.

It's the second class of problems that will be the focus of most of this chapter—those that result from a breakdown in one of the moving parts in a GPO infrastructure. As I think I've shown, GPOs are complex; to perform as expected, several pieces of technology must be synchronized. Knowing where to look when things go awry and what tools to use will go a long way towards helping you keep your GPO infrastructure healthy.

I discuss tools in the next section (see "Using GPO Troubleshooting Tools"). Here, I'll present a troubleshooting matrix (Table 6.1 below) that talks about some of the problems you're likely to encounter, their most likely causes, and potential solutions.

This Problem	Has This Probable Cause	And These Potential Solutions
One or more GPOs hang when a user logs on or a computer starts up.	There are a number of reasons why a GPO will hang, but it often happens in conjunction with script policy. You might have entered a bad path to a script file in logon/logoff or shutdown/startup script policy. Or a script may be poorly written and caught in some kind of loop or timeout condition.	Check all of your paths to the script files in the Group Policy Templates (GPTs) for these GPOs to ensure that they're correct and that the permissions on the files allow them to be executed by the user or computer. Run the scripts manually outside the GPO to ensure that they don't hang during processing. To control how long a GPO can run, you can set per-computer GPO timeouts in an Administrative Template policy (see Figure 6.1). The policy shown in Figure 6.1 is a cumulative limit on the time it takes to run all scripts that a user or computer is subject to. If you're unsure what per-user and per-computer scripts are running, check in the Registry under HKCU and HKLM in \Software\Policies\Microsoft\Windows\System\Scripts for a list of logon/logoff and startup/shutdown scripts that the computer has run.
A single GPO isn't processed at all, or only parts of it are processed.	Many things can cause a GPO to simply not be processed, including: missing or incorrect permissions on the Group Policy Container (GPC) or GPT; the GPO is out of sync (that is, the GPT and GPC don't have the same version number); the GPO is configured to process only if it's changed since the last processing cycle.	First check that the Read and Apply Group Policy permissions are applied on the GPO for the users and computers whom you intend to receive the policy. (For more info on using security to affect which GPOs are processed by a given user or computer, see chapters 3 and 4.) If the policy isn't being processed consistently or at all, verify that it's actually supposed to. For GPOs that might be out of sync, see the discussion in Chapter 2 on using the Replication Monitor (replmon.exe) utility for validating GPO synchronization. Policies such as Administrative Template, Software Installation, and Folder Redirection won't run if nothing in the GPO has changed since the last time it was run. To verify what version of the GPO ran most recently, look in a workstation's or server's Registry under HKLM or HKCU in: \Software\Microsoft\Windows\CurrentVersion\GroupPolicy\History (see Figure 6.2). There you'll see a number of Globally Unique ID (GUID)-based keys that correspond to the client-side extensions (CSEs) that are installed on the computer. Under each CSE is a numbered key that corresponds to a GPO processed by that key. The key stores information about the most recent version of a GPO that was processed. You can change whether a CSE processes a GPO, regardless of whether it's changed, by

This Problem	Has This Probable Cause	And These Potential Solutions
		using Administrative Template policy. Specifically, in Computer Configuration\Administrative Templates\System\Group Policy, each CSE has a policy item that lets you control its policy-processing behavior (see Figure 6.3). In these policy items, you can have the CSE always process a GPO regardless of whether it's changed since the last time. Of course, this increases the time it takes to process policies when a computer starts up or a user logs on, but it can ensure that a policy is always enforced.
A single GPO isn't processed at all, or only parts of it are processed.	The workstation may have detected a slow network link between itself and the domain controller that it's using as the source of the GPO. In this case, Folder Redirection and Software Installation policy are by default not processed.	Verify whether a workstation or server has detected a slow link between itself and a domain controller by viewing the userenv.log file. I describe this log file and how to use it to troubleshoot GPOs later in this chapter (see "Enabling GPO Logging").
A single GPO isn't processed at all, or only parts of it are processed.	If only parts of a GPO aren't being processed, there could be a problem with the CSE dynamic-link library (DLL) that is responsible for processing that policy function. (For more details on CSEs, see Chapter 2.)	If a GPO is failing because of a problem with a CSE, there are a few things you can do. First, verify that all of the CSEs that should be registered on the computer indeed are by comparing the keys under HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\GPExtensions on the bad computer with those on a known good one. Then verify that the DLLs listed under each Registry key are actually found in the file system where they're expected to be (for Microsoft-provided CSEs, this is usually in %systemroot%\system32).
The wrong GPOs are being processed.	If a computer or user is processing GPOs that you think they shouldn't be, a number of problems may be occurring. For example, a computer may think it's in the wrong site and thus receiving a site-linked GPO that you're not expecting.	You can verify which site a computer thinks it's in by viewing the Registry value HKLM\System\CurrentControlSet\Services\Netlogon\Parameters\DynamicSiteName or using the nltest.exe utility in Windows 2000 Support Tools using the following syntax: <pre>nltest /server:<machine name> /dsgetsite</pre> where <i><machine name></i> is the workstation or server whose site information you're trying to retrieve. If a workstation's or server's site is incorrect, use the Active Directory Sites and Services Microsoft Management Console (MMC) snap-in to verify that the subnet that the computer belongs to is associated with the correct site.
The wrong GPOs are being	A user may be getting user-based policy that doesn't correspond to the Organizational Unit (OU) that he or	Verify that a workstation or server is set to loopback processing by examining the Registry key HKLM\Software\Policies\

This Problem	Has This Probable Cause	And These Potential Solutions
processed	she resides in. In this case, the computer being logging in to may have loopback policy applied. (For a discussion of loopback, see Chapter 5.)	Microsoft\Windows\System. If the key contains the UserPolicyMode value and it's set to a value of 1 or 2, loopback processing is active on the computer.
The wrong GPOs are being processed	A GPO could be linked to a container that you don't expect.	Sometimes an administrator inadvertently links a GPO to a container that it wasn't intended to be linked to. If so, your users and computers in the Finance OU could receive a lockdown policy from your Help Desk OU. You can use the Group Policy MMC snap-in's Link Search feature to find all of the links to a particular GPO: In the Active Directory Users and Computers MMC snap-in, select a container object such as a domain or OU, right-click it, then choose Properties from the shortcut menu. Click the Group Policy tab and highlight the GPO that you want to search for links to. Click Properties, then the Links tab. Select the domain in which you want to search for links, then click Find Now. All links to the GPO are displayed (see Figure 6.4).
No GPO processing is occurring	Your users or computers may not be receiving any GPOs. There are a few possible reasons for this, including: The trust relationship between a computer and the domain is broken, so no GPOs can be processed. (In this case, the user can't log on to the domain either.) Domain Name System (DNS) name-resolution problems are preventing a client machine from resolving domain controller services required for GPO processing. Large-scale AD or NT File Replication Service (NTFRS) replication problems are causing GPO synchronization problems.	You can validate the trust between a computer account and its domain using the nltest.exe utility with the following syntax: <pre>nltest /server:<machine name> /sc_query:<domain></pre> Where <i><machine name></i> is the name of the workstation or server whose trust connection you're trying to validate and <i><domain></i> is the domain name you're trying to validate with. You can also use the netdom.exe utility using the form: <pre>netdom verify <machine name> /domain:<domain></pre> If DNS problems might be preventing proper resolution of domain controllers and thus GPOs, use netdiag.exe in the Windows 2000 Support Tools to troubleshoot DNS problems. (See "Using GPO Troubleshooting Tools" later in this chapter.) If the problem has to do with AD or NTFRS replication, use replmon.exe to view problems with these infrastructure services that might be affecting GPO processing.

Table 6.1: Common GPO problems, their causes, and solutions.

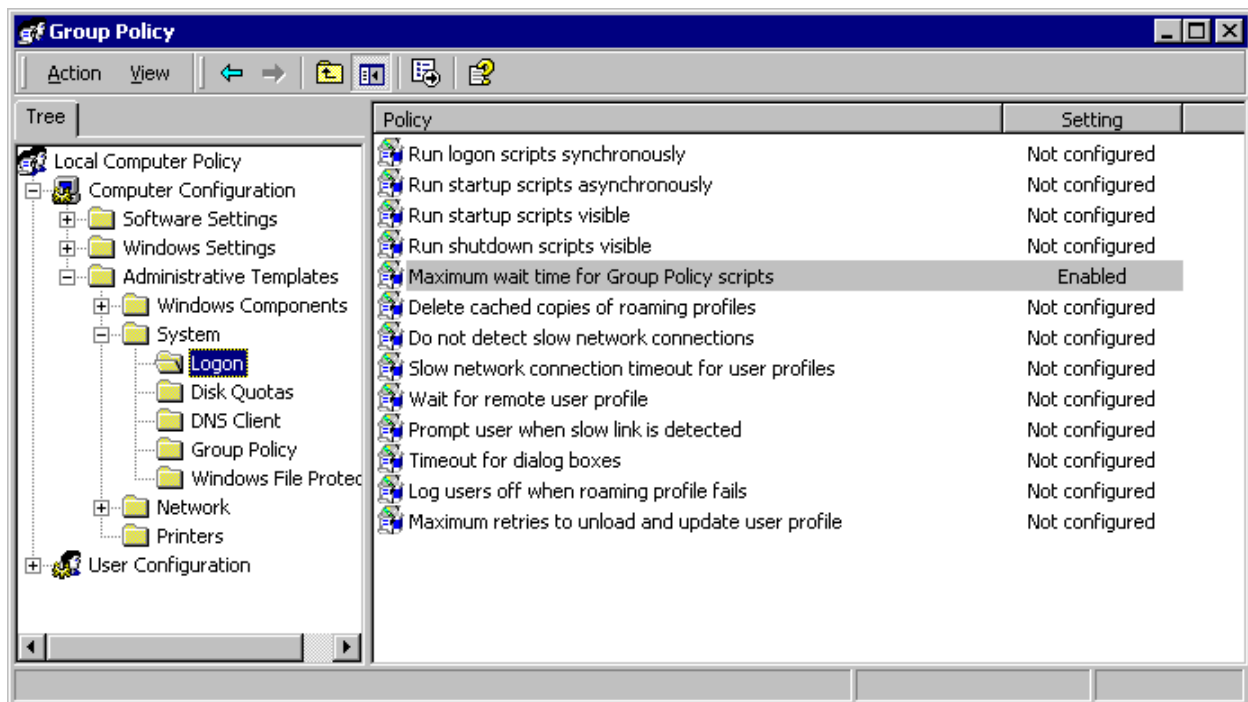


Figure 6.1: Enabling the Administrative Template policy to limit the amount of time that GPO-based scripts (logon, startup, and shutdown) will run.

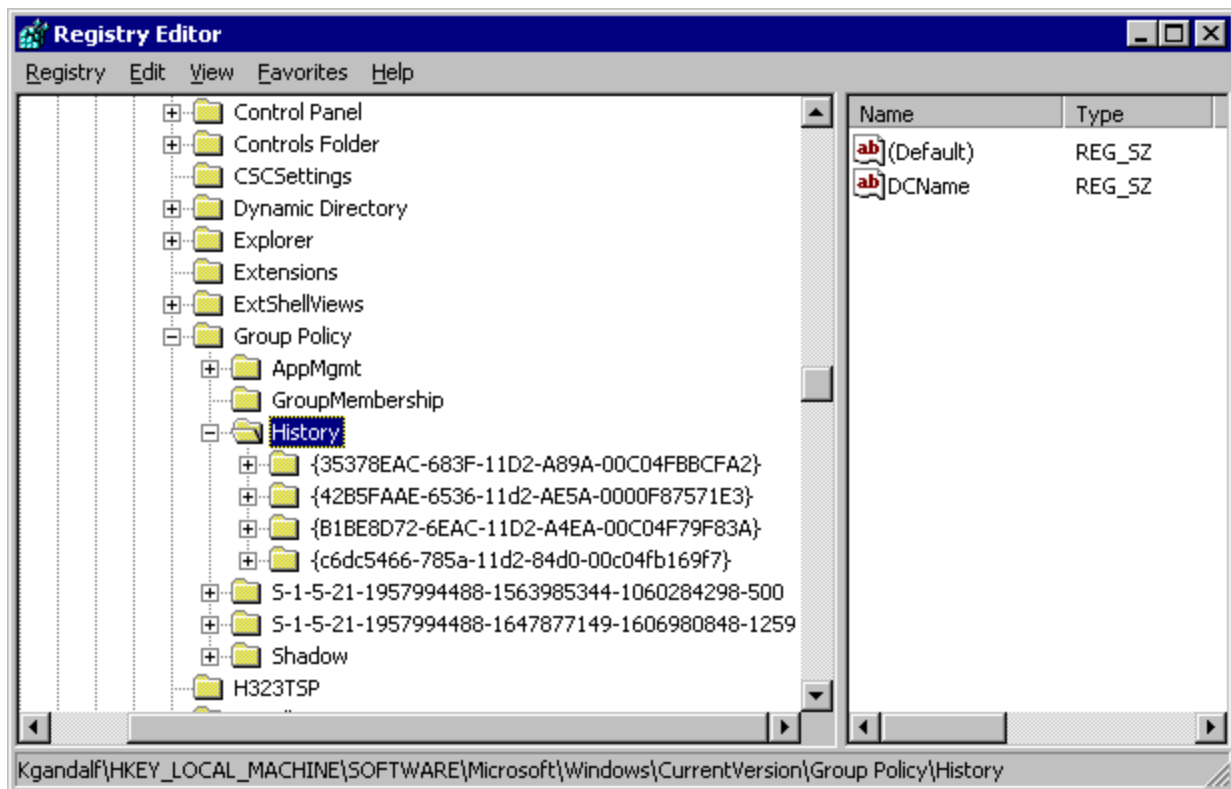


Figure 6.2: Using the History key in the Registry to verify what version of the GPO ran most recently.

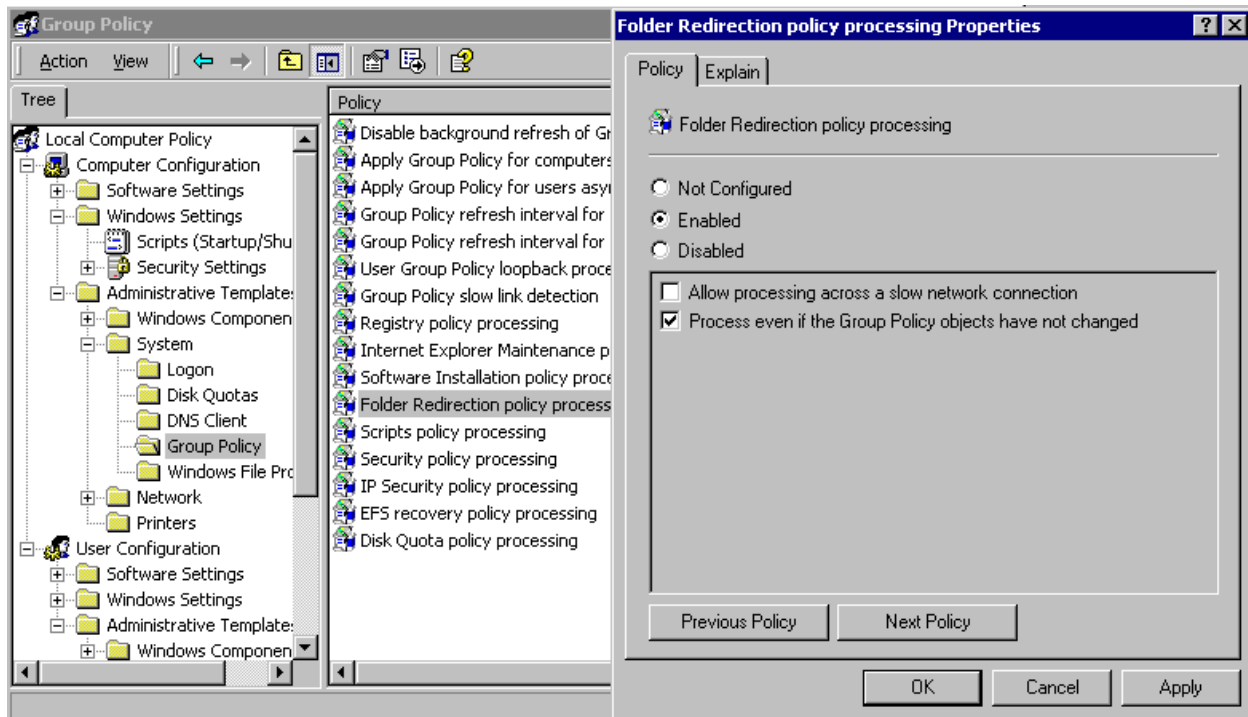


Figure 6.3: Using policy to ensure that a CSE always processes a GPO, regardless of whether it's changed since the last processing cycle.



Figure 6.4: Using the Group Policy MMC snap-in's Link Search feature to find all of the links to a GPO.

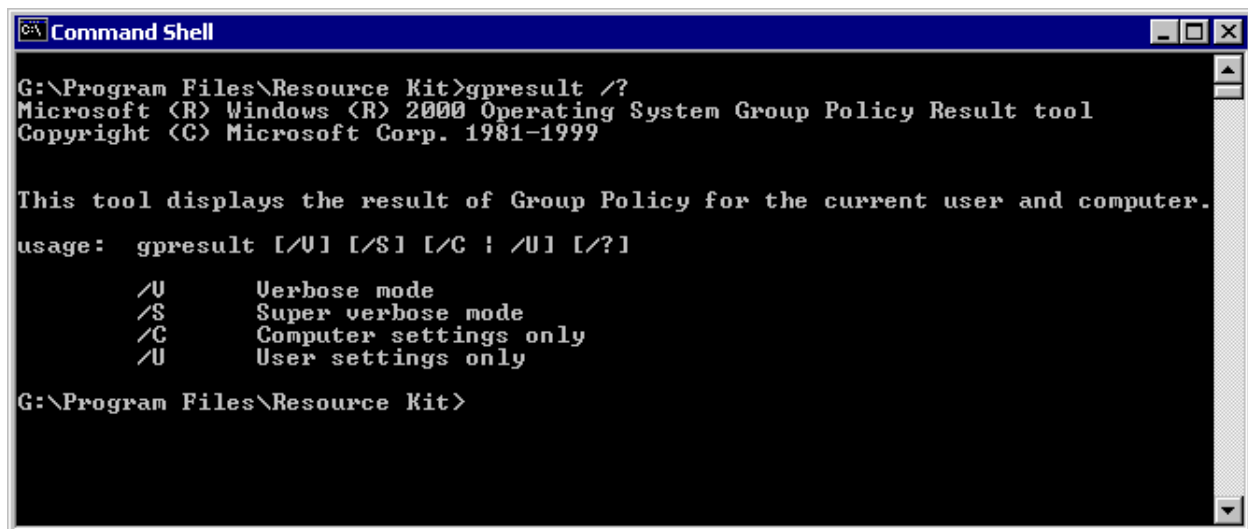
Using GPO Troubleshooting Tools

When there are problems with your GPO infrastructure, it's good to have a set of tools that you can use to help diagnose the problem. Fortunately, the Windows 2000 Resource Kit and Support Tools contain a few useful command-line and graphical user interface (GUI)-based utilities for dealing with GPO problems. In this section, I'll describe the most useful tools and how you can use them in your own environment. I'll also talk about the one third-party tool on the market as of this writing for dealing with GPOs—FAZAM 2000 from Full Armor.

☞ A Reduced Functionality Version of FAZAM 2000 is available for free from Microsoft at <http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/fazam2000-o.asp>.

Group Policy Result Tool (*gpresult.exe*)

The *Group Policy Result tool* (*gpresult.exe*) is a Windows 2000 Resource Kit command-line utility that reports on the current state of a workstation and user with respect to GPOs. It shows what GPOs have run for the current computer and user and, to some extent, what policy settings each GPO has provided. Figure 6.5 shows the command-line options available for *gpresult.exe*.



```
Command Shell
G:\Program Files\Resource Kit>gpresult /?
Microsoft (R) Windows (R) 2000 Operating System Group Policy Result tool
Copyright (C) Microsoft Corp. 1981-1999

This tool displays the result of Group Policy for the current user and computer.

usage:  gpresult [/U] [/S] [/C : /U] [/?]

        /U          Verbose mode
        /S          Super verbose mode
        /C          Computer settings only
        /U          User settings only

G:\Program Files\Resource Kit>
```

Figure 6.5: The command-line options available in gpresult.exe.

As you can see, gpresult.exe supports four options: Verbose mode (/V), Super Verbose mode (/S), Computer Settings Only (/C), and User Settings Only (/U). There are a few differences between Verbose mode and Super Verbose mode.

- The main difference is that in Super Verbose mode, Registry values that are set by Administrative Template policy, and use the REG_BINARY data type, are fully displayed; in Verbose mode, the contents of binary Registry values aren't displayed
- In Super Verbose mode, any applications made available using Software Installation policy are listed, whereas in Verbose mode, they're not
- In Verbose mode, each GPO version is listed as a single value. However, in Super Verbose mode, the version information is broken down by GPT and GPC so that you can see whether any synchronization problems exist between them.

One noticeable piece of information that gpresult.exe doesn't provide is information about any security settings that have been delivered using GPO. This is an unfortunate limitation that makes the tool less valuable. The tool *will* show that you received security policy from one or more GPOs, but it instructs you to use the security configuration and analysis MMC snap-in to view actual security policy. Unfortunately, the security configuration and analysis snap-in doesn't tell you which GPOs are delivering which security policy; it only tells you what effective policy is currently running on the computer. For this more detailed Resultant Set of Policy (RSOP) functionality, you have to turn to a tool like FAZAM 2000 from Full Armor or wait until Microsoft delivers it in the next version of Windows.

The last two modes in gpresult.exe are Computer Settings Only and User Settings Only. They let you display GPO policy per user or per computer, so that you can exclude one or the other from a report. (The default is to present both). This helps minimize the amount of data the tool returns. You can also use the /c or /u option with the /v or /s option to select the verbosity level.

In general, gpresult.exe is a good tool for giving you an overall sense of which GPOs are being processed by the current user and computer. You can't run the tool remotely against a different computer/user combination, but you can get some useful information out of it when you run it locally. Figure 6.6 shows some sample gpresult.exe output. As you can see, the first part of the report shows a lot of useful information about the computer and user running the report, including the site the computer is in (SanFrancisco), the user's distinguished name in AD (CN=Administrator,CN=Users,DC=mar-elia,DC=com), the groups they belong to, and the security rights they have on the domain.

```

superverbose.txt - Notepad
File Edit Format Help
Operating System Information:
Operating System Type:      Domain Controller
Operating System Version:   5.0.2195.Service Pack 1
Terminal Server Mode:      Remote Administration

#####

User Group Policy results for:
CN=Administrator,CN=Users,DC=mar-elia,DC=com

Domain Name:      MAR-ELIA
Domain Type:      Windows 2000
Site Name:        SanFrancisco

Roaming profile:   (None)
Local profile:    E:\Documents and Settings\Administrator

The user is a member of the following security groups:

MAR-ELIA\Domain Users
\Everyone
BUILTIN\Administrators
BUILTIN\Users
MAR-ELIA\Domain Admins
MAR-ELIA\Schema Admins
MAR-ELIA\Enterprise Admins
MAR-ELIA\Group Policy Creator Owners
\LOCAL
NT AUTHORITY\INTERACTIVE
NT AUTHORITY\Authenticated Users

The user has the following security privileges:

Bypass traverse checking
  
```

Figure 6.6: Using gpresult.exe to display details of the computer and user running a report.

The second part of the report (not shown here) lists the GPOs processed by the user and computer, and it's organized by CSE (for example, Administrative Template, Software Installation, Folder Redirection, and so on).

The gpresult.exe tool is quite useful for identifying when a particular Administrative Template policy is being applied because it lists individual Registry entries set by a given GPO. It also provides some summary information about each GPO, and this can be handy when you're troubleshooting problems. This information includes things like the GPO's friendly name, its GUID, version information, and what it's linked to. This information is shown in Listing 6.1.

```

The user received "Application Management" settings from these GPOs:
Domain Software Installation Policy (v1.0)
Revision Number:      13 (Active Directory) 13 (Sysvol)
  
```

Unique Name:	{7D48C869-4882-40D7-A2A7-97193664F282}
Domain Name:	mar-elia.com
Linked to:	Domain (DC=mar-elia,DC=com)

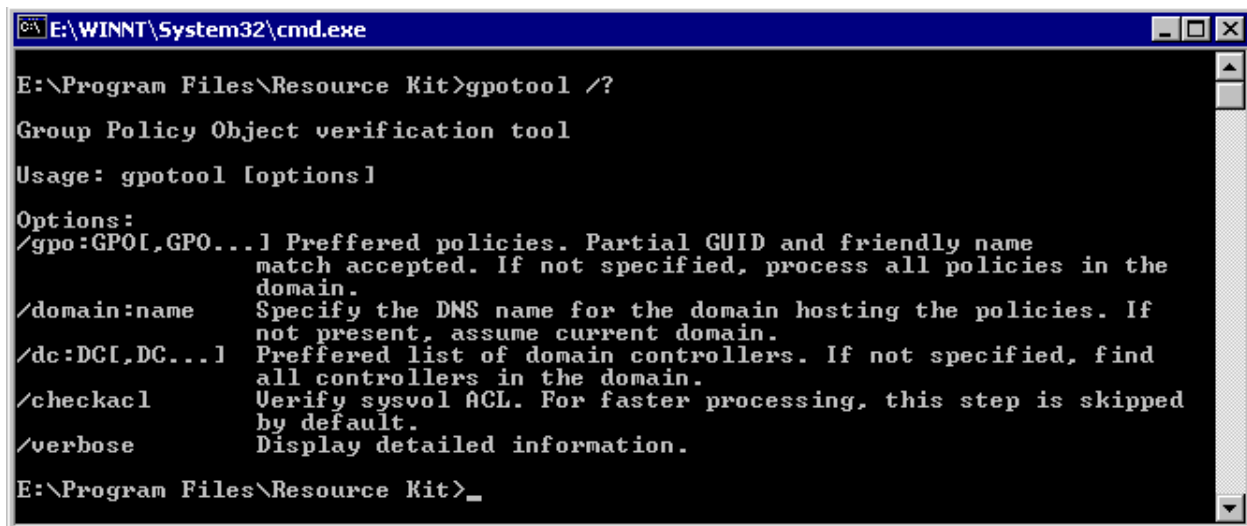
Listing 6.1: The output of gpresult.exe, displaying information about a GPO.

While gpresult.exe isn't a full-featured RSoP tool, it'll do in a pinch—such as when you need to find out what's happening on a workstation or server right now.

Group Policy Object Verification Tool (gpoutil.exe)

The *Group Policy Object Verification tool (gpoutil.exe)* in the Windows 2000 Resource Kit, is valuable for ensuring that your GPO infrastructure is functioning properly. Rather than focusing on the effects of GPOs on a single workstation or user, as gpresult.exe does, gpoutil.exe surveys all GPOs in a given domain and ascertains their health. It does this by querying each domain controller in a domain and verifying whether the GPT and GPC are in sync (that is, of the same version). If not, it reports errors.

gpoutil.exe also provides a bit more control over the scope of the search using a number of command-line options. These are shown in Figure 6.7 and listed below.



```
E:\WINNT\System32\cmd.exe

E:\Program Files\Resource Kit>gpoutil /?

Group Policy Object verification tool

Usage: gpoutil [options]

Options:
/gpo:GPO[,GPO...] Preferred policies. Partial GUID and friendly name
                    match accepted. If not specified, process all policies in the
                    domain.
/domain:name        Specify the DNS name for the domain hosting the policies. If
                    not present, assume current domain.
/dc:DC[,DC...]      Preferred list of domain controllers. If not specified, find
                    all controllers in the domain.
/checkacl           Verify sysvol ACL. For faster processing, this step is skipped
                    by default.
/verbose            Display detailed information.

E:\Program Files\Resource Kit>
```

Figure 6.7: The command-line options for gpoutil.exe.

- **/gpo**— Searches for a particular GPO, rather than all GPOs.
- **/domain:name**—Searches in a domain other than the one in which it's running—that is, you can perform cross-domain consistency checks on GPOs.
- **/dc**— Searches on a particular domain controller, or in a list of domain controllers, rather than in all of the domain controllers in a domain.

- **/checkacl**—Is supposed to validate whether the permissions on the GPT for a given GPO are consistent with those on the GPC. However, in tests that I ran using this option, after changing permissions on a GPT for a GPO, gpoutil.exe registered no problems whatsoever. So my guess is that this feature wasn't quite implemented!
- **/verbose**—Instead of merely reporting back that GPOs are OK, lists version and other information on all GPOs that you report on. Listing 6.2 shows an example of the verbose output on a GPO reported by gpoutil.exe.

Policy {BB5758D9-30FF-4BC5-A262-482321D9F928}
Policy OK Details:
DC: yquem.mar-elia.com
Friendly name: HQ Lockdown Policy Created: 3/11/2001 2:52:04 AM Changed: 5/1/2001 3:39:56 AM DS version: 3(user) 4(machine) Sysvol version: 3(user) 4(machine) Flags: 0 User extensions: [{3060E8D0-7020-11D2-842D-00C04FA372D4}{3060E8CE-7020-11D2-842D-00C04FA372D4}][{A2E30F80-D7DE-11D2-BBDE-00C04F86AE3B}{FC715823-C5FB-11D1-9EEF-00A0C90347FF}] Machine extensions: [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}{0F6B957D-509E-11D1-A7CC-0000F87571E3}] Functionality version: 2

Listing 6.2: Finding information about a GPO using the verbose option in gpoutil.exe.

In this listing, gpoutil.exe reports the following on the HQ Lockdown Policy GPO:

- That the policy is OK
- The details of the domain controller on which the policy was found
- The friendly name for the policy, when it was created and last changed, and its current GPC (*DS*) and GPT (*Sysvol*) versions
- The Flags field shows whether user or computer policy settings have been disabled on the GPO. In the listing above, a value of 0 indicates that both computer and user settings are enabled on this GPO.
- The User and Machine extensions fields show the GUIDs of the CSEs used in this GPO (or example, the GUID {A2E30F80-D7DE-11d2-BBDE-00C04F86AE3B} corresponds to the IE Maintenance CSE, which has been set in this GPO)

- The functionality version corresponds to the version of the Group Policy snap-in tool that was used to create this GPO.

Let's look at an example of how gpoutil.exe finds problems in an environment. Let's say you suspect that there is a problem with the HQ Lockdown Policy GPO on one of your domain controllers. Using the tool with the following syntax, you can zero in on the problem:

```
gpoutil /gpo:"HQ Lock" /dc:yquem
```

You use the /gpo option to search for this GPO. You don't need to enter the entire GPO friendly name; you can enter just the first few characters. (You can't enter another part of the name, such as *Lockdown*.) You also use the /dc option to specify a domain controller by the name of Yquem, which is where you want to focus your reporting. Figure 6.8 shows the output of this command.

```
Select E:\WINNT\System32\cmd.exe
E:\Program Files\Resource Kit>gpoutil /gpo:"HQ Lock" /dc:yquem
Validating DCs...
Available DCs:
yquem
Searching for policies...
Found 1 policies
=====
Policy {BB5758D9-30FF-4BC5-A262-482321D9F928}
Error: Version mismatch on yquem, DS=196612, sysvol=196613
Details:
=====
DC: yquem
Friendly name: HQ Lockdown Policy
Created: 3/11/2001 2:52:04 AM
Changed: 5/1/2001 5:11:56 AM
DS version: 3<user> 4<machine>
Sysvol version: 3<user> 5<machine>
Flags: 0
User extensions: [{3060E8D0-7020-11D2-842D-00C04FA372D4}<3060E8CE-7020-11D2-842D-00C04FA372D4>] [{A2E30F80-D7DE-11D2-BBDE-00C04F86AE3B}<FC715823-C5FB-11D1-9EEF-00A0C90347FF>]
Machine extensions: [{35378EAC-683F-11D2-A89A-00C04FBBCFA2}<0F6B957D-509E-11D1-A7CC-0000F87571E3>]
Functionality version: 2
=====

Errors found

E:\Program Files\Resource Kit>
```

Figure 6.8: Locating an error in a GPO using gpoutil.exe.

This figure shows that there is a problem on this GPO because the version on the GPT (196613) is different than the version on the GPC (196612). In this case, because the version on the GPT is greater, you might infer that AD replication problems on this domain controller are preventing the latest GPO changes from replicating to the GPC. A quick look in the AD event log on this server should confirm such a problem.

Network Connectivity Tester (netdiag.exe)

Another useful Windows 2000 Resource Kit utility is *Network Connectivity Tester (netdiag.exe)*. This tool has many capabilities, but it's basically designed to validate that all network-related aspects of your Win2K device are functioning correctly. Netdiag.exe can do everything from checking the status of your network cards to verifying that all of the correct DNS entries are

registered for a given domain controller. If you're having a problem like that described in Table 6.1 above, where none of the GPOs for a given computer or user in your domain are being processed, this is the tool to use.

Netdiag.exe detects problems with modems, network interface cards (NICs), infrared ports—you name it, this tool finds it. It even has a facility to fix certain basic problems (for example, DNS registrations that are missing). In Debug mode, the tool lists an amazing array of data, including routing information on each NIC installed, current Transmission Control Protocol (TCP) ports connected or listening on, detailed AD configuration information, and so on. Its range is really quite extensive. Listing 6.3 shows all of the tests that netdiag.exe can perform.

```
Ndis - Netcard queries Test
IpConfig - IP config Test
Member - Domain membership Test
NetBTTransports - NetBT transports Test
Autonet - Autonet address Test
IpLoopBk - IP loopback ping Test
DefGw - Default gateway Test
NbtNm - NetBT name Test
WINS - WINS service Test
Winsock - Winsock Test
DNS - DNS Test
Browser - Redir and Browser Test
DsGetDc - DC discovery Test
DcList - DC list Test
Trust - Trust relationship Test
Kerberos - Kerberos Test
Ldap - LDAP Test
Route - Routing table Test
Netstat - Netstat information Test
Bindings - Bindings Test
WAN - WAN configuration Test
Modem - Modem diagnostics Test
Netware - Netware Test
IPX - IPX Test
IPSec - IP Security Test
```

Listing 6.3: A list of tests available in netdiag.exe.

For the purposes of troubleshooting GPOs, the tests of interest are the DNS test and domain membership test. The DNS test verifies that a server, workstation, or domain controller is correctly registered with its DNS servers, while the domain membership test ensures that the membership between the device where the utility is being run and the domain is valid.

The syntax for performing just the DNS test is as follows:

```
netdiag /test:DNS
```

or for verbose output:

```
netdiag /v /test:DNS
```

The results of this test are shown in Figure 6.9.

```
E:\WINNT\System32\cmd.exe
Computer Name: YQUEM
DNS Host Name: yquem.mar-elia.com
System info : Windows 2000 Server (Build 2195)
Processor : x86 Family 6 Model 5 Stepping 0, GenuineIntel
List of installed hotfixes :
    Q147222

Netcard queries test . . . . . : Passed
GetStats failed for 'Infrared Modem Port'. [ERROR_NOT_SUPPORTED]
GetStats failed for 'Infrared Port'. [ERROR_NOT_SUPPORTED]

Per interface results:
    Adapter : Local Area Connection
        Netcard queries test . . . : Passed

Global results:

Domain membership test . . . . . : Passed

NetBT transports test . . . . . : Passed
List of NetBt transports currently configured:
    NetBT_Tcpip_{2928AE87-DCBF-4A82-8C07-30280DF1C1B5}
    1 NetBt transport currently configured.

DNS test . . . . . : Failed
[WARNING] The DNS entries for this DC are not registered correctly on DNS se
rver '192.168.1.151'. Please wait for 30 minutes for DNS server replication.
[_FATAL] No DNS servers have the DNS records for this DC registered.
```

Figure 6.9: Running the DNS test using netdiag.exe.

I ran this utility from a domain controller in my domain. Note that netdiag.exe runs a number of tests by default without explicitly specifying them (e.g. the domain membership and NetBT transports test). As you can see at the end of the output, the DNS test failed with errors indicating missing entries on the DNS server that I'm pointing at. At this time, I can wait to see if the missing entries replicate from another DNS server, or I can use netdiag.exe's /fix option to try and fix the problem.

The /fix option compares the records registered by the domain controller in DNS with the ones that should be registered; they're well known for a given domain controller in a domain. If it finds one or more records missing, it registers them for the server. The syntax for this command is:

```
netdiag /fix
```

Listing 6.4 shows the results reported by netdiag.exe after I ran this command on my domain controller with the DNS problem.

```
DNS test . . . . . : Failed
[FIX] re-register DC DNS entry '_ldap._tcp.SanFrancisco._sites.mar-
elia.com.'
' on DNS server '192.168.1.151' succeed.
```

```
FIX PASS - netdiag re-registered missing DNS entries for this DC
successful
y on DNS server '192.168.1.151'.
[FATAL] No DNS servers have the DNS records for this DC registered.
```

Listing 6.4: The output of the netdiag.exe /fix option.

This listing shows that netdiag.exe was able to fix the missing DNS entry, which was in the _tcp.SanFrancisco._sites.mar-elia.com zone.

Software Installation Diagnostics (addiag.exe)

Software Installation Diagnostics (addiag.exe) is another Windows 2000 Resource Kit utility that provides a wide range of troubleshooting functionality related to GPOs and, more specifically, to Software Installation in GPOs. This utility is a veritable treasure trove of functionality. It provides not only detailed information on applications that have been installed using GPO-based Software Installation but also general GPO information, such as GPO history (GPO version numbers registered during the last policy-processing cycle).

This tool also lets you enable some Software Installation–related logging using command-line options. Normally, you must enable this logging using Administrative Template policy or Registry hacks on a given computer. Let's take a look at how you can use addiag.exe to troubleshoot GPO-based Software Installation issues. Let's suppose you want to find out what applications have been installed using Software Installation on a particular workstation. The following syntax will provide the answers you're looking for:

```
addiag /test:ServerApps
```

This form of the command will dump a list of applications installed for both the computer and the user using GPO-based Software Installation. Figure 6.10 shows an example of the output for this command.

```
E:\WINNT\System32\cmd.exe
User dump for mar-elia.com
Dumping GPO list (3 items)...
GPO GUID: {31B2F340-016D-11D2-945F-00C04FB984F9}
Name: Default Domain Policy
Null
Object GUID: {00000000-0000-0000-0000-000000000000}
Package Flags:
Type: Legacy App
Null
Object GUID: {00000000-0000-0000-0000-000000000000}
Package Flags:
Type: Legacy App
GPO GUID: {7D48C869-4882-40D7-A2A7-97193664F282}
Name: Domain Software Installation Policy (v1.0)
Microsoft Office 2000 SR-1 Premium
Object GUID: {818ECE2E-9A45-40FB-ACB6-613E890327B8}
Package Flags:
Published
PostBeta3
UserInstall
OnDemandInstall
OrphanOnPolicyRemoval
ProductCode: {00000409-78E1-11D2-B60F-006097C998E7}
UI Level: Basic
Windows 2000 Administration Tools
Object GUID: {B83C3E15-C8A7-4AAD-98F0-3584721BBC49}
Package Flags:
PostBeta3
UserInstall
OnDemandInstall
Assigned
OrphanOnPolicyRemoval
ProductCode: {B7298620-EAC6-11D1-8F87-0060082EA63E}
UI Level: Basic
GPO GUID: {CA5C6B5B-DD48-4047-8FF3-99E3E8DA9348}
Name: Admin GPO
No Apps were found.
```

Figure 6.10: Checking for GPO-deployed applications using adddiag.exe.

Figure 6.10 shows the following information:

- The GPO's GUID, friendly name, which is Domain Software Installation Policy (v1.0), and the applications it deploys—Microsoft Office 2000 SR-1 Premium and the Windows 2000 Administration Tools.
- Each installed application, along with its object GUID, its unique product code, and the options, or package flags, which the administrator chose when the application was deployed. For example, in the figure above the Windows 2000 Administration Tools application was assigned on a per-user basis. In addition, I enabled the option to have the application installed by a file-extension association (OnDemandInstall in the figure) was also checked.
- The OrphanOnPolicyRemoval flag means that the administrator chose not to have the application uninstalled when the GPO carrying that application no longer applies to this user.

Another feature of adddiag.exe that I mentioned above is its ability to enable various logging features related to Software Installation on a Win2K device. Using the /trace option, adddiag.exe

lets you enable four types of logging. Table 6.2 lists these options and describes what they do. I'll discuss logging further in "Enabling GPO Logging" later in this chapter.

This Trace Option	Performs This Kind of Logging
AppmgmtOn	Enables a log file in %systemroot%\debug\usermode called appmgmt.log that performs extensive logging during a GPO-based Software Installation operation.
CstoreOn	Provides additional verbose logging for the AppmgmtOn option.
MSIOn	Turns on Windows Installer logging to log the actual installation steps of a Windows Installer application. Enabling this option creates log files in either %systemroot%\temp or %temp% with names like msi*.log for each application that is installed per computer or per user, respectively. Note that this is the same logging that is enabled using the Administrative Template policy item called Logging under Computer Configuration\Administrative Templates\Windows Components\Windows Installer.
UserEnvOn	Enables verbose logging of GPO and user-profile processing to the %systemroot%\debug\usermode\userenv.log file.

Table 6.2: The trace logging options that are available in addiag.exe.

Addiag.exe is a useful tool for providing information on your GPO-based software installations. But sometimes you just need to know what policies are being applied to a particular computer or user. This is where a tool like FAZAM 2000 comes in handy.

Full Armor Zero Administration for Windows 2000 (FAZAM 2000)

Full Armor Zero Administration for Windows 2000 (FAZAM 2000) from Full Armor Corporation (www.fullarmor.com) performs several GPO-related functions, including letting you calculate RSoP and manage all of your domain GPOs from a single interface. I'll talk more about these features in "Managing GPOs" later in this chapter. In this section, I'll focus on this tool's troubleshooting capabilities.

Two main features provide diagnostics. The first is a simple function in the Auditing and Diagnostics module called Client Side Auditing. This tool, shown in Figure 6.11, provides a filtered view of the application event log, showing only events related to GPO processing. This makes it easier to scan the event log for interesting events.

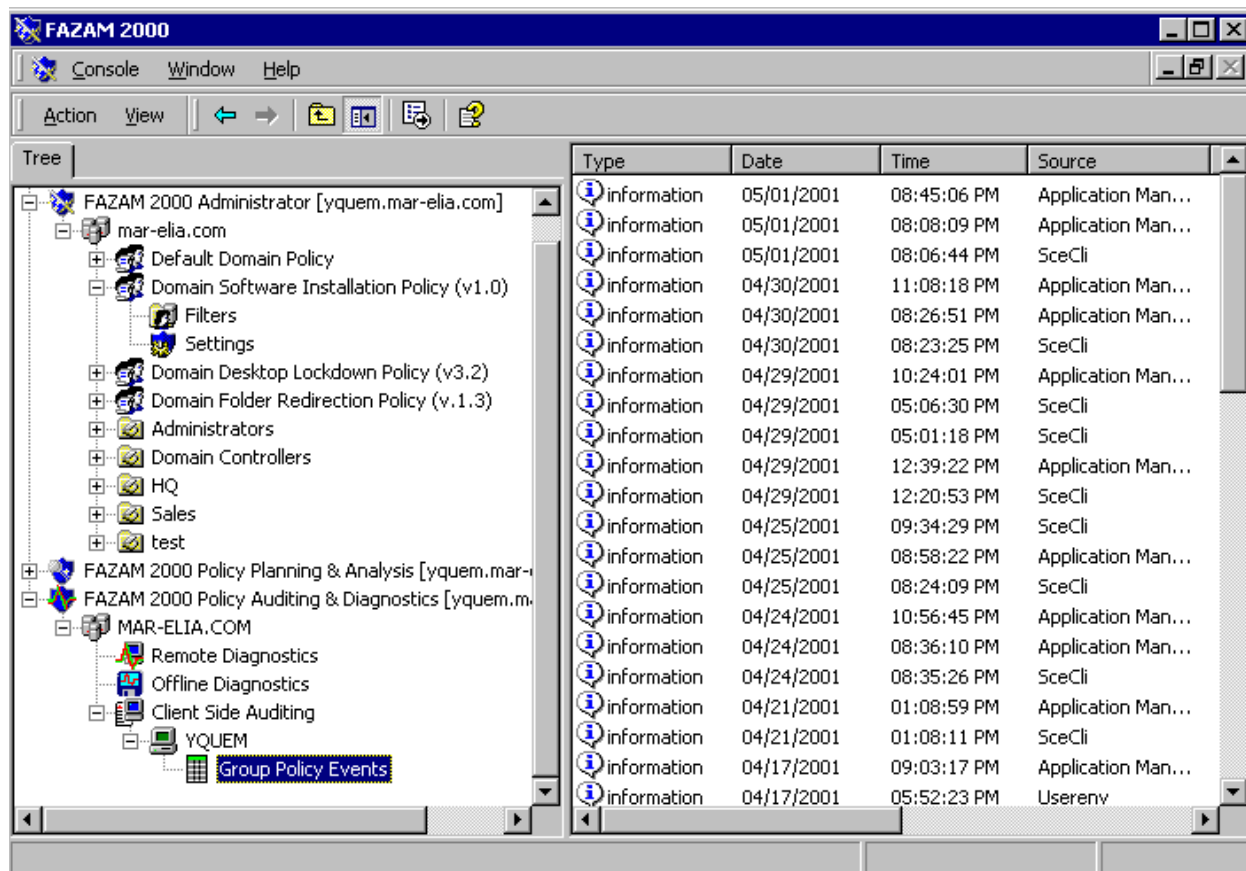


Figure 6.11: Using the Client Side Auditing feature in FAZAM 2000 to show events related to GPO processing.

The other feature in FAZAM 2000 is its ability to perform remote diagnostics. It's like a graphical version of gresult.exe, but it can also provide information on remote computers and users. Specifically, you connect to a computer of interest using the remote diagnostics feature, then FAZAM 2000 displays all of the GPOs and their current policy settings for the currently logged-on user and selected computer. This feature is handy for pointing out when you might have conflicting policy settings among several GPOs being applied to a given user or computer.

Enabling GPO Logging

When you troubleshoot problems with a GPO infrastructure, there is fortunately no shortage of logging available. Unfortunately, it has some disadvantages: It's scattered amongst several different log files on a given Win2K device; in some cases, it must be enabled manually on each device; and it requires the interested parties to collect each set of logs from each individual device they're interested in troubleshooting. Nevertheless, when you need the information, it's good to know how to get to it. This section will describe where and how GPO logging occurs and how you can enable it to your benefit.

The following list describes the main types of logs available for this purpose:

- **Application event logs**—Perform high-level logging of GPO processing per workstation or server
- **%systemroot%\debug\usermode log files**—Text files that display very in-depth information about GPO and user-profile processing
- **Windows Installer log files**—For Software Installation specifically, log application-installation details for applications deployed using Group Policy.

The Application Event Log

Of the bullet points above, the *application event log* is the first point of entry into any GPO-troubleshooting operation. By default, extended logging isn't enabled, so you have to enable it explicitly on every workstation or server on which you want to capture events. To enable event logging for GPO and related functions, you need to edit the Registry to add a series of keys and values. All of these logging entries reside in the Diagnostics key. It isn't present by default on a Win2K computer, so you must create it manually under the following Registry path:

HKLM\Software\Microsoft\Windows NT\CurrentVersion

Once you create the Diagnostics key under CurrentVersion, you need to add values particular to the type of logging you want to carry out. Table 6.3 describes the required values and what kinds of logging they perform.

This Registry Value (Name, Type, Value)	Logs This
RunDiagnosticLoggingGlobal REG_DWORD = 0x1	Logs all Group Policy–related processing, including Folder Redirection, Remote Installation Services (RIS) policy, Software Installation policy, and so on in the application event log in verbose mode.
RunDiagnosticLoggingGroupPolicy REG_DWORD= 0x1	Logs only high-level GPO processing step by step.
RunDiagnosticLoggingIntellimirror REG_DWORD=0x1	Logs only events related to RIS policy.
RunDiagnosticLoggingAppDeploy REG_DWORD= 0x1	Logs only events related to GPO-based Software Installation.

Table 6.3: The Registry values required to enable event-log auditing of GPOs.

Once you've enabled event logging, the application event log generates a new log entry for each step that a computer and user go through as they process their applicable GPOs.

 You can create a custom .adm template file to enable GPO application event logging on all computers in your Win2K infrastructure. The following .adm template snippet will add the RunDiagnosticLoggingGlobal flag to computers that process the GPO that contains it:

CLASS MACHINE

CATEGORY !!Custom

POLICY !!GPOLogging

```
KEYNAME "Software\Microsoft\Windows NT\CurrentVersion\Diagnostics"

EXPLAIN !!GPOLogging_Help

VALUENAME "RunDiagnosticLoggingGlobal"

VALUEON NUMERIC 1

VALUEOFF NUMERIC 0

END POLICY

END CATEGORY ; Custom
```

[strings]

GPOLogging="Enable Verbose GPO Logging"

GPOLogging_Help="By enabling this policy, you are turning on verbose logging of all GPO events in the application event log"

Custom="Custom Preference"

You can actually scroll through each event log to follow which GPOs are processed and whether any high-level errors are generated during a particular step. Figure 6.12 shows an example of one such event-log entry after enabling verbose global logging.

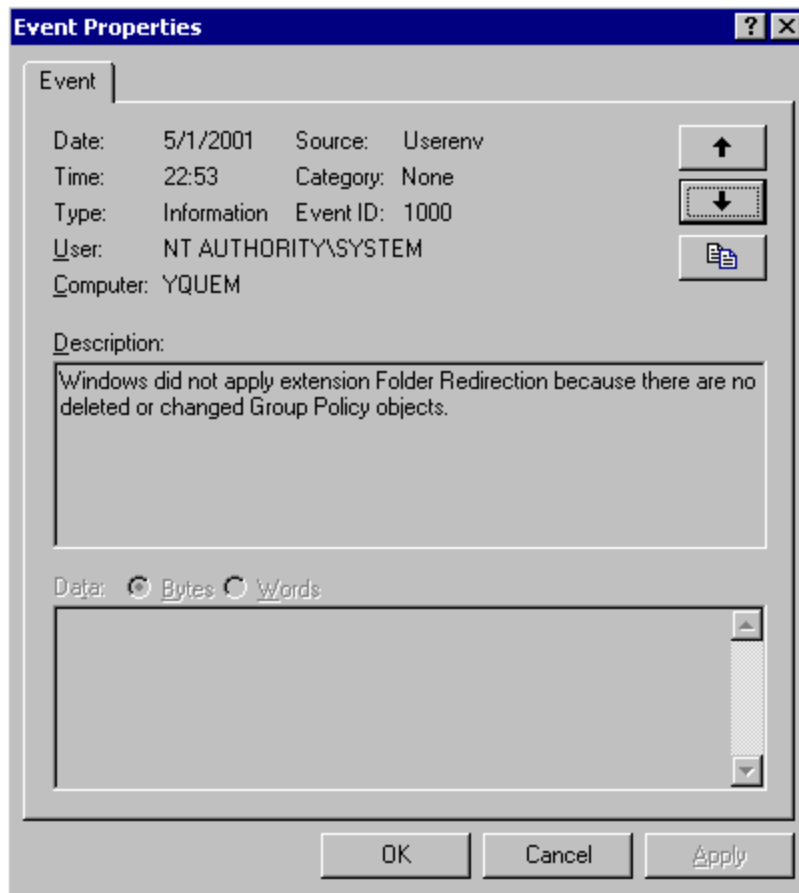


Figure 6.12: Viewing a log entry after enabling GPO logging.

GPO-related events typically appear in the application event log from one of three sources. Each source represents a different set of policy processing.

- **Userenv**—The event source responsible for enumerating the applicable GPOs and figuring out which ones apply or don't, as shown in Figure 6.12 above
- **Application Management**—Shows software-installation events
- **Scecli**—Represents security policy-processing events.

In some cases, a log event might return some fairly obscure error messages, which you may find no help at all. Listing 6.5 shows the text returned by one such event.

```
The Group Policy client-side extension Security was passed flags (17)
and returned a failure status code of (5).
```

Listing 6.5: An error message returned by the application event log during GPO processing.

How do you know what these flags and status codes mean? The flags refer to error messages that are part of the Win2K DLL responsible for handling much of the processing of GPOs and user

profiles, userenv.dll. The actual error codes can be found in the header file for this DLL—userenv.h in the Microsoft Platform Software Developer’s Kit (SDK).

Table 6.4 shows the values and a description of the flags related to GPO. These flags are given in hexadecimal (hex) values, but the event log shows decimal values, so you need to convert them.

This Flag Value in Hexadecimal (and Decimal)	Means This
0x00000001 (1)	The policy being applied is a computer-based (not user) policy.
0x00000010 (16)	This is a background refresh of policy. (That is, it’s not happening as a function of a user logging on or a computer starting up.)
0x00000020 (32)	The policy is currently being applied across a slow link.
0x00000040 (64)	The policy is set to output status in verbose mode to the event log.
0x00000080 (128)	No changes were detected for this GPO from the last processing cycle.
0x00000100 (256)	The network-link speed has changed between the last time the policy was processed and the current processing cycle.

Table 6.4: The flags that are shown in GPO-specific event-log entries.

The values shown in the event log, like the ones in Listing 6.5, are displayed in decimal notation and are calculated using a bitwise operation on the relevant flags shown in Table 6.4. For example, Listing 6.5 shows a flag value of 17. You do a bitwise OR operation on the 0x1 and 0x10 flags shown in the table to obtain the hex value, which is 11. Thus, the flag value of 17 indicates that the policy being reported on is a computer policy and that it’s a background refresh event, as opposed to a policy that is processed when a user logs on or a computer starts up.

The failure status codes shown in Listing 6.5 are simply Win32 error codes. You can figure out what those errors are by looking in the winerr.h header file in the platform SDK or simply typing the following at a command prompt:

```
Net helpmsg <#>
```

where <#> is the decimal error code. In Listing 6.5 above, Win32 error code 5 is “Access Denied”.

Usermode Logs

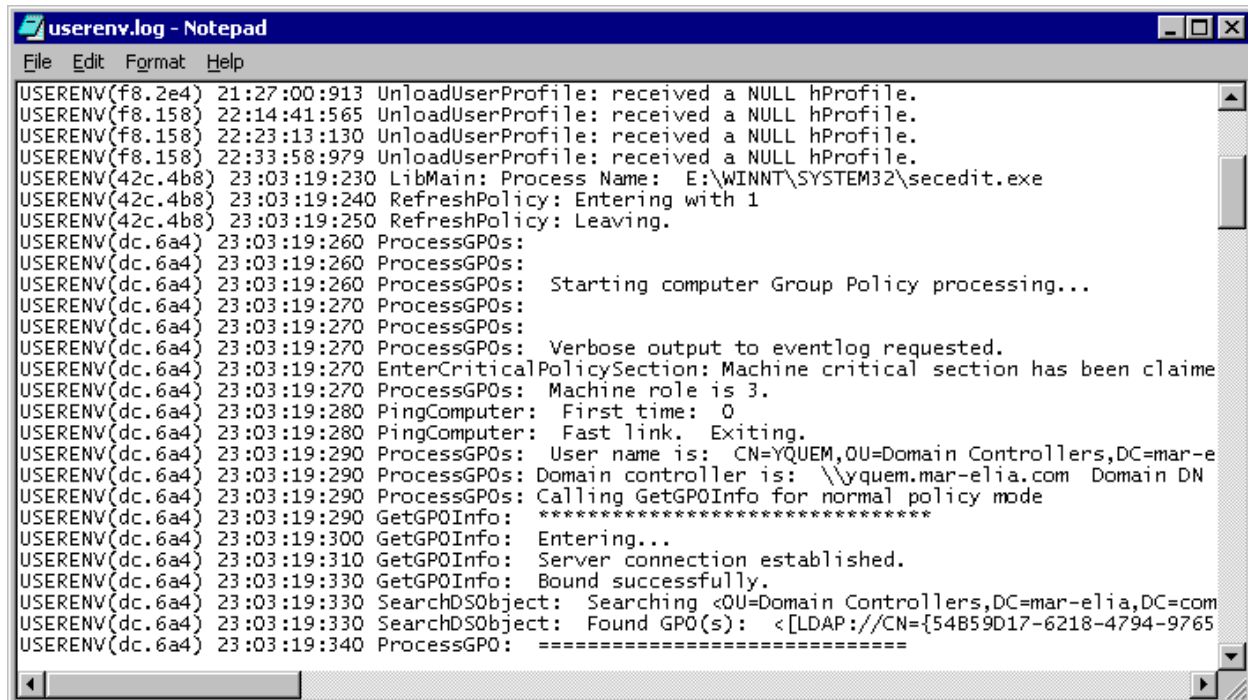
Usermode logs are a set of log files that you can activate in the %systemroot%\debug\usermode folder on a Win2K system. Depending on what you’ve activated, this folder contains several log files of interest. These logs provide very detailed logging of what is actually happening during GPO processing. If you don’t obtain the information that you need from the application event log, these usermode logs are your next point of investigation.

The userenv.log File

The main log file that contains detailed trace information on GPO and user-profile processing is *userenv.log*. This file is created by default when you install Win2K, but to enable verbose logging of policies and profiles, you need to set the following Registry value:

```
HKLM\Software\Microsoft\Windows
NT\CurrentVersion\Winlogon\Userenvdebuglevel= REG_DWORD 0x10002
```

Scanning through the userenv.log file to troubleshoot GPO problems can be a tedious process. This log file is very low-level, so to find what you're looking for, you need to exercise patience. Figure 6.13 shows an example of the output from this log, with verbose logging enabled.



```
userenv.log - Notepad
File Edit Format Help
USERENV(f8.2e4) 21:27:00:913 UnloadUserProfile: received a NULL hProfile.
USERENV(f8.158) 22:14:41:565 UnloadUserProfile: received a NULL hProfile.
USERENV(f8.158) 22:23:13:130 UnloadUserProfile: received a NULL hProfile.
USERENV(f8.158) 22:33:58:979 UnloadUserProfile: received a NULL hProfile.
USERENV(42c.4b8) 23:03:19:230 LibMain: Process Name: E:\WINNT\SYSTEM32\secedit.exe
USERENV(42c.4b8) 23:03:19:240 RefreshPolicy: Entering with 1
USERENV(42c.4b8) 23:03:19:250 RefreshPolicy: Leaving.
USERENV(dc.6a4) 23:03:19:260 ProcessGPOs:
USERENV(dc.6a4) 23:03:19:260 ProcessGPOs:
USERENV(dc.6a4) 23:03:19:260 ProcessGPOs: Starting computer Group Policy processing...
USERENV(dc.6a4) 23:03:19:270 ProcessGPOs:
USERENV(dc.6a4) 23:03:19:270 ProcessGPOs:
USERENV(dc.6a4) 23:03:19:270 ProcessGPOs: Verbose output to eventlog requested.
USERENV(dc.6a4) 23:03:19:270 EnterCriticalPolicySection: Machine critical section has been claime
USERENV(dc.6a4) 23:03:19:270 ProcessGPOs: Machine role is 3.
USERENV(dc.6a4) 23:03:19:280 PingComputer: First time: 0
USERENV(dc.6a4) 23:03:19:280 PingComputer: Fast link. Exiting.
USERENV(dc.6a4) 23:03:19:290 ProcessGPOs: User name is: CN=YQUEM,OU=Domain Controllers,DC=mar-e
USERENV(dc.6a4) 23:03:19:290 ProcessGPOs: Domain controller is: \yquem.mar-elia.com Domain DN
USERENV(dc.6a4) 23:03:19:290 ProcessGPOs: Calling GetGPOInfo for normal policy mode
USERENV(dc.6a4) 23:03:19:290 GetGPOInfo: *****
USERENV(dc.6a4) 23:03:19:300 GetGPOInfo: Entering...
USERENV(dc.6a4) 23:03:19:310 GetGPOInfo: Server connection established.
USERENV(dc.6a4) 23:03:19:330 GetGPOInfo: Bound successfully.
USERENV(dc.6a4) 23:03:19:330 SearchDSObject: Searching <OU=Domain Controllers,DC=mar-elia,DC=com
USERENV(dc.6a4) 23:03:19:330 SearchDSObject: Found GPO(s): <[LDAP://CN={54B59D17-6218-4794-9765
USERENV(dc.6a4) 23:03:19:340 ProcessGPO: =====
```

Figure 6.13: The output of the userenv.log file with verbose logging enabled.

You'll immediately notice a couple of things about the userenv.log file. First, no event dates are shown, only times. This means that you have to guess that the most recent events occurred on the current day. Next, the log file is filled, as you'd expect, from top to bottom, so newer events are at the end, not the beginning, of the file.

The figure above shows some useful information. First, you see that GPO processing starts at 23:03:19:260. At 23:03:19:280, you see an operation called PingComputer. This is where the workstation processing the GPO tests its network connection to the domain controller that is providing that GPO. If the workstation detects a fast link, as is shown here, normal policy processing occurs. If it detects a slow link, this is where you can discover it, and then see which CSEs are being skipped as a result.

Figure 6.14 shows the output a little further on in this log. It shows how each GPO is evaluated to determine whether anything has changed since the last processing and, if something has changed, which CSE has to process which GPO.

```

userenv.log - Notepad
File Edit Format Help
23:03:19:550 ProcessGPO: =====
23:03:19:550 ProcessGPO: Searching <CN={6AC1786C-016F-11D2-945F-00C04fB984F9},CN=Policies,CN=Sy
23:03:19:560 ProcessGPO: Machine has access to this GPO.
23:03:19:560 ProcessGPO: Found functionality version of: 2
23:03:19:560 ProcessGPO: Found file system path of: <\\mar-elia.com\sysvol\mar-elia.com\Polici
23:03:19:560 ProcessGPO: Found common name of: <{6AC1786C-016F-11D2-945F-00C04fB984F9}>
23:03:19:570 ProcessGPO: Found display name of: <Default Domain Controllers Policy>
23:03:19:570 ProcessGPO: Found machine version of: GPC is 4, GPT is 4
23:03:19:570 ProcessGPO: Found flags of: 0
23:03:19:570 ProcessGPO: Found extensions: [{827D319E-6EAC-11D2-A4EA-00C04F79F83A}{803E14A0-B4
23:03:19:570 ProcessGPO: =====
23:03:19:580 GetGPOInfo: Leaving with 1
23:03:19:580 GetGPOInfo: *****
23:03:19:580 DebugPrintGPOList: List of GPO(s) to process: "Local Group Policy" "Default Domain
23:03:19:590 ProcessGPOs: OpenThreadToken failed with error 1008, assuming thread is not impersonating
23:03:19:590 ProcessGPOs: -----
23:03:19:590 ProcessGPOs: Processing extension Registry
23:03:19:590 DebugPrintGPOList: List of GPO(s) to process: "Local Group Policy" "Default Domain
23:03:19:590 CompareGPOLists: The lists are the same.
23:03:19:590 CheckGPOs: No GPO changes and no security group membership change and extension Registry
23:03:19:600 ProcessGPOs: -----
23:03:19:600 ProcessGPOs: -----
23:03:19:600 ProcessGPOs: Processing extension Folder Redirection
23:03:19:600 ProcessGPOs: Extension Folder Redirection skipped with flags 0x90007.
23:03:19:600 ProcessGPOs: -----
23:03:19:600 ProcessGPOs: Processing extension Microsoft Disk Quota
23:03:19:600 ProcessGPOs: Extension Microsoft Disk Quota skipped with flags 0x90007.
23:03:19:600 ProcessGPOs: -----
23:03:19:600 ProcessGPOs: Processing extension Scripts

```

Figure 6.14: The userenv.log file also shows how GPOs are processed.

At the top of the figure above, you see that a GPO has been found, that the computer processing it has access to it, and that its friendly name (display name) is Default Domain Controllers Policy. After all GPOs are found that are applicable to this computer and user, each CSE then figures out which GPOs it needs to process and does so in turn. You see this towards the bottom of the figure, where the following are processed sequentially: the Registry extension (that is, Administrative Templates), Folder Redirection, disk quota, then scripts.

Other Log Files

You can enable other logs in the usermode folder to provide additional troubleshooting. These additional logs, and the Registry values required to enable them, are shown in Table 6.5 below.

This Log Function	Needs This Registry Value to Enable This Log
Enable logging during GPO editing (client-side errors only) —produces gpedit.log	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\GPEditDebugLevel = REG_DWORD 0x10002
Enable logging of the loading of .adm template files from the GPT —produces gptext.log	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\GPTextDebugLevel = REG_DWORD 0x10002
Enable logging of Folder Redirection processing —produces fdeploy.log	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics\FdeployDebugLevel= REG_DWORD 0x0f
Enable logging of Software Installation processing—produces	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics\AppmgmtDebugLevel=

appmgmt.log	REG_DWORD 0x9b
-------------	----------------

Table 6.5: Enabling usermode logging of various GPO-based policy.

Windows Installer Logging

The last area of logging that I'll mention are the logs related to Windows Installer-based application installations. If you plan to use the Software Installation feature in Group Policy to deploy applications that have been packaged using the Windows Installer (.msi) technology, you'll want to enable some amount of logging on your Win2K systems.

Unlike the application event logs or usermode logs, however, which you need to enable using manual Registry hacks, you can enable Windows Installer logging using an existing Administrative Template policy. Specifically, in Computer Configuration\Administrative Templates\Windows Components\Windows Installer, a policy item called Logging lets you enable verbose logging of .msi-based application installations. Figure 6.15 shows an example of enabling this policy.

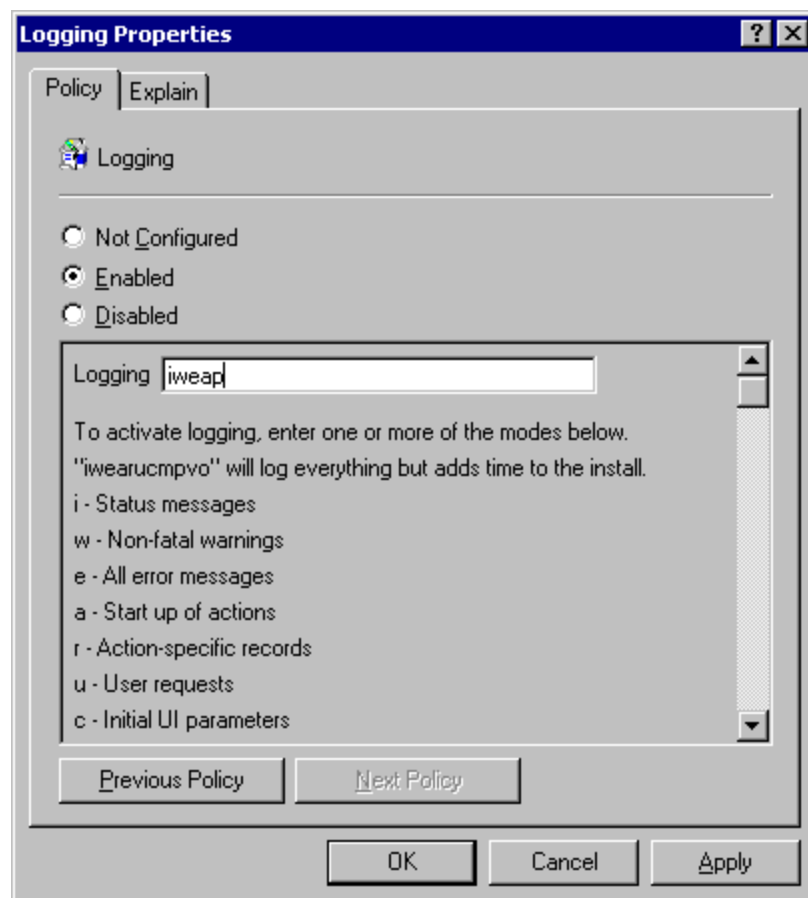


Figure 6.15: Enabling verbose Software Installation logging using Administrative Template policy.

Depending on how much verbosity you need, you can choose a number of different levels of logging. After enabling logging on a Win2K system, application-installation logs are stored in one of two places:

- **%systemroot%\temp**—When a GPO-deployed .msi-packaged application is installed from the computer's security context (for example, if the application was a computer-based assignment)
- **%temp% (the user's temporary folder)**—When the installation is triggered by a user.

In “Software Installation Diagnostics (addiag.exe)” earlier in this chapter, I mentioned that you can also enable Software Installation logging using addiag.exe. It does the same thing as Administrative Template policy, shown in Figure 6.15. However, addiag.exe doesn't give you a choice of the logging level; it enables all logging. Each application installation generates its own unique log file, which has a name using the form *msi*.log*, where * is some unique set of hexadecimal characters.

As Figure 6.16 shows, the contents of a Software Installation log file are fairly esoteric. Unless you have some in-depth knowledge of how the Windows Installer technology works, your best bet is to log only errors. That will give you a better chance of discovering problems without having to wade through a lot of unrelated messages.

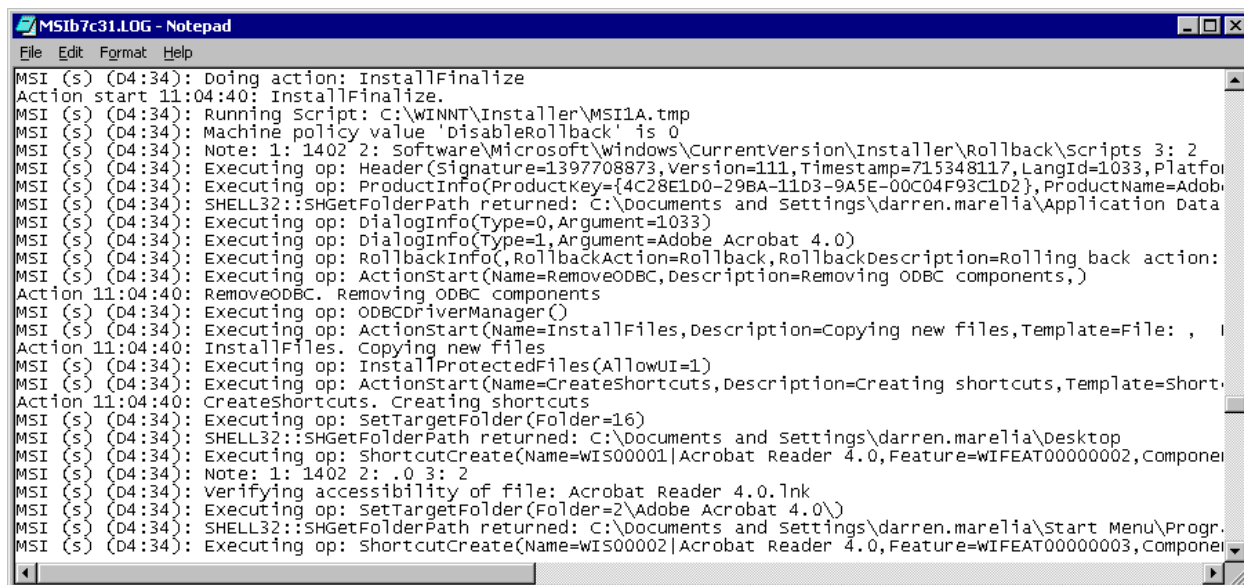


Figure 6.16: The output of a Windows Installer log file.

Managing GPOs

Now that I've discussed all the ways that you can “instrument” your GPO infrastructure to discover and troubleshoot problems, let's look at some of the more day-to-day management challenges that you're likely to face and ways to handle them.

Managing your GPO infrastructure involves a number of key tasks, which fall outside the scope of basic troubleshooting, including:

- Planning for new GPOs and seeing their potential effect on users and computers in a forest
- Backing up and restoring individual GPOs in a forest
- Viewing the current effective policy for a given workstation
- Scripting GPO creation and editing.

Some of these capabilities are available out-of-the-box, but most of them either don't exist yet or are available with third-party tools. In this section, I'll talk about each of these capabilities and how you can achieve them using available techniques and tools.

Resultant Set of Policy (RSoP)

I've referred to the idea of *Resultant Set of Policy (RSoP)* several times in this book. RSoP is the ability to view and plan for the effects of GPOs on a particular user-and-computer combination. You need to perform real-time analysis of the effects of multiple GPOs on a given user or computer in a given OU or domain. You also need to be able to perform what-if scenarios before you move a user or computer to a new OU or change its group membership. It's these two requirements that really encompass RSoP.

In "Group Policy Result Tool (gpresult.exe)" earlier in this chapter, I described how the gpresult.exe utility, to a small degree, lets you view the current effective policy on a given user and computer. However, you can't run this tool against remote computers easily, and it can't help you if you want to perform what-if scenarios on future changes.

This is where RSoP tools like FAZAM 2000 come in. FAZAM 2000 has a planning and analysis module that lets you create what-if scenarios and experiment with the effects of moving a user or computer into new OUs or new groups (or removing them from groups). Figure 6.17 shows an example of the kinds of scenarios you can create with FAZAM.

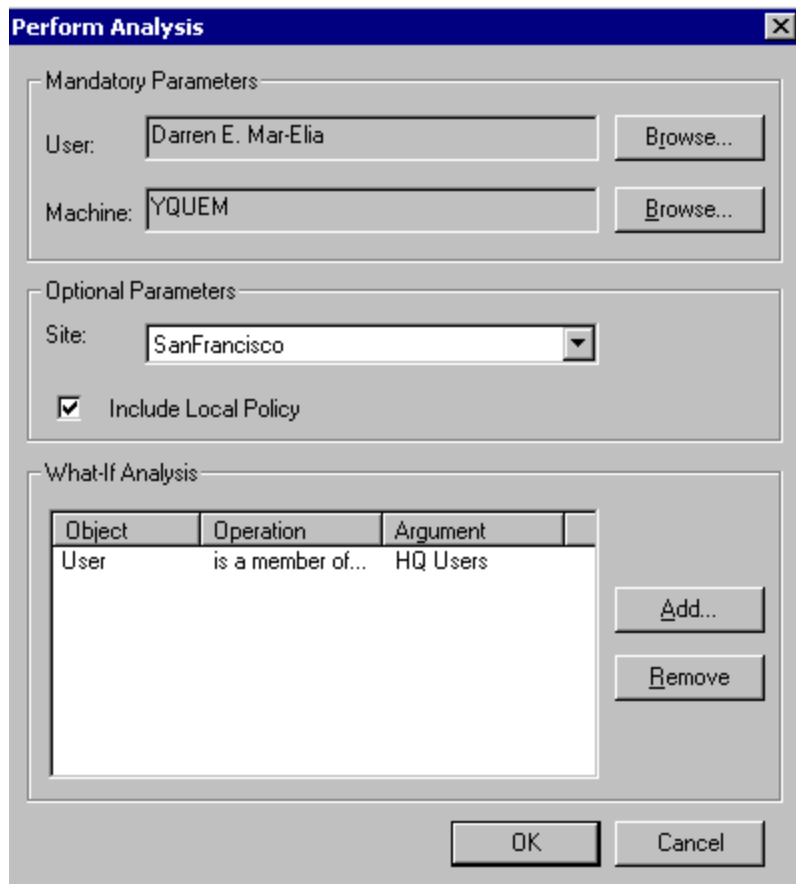


Figure 6.17: Creating a scenario using the what-if planning feature in FAZAM 2000.

In this figure, I'm asking the tool to tell me what my effective policy will be if the user account for user Darren Mar-Elia, logged on to the computer called Yquem, is added to the HQ Users group. I'm also telling the tool to include in the calculation site-based policy for the San Francisco site and the local GPO. Once I complete the scenario, FAZAM calculates my new effective policy settings and reports three things.

- **User hierarchy**—Which GPOs apply to the chosen user from which containers
- **Computer hierarchy**—Which GPOs apply to the chosen computer from which containers
- **Resultant policy**—What the effective policy settings are at the computer and for the user.

Figure 6.18 shows the results of this analysis.

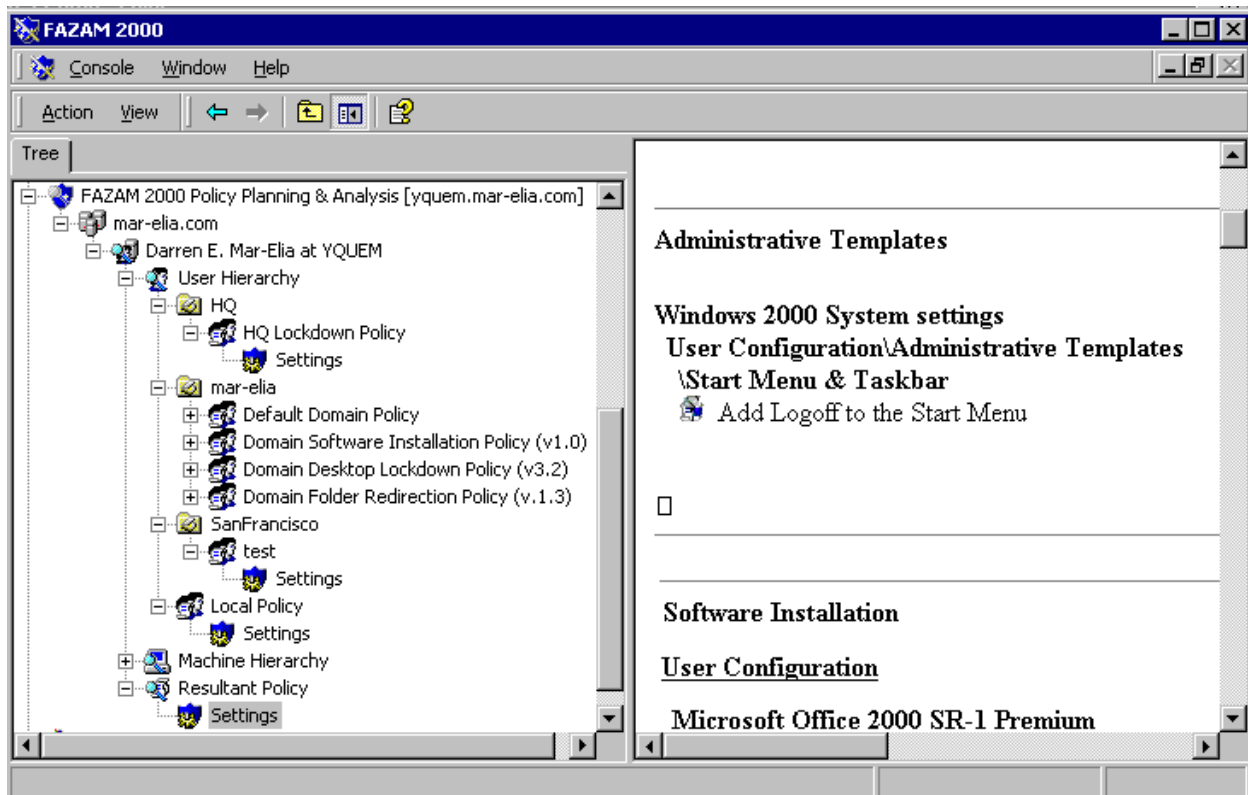


Figure 6.18: The results of an RSoP analysis in FAZAM 2000.

As you can see in this figure, Darren Mar-Elia at Yquem is the scenario I established, and the resulting user and computer policy hierarchies are shown in the scope pane on the left. The results pane on the right shows a portion of the individual policy settings that the user and computer in my scenario will receive. From this information, I can determine what effect adding my user account to the HQ Users group will have.

You can see that if you have a large GPO infrastructure and are making frequent changes to both GPOs and users or computers, this kind of functionality can be very useful. Microsoft has plans to provide some RSoP tools in the next version of Win2K, but for now, you'll need to rely on tools like FAZAM 2000 to provide this valuable capability.

Backing Up and Restoring GPOs

Another function missing from the Win2K product line is the ability to easily back up and restore individual GPOs. You can, of course, use backup utilities to back up the System State on a given domain controller, including AD and SYSVOL. Unfortunately, the restore process isn't granular and doesn't allow you to restore selective parts of these elements.

Remember that a GPO consists of two parts—the GPC and the GPT. This means that a backup utility needs to back up both AD and SYSVOL components for a particular GPO, then restore them, while maintaining any unique elements of that GPO. In addition, because container objects like sites, domains, and OUs link to a GPO by its GUID, you need to ensure that when the

backup utility restores a GPO, it re-creates its original GUID in AD. Otherwise, you have to re-link container objects to the new GPO.

Fortunately, FAZAM 2000 (and the Reduced Functionality Version, called FAZAM 2000 RFV, that comes in the Windows 2000 Resource Kit) provides a backup and restore capability that lets you back up individual GPOs to a file system, then restore them to the original domain or, what is more interesting, to other domains and even other forests. Figure 6.19 shows FAZAM 2000's backup capability.

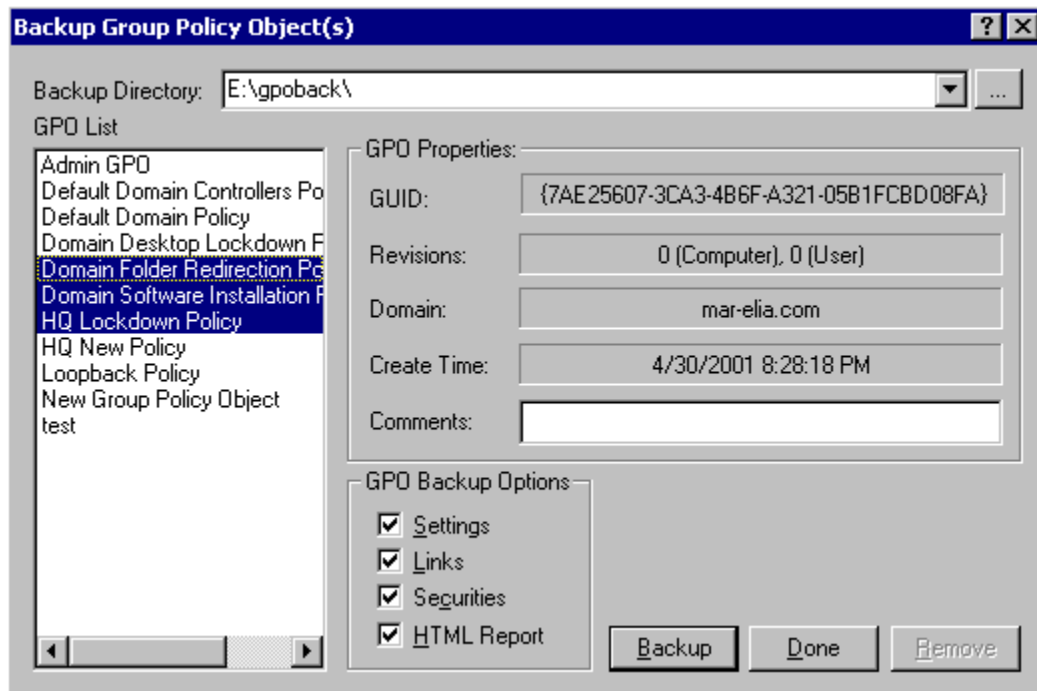


Figure 6.19: Using the backup capability in FAZAM 2000.

The Role of the Local GPO

Throughout this book, I've focused mostly on AD-based GPOs because this is where most of the power and features of Group Policy are enabled. However, as I finish this chapter and this book, I want to talk about the role of the *local GPO* in Win2K.

As I mentioned earlier in this book, the local GPO is stored on every Win2K device in the %systemroot%\system32\GroupPolicy folder. The local GPO is processed first, before site, domain, or OU policies, so settings that are specified in the local GPO are overwritten by GPOs processed later. However, the local GPO can be useful in a number of different scenarios, so I'll describe how it works and how to manage it.

The easiest way to open a local GPO on a workstation or server is to choose Start>Run, then in the Run dialog box, type *gpedit.msc*. This starts a pre-created Group Policy MMC snap-in tool that is focused on the local GPO. The first thing you'll notice about the local GPO is that it doesn't support a few of the policy functions that are present in AD-based GPOs—for example, Software Installation or Folder Redirection policy.

Despite these limitations, local GPOs do provide some interesting capabilities. First, if you don't have an AD infrastructure supporting your Win2K devices, you can of course use local GPOs to manage policy. The challenge here is to manually edit each local GPO on each workstation or server in your enterprise to set local policy. Well, that's not exactly the case.

Because local GPOs are stored in the local Win2K file system, and not spread across AD and SYSVOL, you can simply copy the contents of the %systemroot%\system32\GroupPolicy folder from one computer to the next, and the policy settings that you set on the source are used by the destination. This makes it easy to set local GPO policy the way you want on one computer, then "clone" it to all of the other Win2K devices without using special tools.

Another neat thing about local GPOs is that you can use them to view the effective security policy on a given computer. For whatever reason, Microsoft decided to provide this view of effective policy in the Security Settings node in the Group Policy MMC snap-in but not in any other nodes. The local GPO shows a view of the local security setting compared to the effective settings—those that are being enforced by AD-based GPOs. You can see this in Figure 6.20 below.

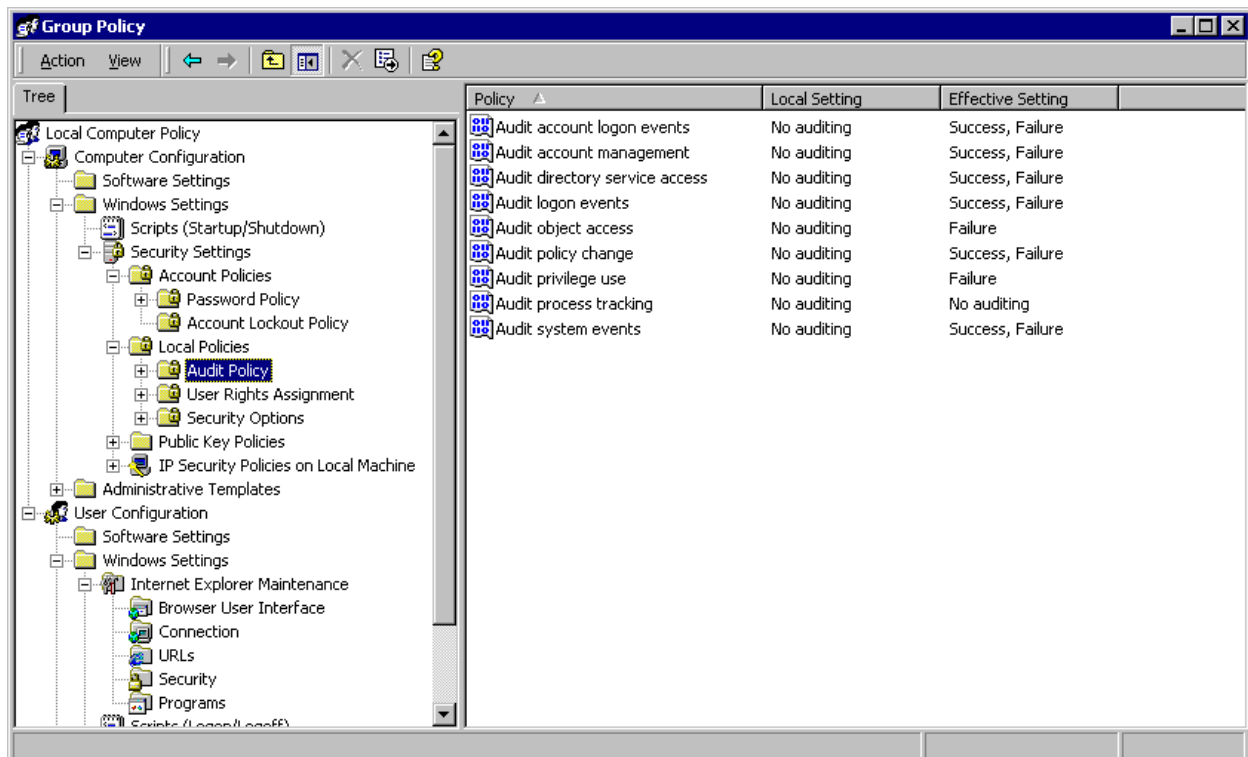


Figure 6.20: Viewing the effective security policy compared to local security policy in a local GPO.

In this figure, the Local Setting column shows what audit policy has been set in the local GPO—in this case, none. However, based on information received from AD-based GPOs that this computer has processed, the Effective Setting column says what the current audit policy really is for this computer. This can come in handy when you're trying to troubleshoot security-related problems in your AD infrastructure. Unfortunately, however, there is no way to know

which AD-based GPO delivered these auditing settings to the computer. This is where a more full-featured RSoP tool comes in handy.

Summary

In this chapter, I started by discussing some of the most common problems you're likely to encounter in your GPO infrastructure. Then I moved on to describe useful tools for troubleshooting GPOs. From there, I examined the multitudinous logging options available to you to keep track of what your GPOs are really up to. Finally, I discussed some of the management issues with AD and local GPOs and the tools that you can use to address them.

I'd like to thank you all for reading and supporting this book, and I hope it's been as useful for you to read as it was fun for me to write!