Realtime publishers
"Leading the Conversation"

May 2008
Volume 1, Number 4

# Windows Administration
## *in Realtime*


**SmartShedding® Technology**

Allows the master outlet to sense when your computer has either been turned off or has gone into sleep mode, so it can shut off power to peripherals plugged into the controlled outlets–saving you power and money.

# Your data should last forever.
# And so should our planet.

## Save $40* a year on your electric bill with the most efficient battery backup yet.

### Let's protect what's important

What's in your computer? Photos, music, personal files, financial data, broadband access, videos, and more. Your computer has never been more important, and yet it has never been at higher risk for damaging power surges and other disturbances.

So like most people, you need to protect your assets. But like most people, you'd also like to protect the environment. With our new energy conscious products, you can do both. Energy efficient by design, our new smart products protect the power going in your computer, at a cost that is quickly offset by big energy savings. How? Not only do the new Back-UPS ES® and SurgeArrest® use power very wisely, they also boast a master/controlled outlets feature, which automatically powers down idle devices to conserve energy.

**APC power protection products are available at:**

BEST BUY · CDW · Circuit City · STAPLES (that was easy.) · Office DEPOT

*"The price tag on the new UPS is $99.99. While I'm not in the habit of endorsing products in this blog, if you're in the market for a workstation-class UPS, why not opt for the greener option?"*

*- Heather Clancy*
*ZDNet.com*

In fact, while protecting your power supply, we're up to 5 times more energy efficient than any other solution. By saving you $40 a year in energy costs, our Back-UPS ES pays for itself in 2 short years. The high frequency, low copper design has a smaller transformer and environmental footprint. Even the packaging has been carefully selected and manufactured to maximize use of recycled materials and minimize waste.

In this world, every decision you make counts. So protect your power with a battery backup that works to protect the environment. It conserves power, it pays for itself, and it's backed by APC's 20-plus years of legendary reliability. For more information on this or our other great products, or for information about environmentally responsible disposal of your old battery, visit ***www.apc.com***

INNOVATIONS 2008 Design & Engineering Showcase Honors

## Energy Efficient Solutions for Every Level of Protection:

Save $25 per year* on your electric bill!

### Surge Protection

Starting at $34

Guaranteed protection from surges, spikes, and lightning.

7 outlets, Phone/Fax/Modem Protection, Master/Controlled Outlets

SurgeArrest® P7GT

Save $40 per year* on your electric bill!

### Battery Back-UPS®

Starting at $99

Our most energy efficient backup for home computers.

10 outlets, DSL and Coax protection, Master/Controlled Outlets, High Frequency Design, 70 minutes of runtime†

Back-UPS® ES 750G

*APC can help with your other power protection needs. Visit apc.com to see our complete line of innovative products.*

## Enter to Win a Back-UPS® ES 750G! (a $99.99 Value)

*Also, enter keycode to view other special offers and discounts.*

Visit www.apc.com/promo Key Code c437w or Call 888.289.APCC x9542 or Fax 401.788.2797

APC® Legendary Reliability®


e-mail: esupport@apc.com • 132 Fairgrounds Road, West Kingston, RI 02892 USA • 998-0968 †Runtimes may vary depending on load.
*Average savings are based on comparable competitive models, and are comprised of two energy saving features: An ultra efficient electrical design, and the master/controlled outlets feature.

# Letter from the Editor

# *Internet Explorer 8's Potential Vista Problem?*

*by Greg Shields*

After an onslaught of pressure within Web developer and standards circles, Microsoft in early March announced a reversal of their earlier decision to make optional Internet Explorer 8's "super-standards" mode. This change of opinion makes mandatory the way in which IE8 renders Web sites. Specifically, IE8 will render Web sites according to established Web standards rather than Microsoft's own definitions.

In the long-run, this decision is a great move for Microsoft. It puts their Web browser on par with others who already render pages according to established Web standards. The bad news is that it also has the potential to break many Web sites that were previously "hacked" to enable them to work with IE7 and earlier versions.

Here's my conundrum: We've seen this problem before with Windows Vista. With Vista being the first OS written under Microsoft's Secure Computing Initiative, the decision was made to force all drivers and applications out of the kernel, or what the techies call "ring zero." This decision significantly improves the security posture of the Vista OS, but does so at the expense of forcing virtually all application and driver vendors to completely re-code their wares.

With Vista, many users and administrators alike saw the resulting application conflicts as an outward (and incorrect) sign that "Vista sucks." The resulting adoption rate for Vista in enterprises slowed to a crawl, where it remains today even as most applications and drivers have now been made Vista-ready. First impressions are hard to change. Perception is reality.

IE8 is yet another awesome story of Microsoft coming clean with the sins of its past. Yes, Microsoft should have followed Web standards from the get-go. But hindsight is 20/20. With IE at 90% market share, a change of this magnitude has the potential to immediately produce a backlash similar to Microsoft's Vista decision if not handled correctly.

Where Microsoft failed with Vista was in getting administrators on board. Being closest to the technology, administrators are often its loudest proponents—or opponents—of change. When they see value in a change, they'll tell you about it. When they don't, they'll tell you about it even louder. With IE8's potentially difficult growing up period coming soon, one hopes that Microsoft will learn from previous mistakes and ensure admins are on board. If not, the company faces the wrath of a resounding (and fully incorrect) round of "IE8 Sucks!"

There's more than IE8 on the menu in this issue. Darren Mar-Elia will bring us a down-and-dirty look at troubleshooting Group Policy. I analyze the Windows DNS resolution process, while J. Peter Bruzzese looks at the Exchange 2007 upgrade process, and Jeff Hicks talks about local password management. There's always great stuff to learn here at *Windows Administration in Realtime*.

What about you? We're always on the lookout for ideas. Are you interested in learning more about a particular product or feature of Windows administration? Let us know at feedback@realtimepublishers.com.

# The Industry Outlook

# *Hot Topics in the IT Arena*

*by Don Jones*

### *Free Workflow for AD Management*

At the recent Director Experts Conference in Chicago, NetPro announced the release of a free software solution that will provide workflow, reviews, approvals, and task scheduling for AD management. The solution will include policy-based standards enforcement and will integrate with the company's ChangeAuditor product so that request details become part of a permanent audit log.

### *Apple iPhone, Again*

The product that couldn't get enough press coverage gets more: Apple announced that the v2.0 update to the iPhone software will introduce Exchange Server and ActiveSync support. Early Apple documents indicate that the company plans to make a go for the Blackberry user market.

### *Windows Vista, Again*

The product that can't get enough bad press coverage gets more: Windows Vista users are furious about Microsoft's announcement that Vista's Service Pack 1 (SP1) might not show up in Windows Update if you have any of 31 language packs installed, any earlier versions of SP1 installed, are missing certain pre-req updates, or if you have certain device drivers. The device driver "black list" is the contentious item; MS claims certain drivers are "problematic" on SP1. Users are having a difficult time identifying the drivers that SP1 doesn't like on their computers, essentially leaving them in a lurch for installing the update. Common "problem" drivers are SigmaTel audio drivers; with SigmaTel now out of business, it's difficult to determine who will fix the problem, which affects many Sony, Dell, Gateway, and other popular brands of computers.

### *Dell Says SSDs Okay*

Dell claims that reports of 20 to 30% return rates on laptops containing solid-state drives (SSDs) are inaccurate and that the drives are performing as well as or better than the company's traditional hard disks. Although the issue currently revolves around laptops, many analysts have speculated that SSDs will soon make their way into small form factor (1U) infrastructure servers running Windows Server 2008 Server Core and acting as "black boxes" for low-storage applications such as DNS Server, DHCP Server, and so forth.

### *Easy on the Mac*

Computerworld reports that Auto Warehousing Company's (AWC) move from Windows-based PCs to Macs hasn't been as easy as everyone hoped. Workers protested the change, and customers perceive the switch as unnecessarily costly. The company's CIO argues that AWC will save more than $1.82 million over 3 years against a switch cost of $335,000.

### *Office = Standard*

Bill Gates recently argued before the US House of Representatives, supporting the International Committee for Information Technology Standards (INCITS), which is considering adopting Microsoft's Open Office XML (OOXML) format as an ISO standard. Exactly why US lawmakers are concerning themselves with Office file formats in the current political and economic situation is unclear, but it is known that the INCITS executive board is generally in favor of granting OOXML status as an ISO standard.

# Group Policy Troubleshooting Tips and Techniques

*by Darren Mar-Elia*

Windows Group Policy presents a dichotomy. On the one hand, you rely on this technology to deploy critical security and lockdown configurations to your desktops; on the other hand, the technology can be complex to use, with many moving parts that don't always work the way you expect. In this paper, I'll start off by discussing how Group Policy processing works so that you get a fundamental understanding of the plumbing of this technology. I'll then look at where problems can arise and provide tips and techniques for solving even the thorniest Group Policy processing issues.

## *Group Policy Processing Uncovered*

Group Policy processing is like a precision watch. There are many moving parts that must work together in order for successful processing to occur. It requires both your Windows server and desktop infrastructures to be healthy, and it requires your understanding of the underlying targeting and security model in order to ensure that the computers and users are receiving the configuration settings you expect. Let's start off by looking at how the processing cycle works; consider the following high-level "rules" regarding Group Policy processing:

- Group Policy is processed by user and computer objects in an Active Directory (AD) environment, but you can use security group filtering to target specific users or computers.
- Every Windows server or workstation has a local Group Policy Object (GPO—or multiple local GPOs in the case of Windows Vista and Windows Server 2008) that is processed regardless of whether that system is part of an AD domain.
- There is an order of precedence to Group Policy processing within an AD environment:
  - Local GPOs are processed first
  - AD site-linked GPOs are processed next
  - Then AD domain-linked GPOs are processed
  - Finally, Organizational Unit (OU)-linked GPOs are processed
- Within the order of precedence, the "last writer wins" rule applies. That is, any conflicting settings processed early in the order can be overwritten by settings processed later, such as domain-linked GPO settings being overwritten by OU-linked GPO settings. Local GPOs process first and always have the lowest order of precedence (that is, they can be overwritten by any AD-based GPOs).

It is also important to understand how GPOs are stored in AD and how they are processed by client machines.

## *GPO Storage*

When you make a change to a GPO using the GPO editor (GP Editor) or you create a new GPO using the Group Policy Management Console (GPMC), by default, what is happening under the hood is that the GPO is stored in two distinct pieces on your AD domain controllers. By default, this all happens at the PDC-emulator domain controller first. That is, GPMC and GP Editor focus on this domain controller by default whenever you make GPO changes. The GPO itself is stored in both AD and the SYSVOL file system that exists on each domain controller in a domain. The AD portion of a GPO is called the Group Policy Container (GPC) and exists in your AD domain under the CN=Policies, CN=System container. Each GPO is stored as a Globally Unique ID (GUID)-named object under that Policies container, as Figure 1 shows. The GPC typically contains general information about the GPO, but some policy areas, such as Software Installation policy, store some of the actual settings related to that policy within this AD portion of the GPO as well.

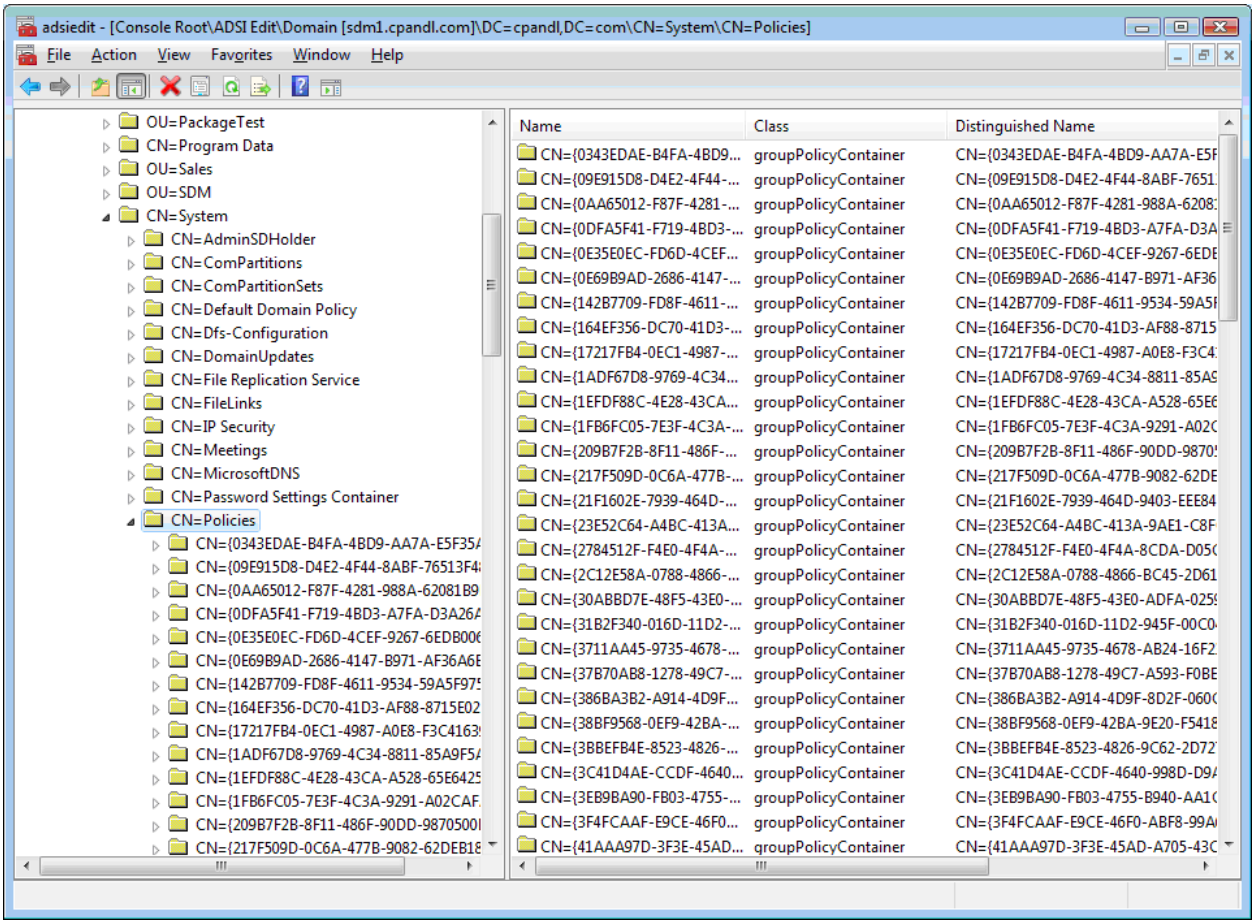


*Figure 1: Each GPO is stored as a GUID-named object under that Policies container.*

The portion of the GPO that is stored in SYSVOL is called the Group Policy Template. The GPT is where most policy areas store the actual settings that you define within the GPO, in files that are stored in \\<domain>\sysvol\<domain>\policies\<GUID of GPO>. Whenever you make changes to a GPO, those changes are written to the GPC and GPT and then replicated from the PDC-emulator domain controller to all domain controllers within the AD domain in which the GPO resides. In this way, GPO information is made available to all clients of any GPO in the domain. Thus, a client in a remote site that has a domain controller can retrieve GPO information from its local domain controller without having to cross a Wide-Area Network (WAN) link. But as I will talk about later, this setup can also be a source of problems because these changes must replicate successfully to all domain controllers in a domain in order for all clients to receive the latest and greatest policy settings. Failure of this replication process for any reason can mean that some clients don't get the most up-to-date policies.

### *Client-Side Processing*

The Group Policy client is where all the action is when it comes to troubleshooting Group Policy problems. The client reads GPO information from AD and SYSVOL and then does the work of reading the settings that apply to it and processing them on the local system. And it's the client portion of the process that usually causes much of the problems in Group Policy. Client-side processing of Group Policy is broken into two pieces—Core processing and Client Side Extension (CSE) processing.

Core processing, also known as "Group Policy Infrastructure" if you are viewing processing status in tools such as the GPMC Group Policy Results Wizard, is the portion of the processing cycle where the client (user or computer) queries AD to determine which GPOs apply to it, whether those GPOs have changed since the last time processing occurred, and

which CSEs need to be called during the CSE portion of the processing cycle. If you have infrastructure problems in your environment—such as DNS, Network, File Replication Service (FRS), or AD replication issues—they are most likely to impact the Core processing part of the cycle. One problem with Core processing is that it is fairly brittle. If one part of the Core processing cycle fails (for example, a domain controller cannot be located or the slow-link test between client and domain controller fails), then all Group Policy processing will fail.

CSE, or CSE processing, is the portion of the cycle where Group Policy settings are actually processed and applied to the user or computer. Each policy area (for example, Administrative Templates, Software Installation, Folder Redirection, and so on) roughly corresponds to a CSE. During the CSE portion of the processing cycle, each CSE installed on the client takes the list of GPOs generated during the core processing cycle, reads the settings found in those GPOs, and applies them to the system. During this phase, a single CSE can typically fail and the others will still work. In some rare cases, you may see a CSE fail so completely that it causes the remaining CSEs to not run, but this situation is the exception rather than the rule.

With this overview of Group Policy processing, let's get to the heart of the matter and explore how we can troubleshoot Group Policy problems when they do arise.

### *The Categories of Group Policy Problems*

I typically break Group Policy problems into three main categories:

- Infrastructure Problems: Problems related to DNS, network, FRS, or AD that prevent Core processing from succeeding
- Client-Side Problems: Issues on the client that prevent Core or CSE processing from succeeding; these can include network issues from the client's perspective, key services not running on the client, or other issues that prevent processing from proceeding
- Misconfiguration Problems: Problems caused by administrators misconfiguring or mis-applying Group Policy such that settings are not received by the correct users or computers; this area is the most common for problems and the easiest to fix

Let's take a look at steps you can take to troubleshoot these problem categories.

### *Where to Start Troubleshooting*

Whenever you suspect you have problems with Group Policy, the first place to start troubleshooting is with Resultant Set of Policy. RSoP helps you determine what is happening with respect to Group Policy for a given computer or user. The tools you can use to gather RSoP information are the command-line gpresult.exe that comes with Windows XP and later and the GPMC GP Results Wizard, which is a graphical version of gpresult.exe. I like to use the GPMC version of the tool because it provides an easy-to-read HTML report of what is going on with a given remote system, and it's easy to run. From the GPMC, simply right-click the Group Policy Results node, start the wizard, select the local or remote computer to which you want to connect, and then select the user on that computer whose Group Policy settings you want to collect. The tool goes out and gathers information about what happened during the last Group Policy processing cycle.

**Note: When using Group Policy Results to gather policy processing results against a remote computer, you need to be able to connect to that remote machine's WMI repository. This requires use of the Distributed COM (DCOM) protocol. If you are using Windows Firewall, adding the "Remote Administration" exception will allow DCOM traffic to successfully pass.**

Once you complete the wizard, three tabs will appear on the right side of the GPMC, as Figure 2 shows.

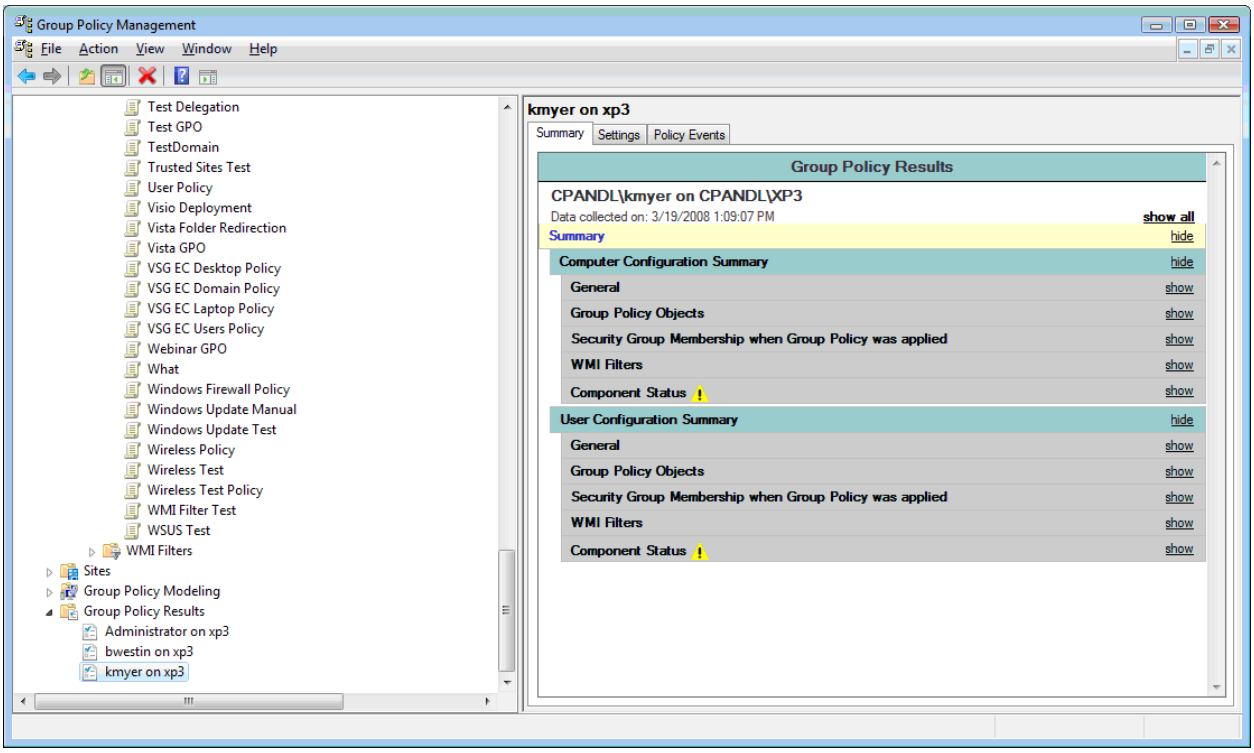


*Figure 2: Using the GPMC for troubleshooting.*

The report returned provides Summary, Settings, and Policy Events tabs. The Summary tab is the most useful for troubleshooting the Client-Side and Misconfiguration categories of problems. For example, using the GPOs section within this Summary report, you can see which GPOs apply to a particular computer or user. You can also see the list of GPOs that would normally apply but that are denied for various reasons, as Figure 3 illustrates.

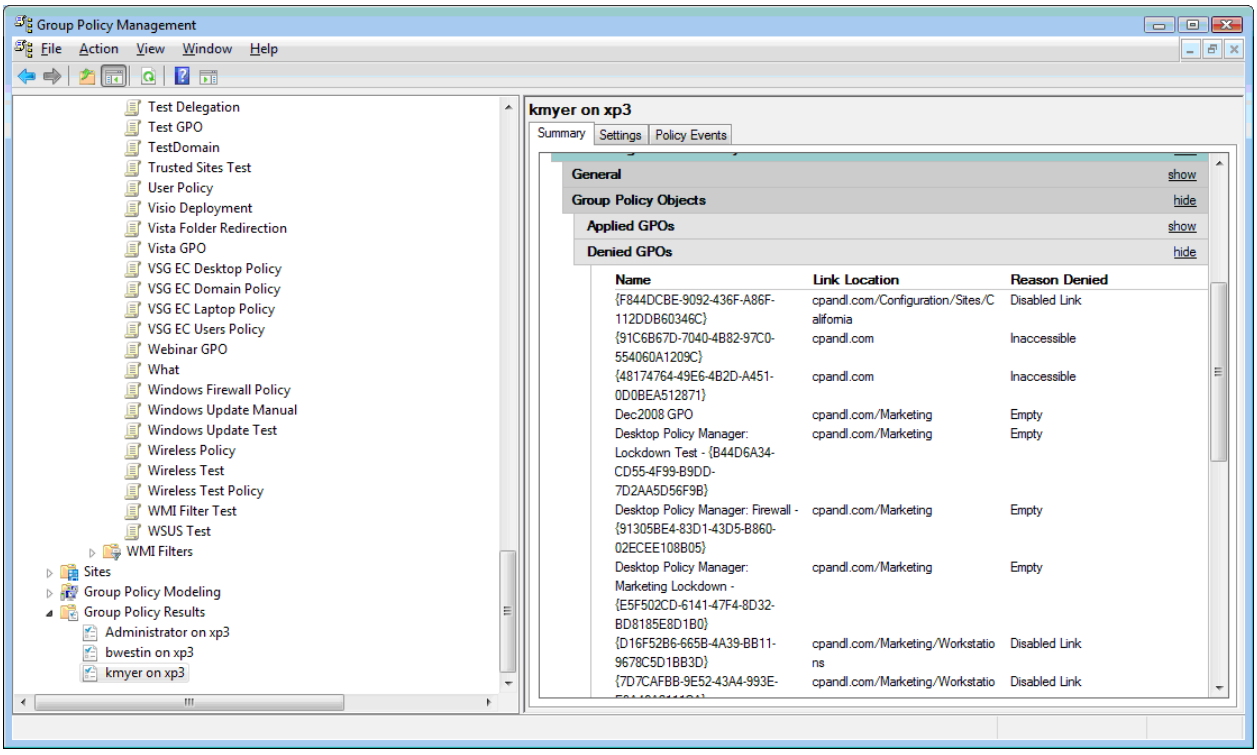


*Figure 3: Viewing denied GPOs in the Summary GPMC report.*

This will quickly point out misconfiguration issues. For example, if you are trying to deploy a policy to a given computer or user and you don't see that GPO in either the Applied GPOs or Denied GPOs list, the chances are good that you have not linked that GPO to the right location. Alternatively, the GPO has not yet replicated to the domain controller that the client is using to retrieve policy information.

The other interesting section within this Summary report is the Component Status section. This section shows, for the computer and user, whether there have been errors in either the Core (Group Policy Infrastructure) or CSE portion of processing, as Figure 4 shows.

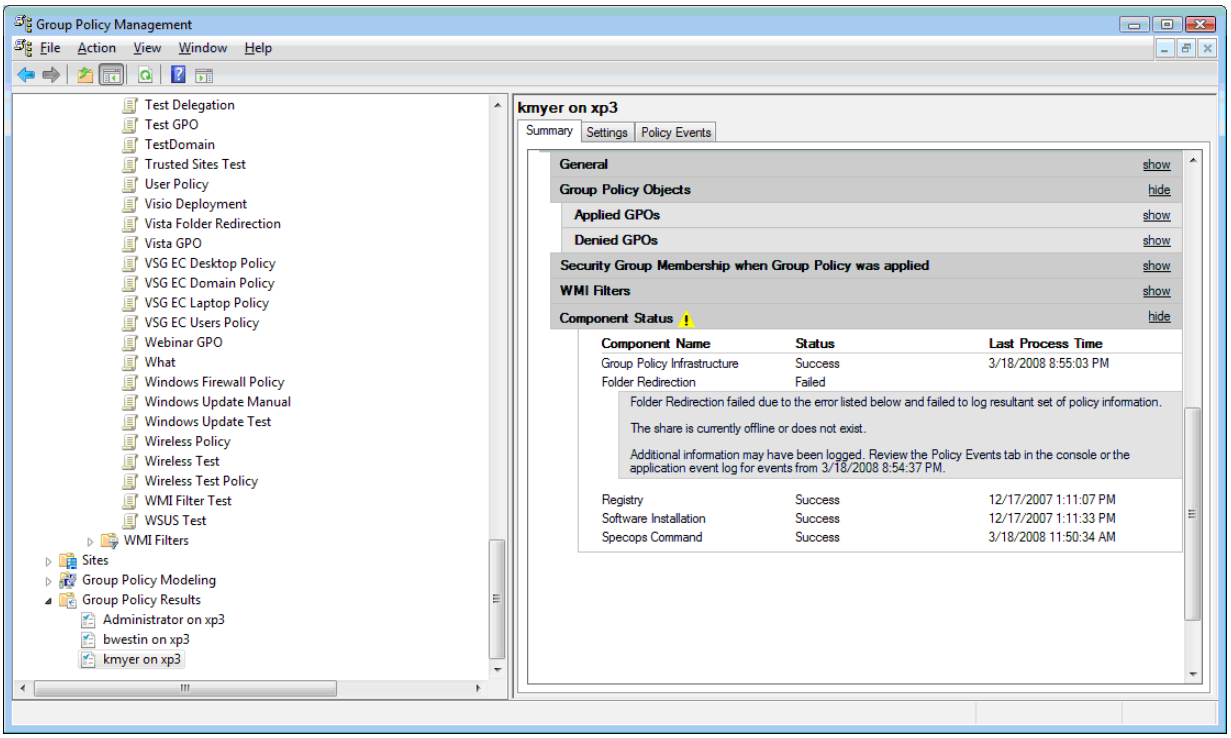


*Figure 4: Exploring errors in the Group Policy Infrastructure and CSE portion of processing.*

**Note: In Figure 4, Core (Group Policy Infrastructure) processing for the user succeeded, but Folder Redirection failed because the share being redirected to was not online or otherwise unavailable.**

The Settings tab within this Group Policy Results report will show you all the policy settings that have been applied to this computer or user, and which GPO delivered them (that is, the Winning GPO). It will not show you whether you have conflicting settings within the GPOs that apply to the computer or user, however. If you are not seeing the effects of a policy setting to a computer or user, this report can tell you whether that setting is truly being applied. If you see a setting has been applied but the effects of that setting are not showing up, remember that some settings require the application that respects them to be restarted or, in some cases, they may require a Windows reboot or re-logon of the user to take effect.

The final tab is the Policy Events tab. This tab presents a view of the Application event log on the remote system, filtered specifically for Group Policy–related events. If you are seeing problems in the Component Status section of the Summary tab, you might find more details on the errors you are seeing by looking in the Policy Events for that computer.

### *Drilling into Client Problems*

After you've used the Group Policy Results wizard to identify a problem, the next step is finding out what is causing the problem. In the case of client-side problems, either in Core or CSE processing, you might need to resort to lower-level logging to expose the problem. There are a number of logs that can be enabled on a Windows system that provide detailed trace logging for both Core and many of the CSEs that Windows ships.

The Userenv.log file is chief among the logs that can be used to troubleshoot both Core and CSE issues. It must be enabled for verbose logging before it can be used, however. See the Microsoft article "How to Enable User Environment Debug Logging in Retail Builds of Windows" at http://support.microsoft.com/kb/221833 for information about how to do this or download the GPOLOG.ADM template from http://www.gpoguy.com/gpolog.htm, which lets you enable a variety of core and CSE logs, including userenv.log.

Note: SDM Software's GPExpert Log Analyzer, which is part of the GPExpert Troubleshooting Pak, also lets you enable this log and makes these logs easy to read with consistent parsing and troubleshooting assistance. The Log Analyzer supports the userenv.log file as well as the logs for all other CSEs that support trace logging, such as Security, IE Maintenance, Software Installation, and Folder Redirection.

There is also a class of client problems related to an inability to read the contents of AD and SYSVOL. These problems can be difficult to pin down but often relate to client network stack timing issues, where the client's network interface is not fully initialized when Group Policy processing is occurring. The Microsoft article "Group Policy Application Fails on a Computer that Is Running Windows 2000, Windows XP Service Pack 1, or Windows XP Service Pack 2" at http://support.microsoft.com/default.aspx?scid=kb;en-us;840669 discusses one workaround for this issue in pre-Windows Vista systems. Other issues related to an inability to read the contents of AD or SYSVOL could relate to server-side infrastructure problems, which we'll look at next.

### *Identifying Infrastructure Problems*

Problems within your server infrastructure that are causing Group Policy issues can be difficult to resolve. The key thing to remember is that a number of pieces of your infrastructure must work in concert in order for policy to work properly. Clients must be able to resolve their closest domain controller via DNS. They need to be able to authenticate to AD using Kerberos and, in pre-Vista/Server 2008 systems, clients must be able to ping their local domain controller using ICMP to determine whether a slow-link has been detected. If any of these processes fail, Core processing will likely fail and Group Policy processing will not occur.

In addition to these issues, any changes that you make to GPOs must successfully replicate to all domain controllers in a domain, both from an AD and SYSVOL perspective. SYSVOL replication has historically been problematic which has caused issues with Group Policy. GPO changes are typically originated at the PDC-emulator domain controller and then propagated to all domain controllers in a domain. Client systems try to grab policy settings from the domain controller in their own AD site, which means that any changes you make to GPO settings must successfully replicate to all domain controllers before clients will get up-to-date policy. If AD or SYSVOL replication is broken, this process of getting the most up-to-date policy breaks down.

SYSVOL replication problems in particular can be tricky. In some cases, whole GPOs may not have replicated to a remote domain controller. In other cases, the remote domain controller might only have partial replicas or permissions on the replicas may be inconsistent with the PDC-emulator for a given GPO. This situation causes unexpected results or lack of access by the client. The GPOTool.exe command-line utility, which is part of the Windows Resource Kit tools, can help identify Group Policy-related replication problems by listing the GPO version numbers of both the AD and SYSVOL portions of a GPO on each domain controller in a given domain, as Figure 5 shows.

In Figure 5, I am telling GPOTool.exe to look for copies of the "Default Domain Policy" GPO on all of my domain controllers. I'm also telling it to check the Access Control Lists (ACLs) on each domain controller to ensure that permissions between the replicas are consistent. This tool can be useful to determine whether you have AD or FRS replication issues that might be impacting Group Policy processing. In addition, GPExpert Log Analyzer has the ability to perform a diagnostic test that compares GPO versions between a client, the domain controller it is using for Group Policy processing, and the PDC-emulator domain controller to quickly and easily determine whether a client has processed the most recent version of policy or if its local domain controller has replicated the most recent version of policy.

Once you know that there is, for example, a SYSVOL replication issue, you can begin to troubleshoot this issue. For more information about the steps you can take to get SYSVOL replication working again, check out http://support.microsoft.com/kb/315457/en-us.

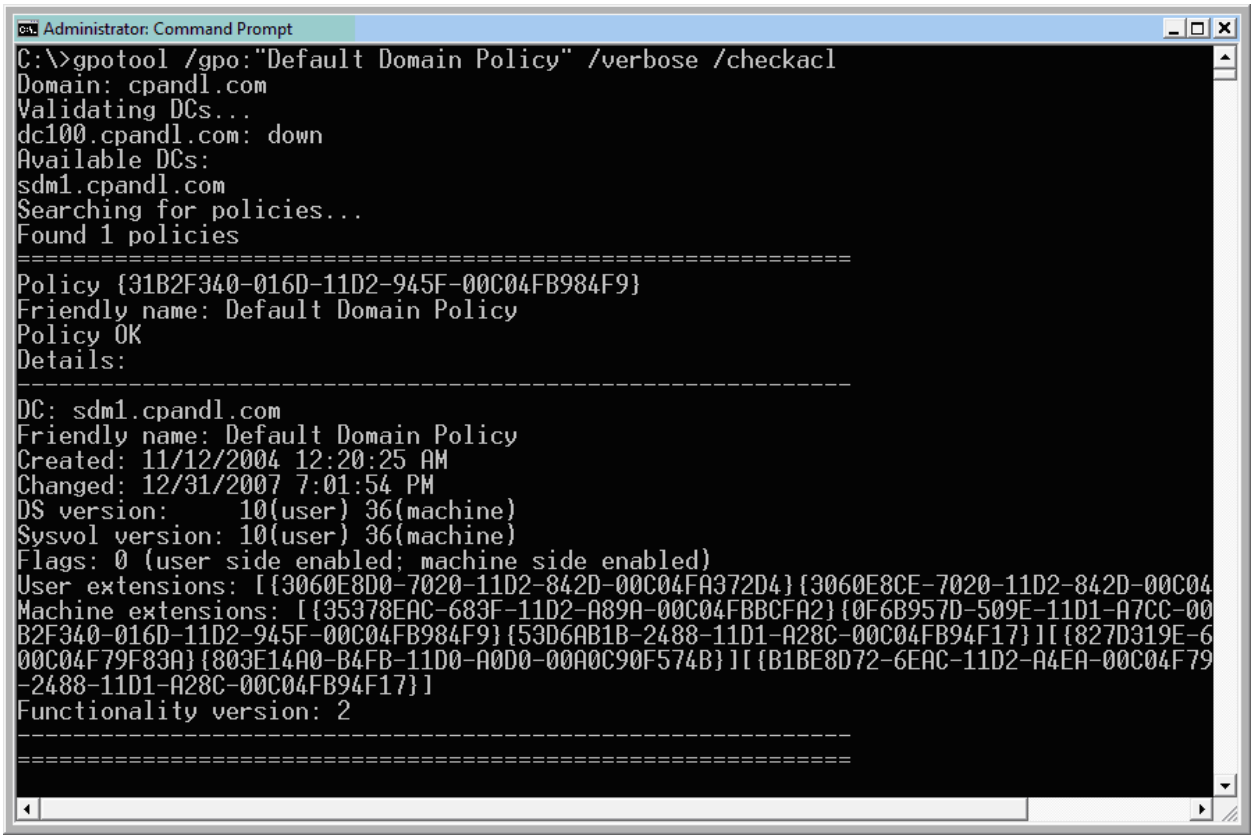


*Figure 5: Employing the GPOTool.exe command-line utility to identify replication problems.*

## *Summary*

Group Policy Troubleshooting can be a complicated process. Misconfiguration problems, infrastructure issues, and client-side problems can all contribute to policy processing failures. It's important to first identify and isolate the problem using a tool such as the GPMC Group Policy Results wizard, and then use additional tools such as trace logs and products like the GPExpert™ Troubleshooting Pak to drill into and fix the underlying problem. ◆

*Darren Mar-Elia is CTO & Founder of SDM Software, Inc., a Windows Group Policy management solutions company. Prior to starting SDM Software, Darren was Sr. Director of Product Engineering for DesktopStandard -- a Group Policy tools company that was acquired by Microsoft. Darren was also Quest Software's CTO for Windows Management and is a Microsoft MVP for Group Policy. Darren has more than 20 years of IT experience in systems and network administration design and architecture. His IT expertise was on large-scale enterprise implementations of Windows infrastructures in distributed and data center environments. Prior to joining Quest, he worked as director of distributed systems architecture and planning for Charles Schwab & Co., Inc. In that capacity he was technical lead for the company's Windows NT & 2000 design and migration efforts and help create the company's strategy for Linux.*

*He maintains the popular Group Policy resource web site at www.gpoguy.com and has been a contributing editor for Windows IT Pro Magazine since 1997. He has written and contributed to twelve different books on Windows including, most recently, the "Windows Server 2008 Security Resource Kit Guide", the "Windows Group Policy Guide", published by Microsoft Press in 2005 and "The Definitive Guide to Windows 2000 Administration," "The Definitive Guide to Windows 2000 Group Policy" and "The Tips & Tricks Guide to Group Policy," all published online by Realtimepublishers.com. Darren also speaks frequently at conferences on Windows infrastructure topics.*

# Don't Leave Your Group Policy Health to Chance.

## ELIMINATING RISK IS AS SIMPLE AS SPOTTING RED FROM GREEN.

Your Group Policy health is not something to gamble with. You use it to configure critical security and lockdown settings on your Windows systems. Why leave its performance to chance?

The GPExpert™ Troubleshooting Pak gives you four products that work together to ensure that all the moving parts in your complex Group Policy infrastructure are working properly. Collect information from a variety of sources with the GPExpert Health Reporter and get a quick visual "red or green" health status so you know when something is dicey. When trouble strikes, use the GPExpert Log Analyzer to pinpoint the source of your Group Policy problem. Or, use the GPExpert Group Policy Spy to watch registry policy activity in real time to find conflicts. Finally, give your help desk staff the GPExpert Status Monitor, running on your users' desktops, to easily see when Group Policy is functioning correctly on their machine.

With the GPExpert Troubleshooting Pak you never have to leave Group Policy health to chance. Visit **http://www.sdmsoftware.com/products** today to get your free trial.

sdmsoftware

The Group Policy Experts

# The Deep Dive

# *Back to Basics: Unraveling Windows DNS Resolution*

*by Greg Shields*

The Domain Name System (DNS) for translating fully qualified domain names (FQDNs) to IP addresses has been around since 1983. In 2000, this protocol got an additional popularity bump when Microsoft announced its intention to move away from its proprietary WINS protocol to DNS for name resolution. What's interesting about this pervasive service is how misunderstood it remains today by users and administrators alike. Eight years after Microsoft's intention to base all name resolution on DNS, many administrators still don't have a firm grasp on its internal workings.

The little protocol that could, it's likely that the misunderstandings about DNS stem from Windows administrator's focus on its server side. Microsoft's implementation of DNS is simple to install and configure, and its support of dynamic registration makes it one of those set-it-and-forget-it kinds of services on the network. But many of the interesting happenings of DNS occur at the client level. The process of DNS resolution and client configuration, whether be it on servers or workstations, has changed relatively little over Microsoft's past few OS updates.

In this month's column, let's take some time to unravel the complexities of DNS resolution at the client level. We'll ignore completely for now much of the configuration that occurs at the server level, instead focusing our attention on the process of resolution and the configuration settings you're seeing today on Windows XP and Vista workstations.

DNS queries actually occur in one of two flavors. The first, recursive queries occur when the DNS server being asked for resolution already has the answer to the question the client is asking within the DNS server's database or local cache. For recursive queries, the DNS server will respond to the requesting client with either the answer or an error message. The answer will always be the correct answer with a full response.

The other type of query is called an iterative query. In this case, the queried DNS server provides as much information as it has about the request. This information may come in the form of an IP address for another server that might have the answer to the query. It is within this query type that many Windows administrators

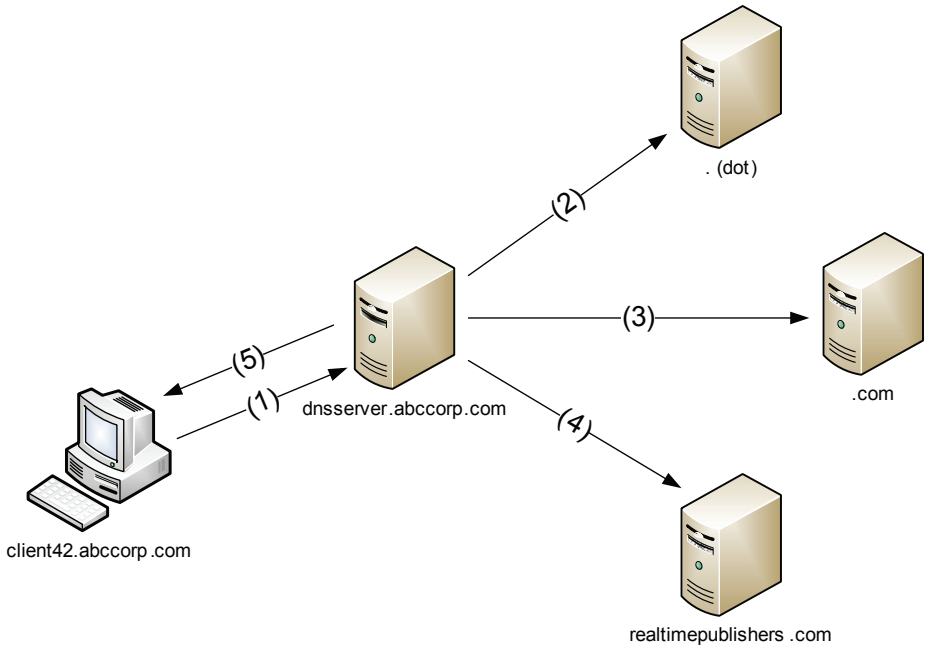


*Figure 1: The process to resolve a DNS address through an iterative query can involve numerous queries and multiple DNS servers.*

become confused about the overall process. To better explain this process, let's take a look at it in detail using Figure 1 as a guide.

In Figure 1, the desktop is currently configured with an FQDN of client42.abccorp.com. The desktop points to the server dnsserver.abccorp.com as its primary DNS server. The DNS suffix for both machines is abccorp.com. This suffix is important because it is used by the DNS server to help in identifying the record desired by the client. In this example, the client is attempting to resolve www.realtimepublishers.com. Perhaps the user on this desktop is interested in learning more about the great eBooks and other guides that can be found on this Web site. When the client begins the resolution process, the following conversations happen.

In Step 1, the client asks the server for the IP address for www.realtimepublishers.com. The DNS server first looks in its local cache to see whether a record exists. This record may exist because the DNS server had previously attempted a resolution. If the record exists in the DNS cache, the server responds immediately with the answer. In our case, the server does not have a copy of the response locally, so the process continues.

The DNS server then looks in its cache again, but this time after appending its DNS suffix to the query. In this case, the query will resemble www.realtimepublishers.com.abccorp.com. Why does this step occur? In the Windows world, administrators often make resolution requests to DNS servers using just a server's host name. This is similar to the situation when you enter nslookup exchangeserver rather than fully qualifying the name nslookup exchangeserver.abccorp.com. In order to make this process possible, Windows DNS is configured to append its primary suffix in any attempt at resolution. This response obviously will not be present in the local database or cache, so the process continues further.

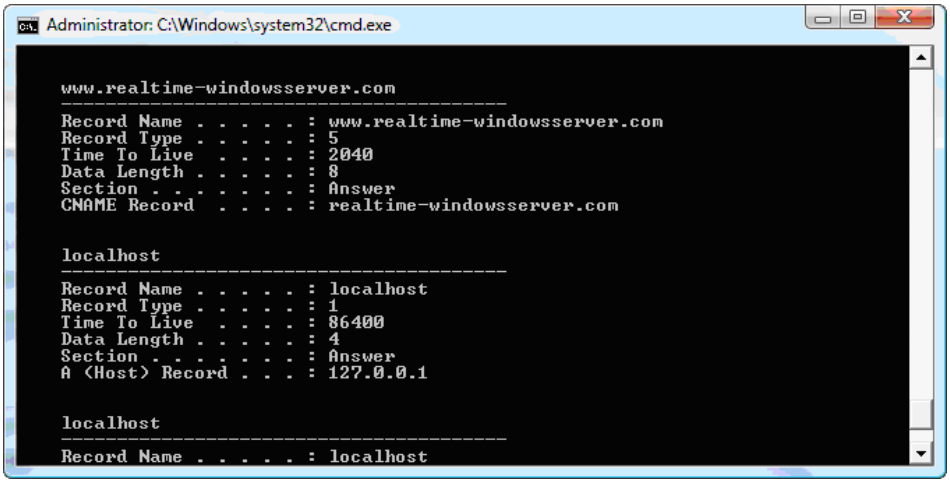


*Figure 2: Running ipconfig /displaydns on a client shows the client's local cache.*

For Step 2, once the server realizes that no combination of the client's request along with any permutations of its DNS suffix are locally available, the server next contacts the Internet root servers—. (dot) in the diagram. The DNS server asks the root server for the response to the entire query www.realtimepublishers.com. This query fails, as the Internet root servers are responsible for storing information not about our entire query but only for the answer to the rightmost element in our request—the .com portion. (dot) does not respond with an answer but instead a pointer to the DNS server that it believes to be authoritative for .com.

Our server then moves to Step 3. Armed with information about the location of .com, our local DNS server then queries this server with its entire query. The server authoritative for .com does not have a full and complete answer for the query. It does, however, have information about the server that is authoritative for realtimepublishers.com. This information is returned back to dnsserver.abccorp.com.

Now two-thirds of the way to completing the request, our server makes one final attempt. It knows the server that is authoritative for the realtimepublishers.com domain, but it needs to know the specific IP address for the server named "www." Getting ever closer to a response, our local DNS server moves to Step 4 and presents its entire query to the authoritative server for realtimepublishers.com. Unlike all the other servers, this server contains the full and complete answer and responds with the IP address for www.realtimepublishers.com.

With the IP address in hand, in Step 5, the local DNS server then returns the information back to the client. The client can then use this information for any of its local uses.

You can see that this process involves a lot of individual steps to go from initial query through to full completion. But you may be wondering, "If there are so many steps involved with this process, why do my DNS names resolve so quickly?" The strict hierarchy of DNS servers is necessary to support its extensibility, and the communication between the servers in that hierarchy can take some time, especially when there are multiple levels to the query.

DNS's "speed" comes from its local cache. Every time a DNS server completes a request, it stores a copy of that record in its local cache for a period of time (12 hours by default in Server 2008). When it receives a query for an address, it first checks the cache. Doing so gives it the ability to respond almost immediately for already-resolved addresses.

A cache also exists on the client, with individual records being stored there for 1 day (86,400 seconds) by default. It is possible to view the local client cache by using the command:

```
ipconfig /displaydns
```

The result looks similar to Figure 2, with each entry showing the record name and time to live in addition to other data.

For such a set-it-and-forget-it service, DNS still involves a lot of moving parts. As you can see here, even the simplest processes of DNS resolution involve the interworking of clients with numerous servers all across the Internet.

*Greg Shields, MCSE: Security, CCEA, is an independent author, speaker, and consultant, based in Denver, Colorado. With more than 10 years of experience in information technology, Greg has developed extensive experience in systems administration, engineering, and architecture. Greg is a contributing editor for both Redmond magazine and MCPmag.com, authoring two regular columns along with numerous feature articles, webcasts, and white papers. He is also the resident editor for Realtime Publishers' Windows Server Community at www.realtime-windowsserver.com. Greg is currently finishing his new book Windows 2008: What's New, What's Changed through SAPIEN Press.*

# Focal Point

# *Product Spotlight: Quest PowerGUI*

*by Don Jones*

Although Windows PowerShell is a powerful administrative tool with rapidly growing application in enterprises and smaller businesses alike, its adoption is hindered by one simple fact: It isn't a graphical user interface (GUI). Many less-experienced administrators have a tough time working with a command-line interface (CLI) because it offers few or no visual cues about what's possible or how to perform specific tasks. Even experienced administrators face a bit of a learning curve with PowerShell because they need to learn new commands and techniques in order to be effective.

Although eventually every administrator needs to learn to use Windows PowerShell natively, at the command-line, tools such as PowerGUI can provide a way of easing that transition. Provided as a completely free, no-registration-required download from www.PowerGUI.org, the application actually hosts the PowerShell engine underneath a straightforward GUI. For example, as Figure 1 shows, you can retrieve a list of processes simply by clicking the Processes node in PowerGUI's task tree view. Related actions are automatically exposed, such as stopping a process, changing its priority class, or even performing common tasks such as creating an XML, CSV, or HTML report.

One of the real values in PowerGUI, however, is the PowerShell Code tab, which—as Figure 2 shows—displays

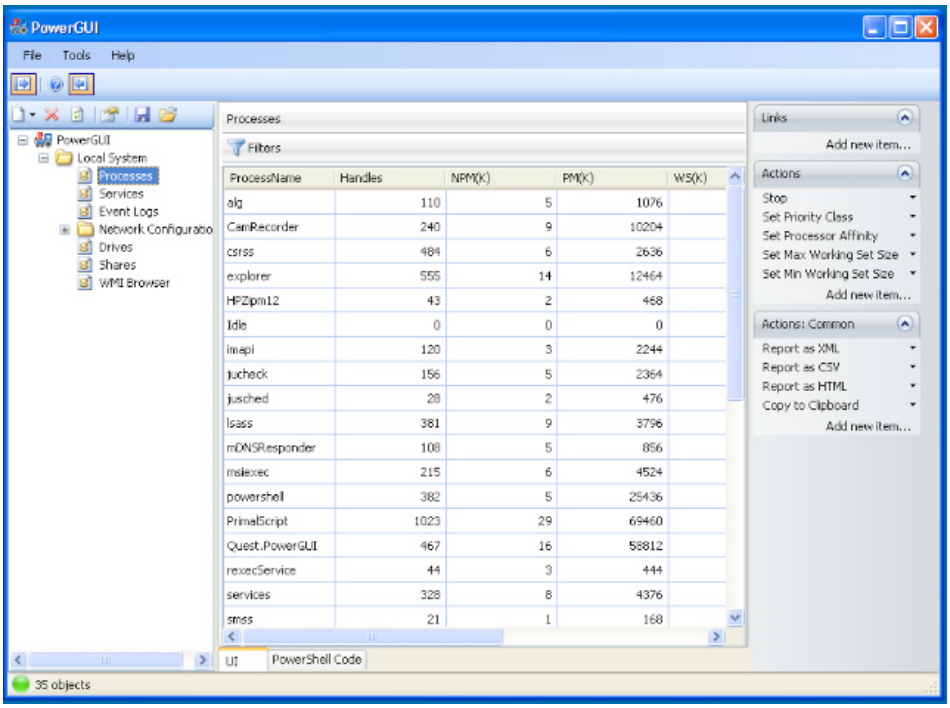


*Figure 1: Retrieving a list of processes through PowerGUI.*

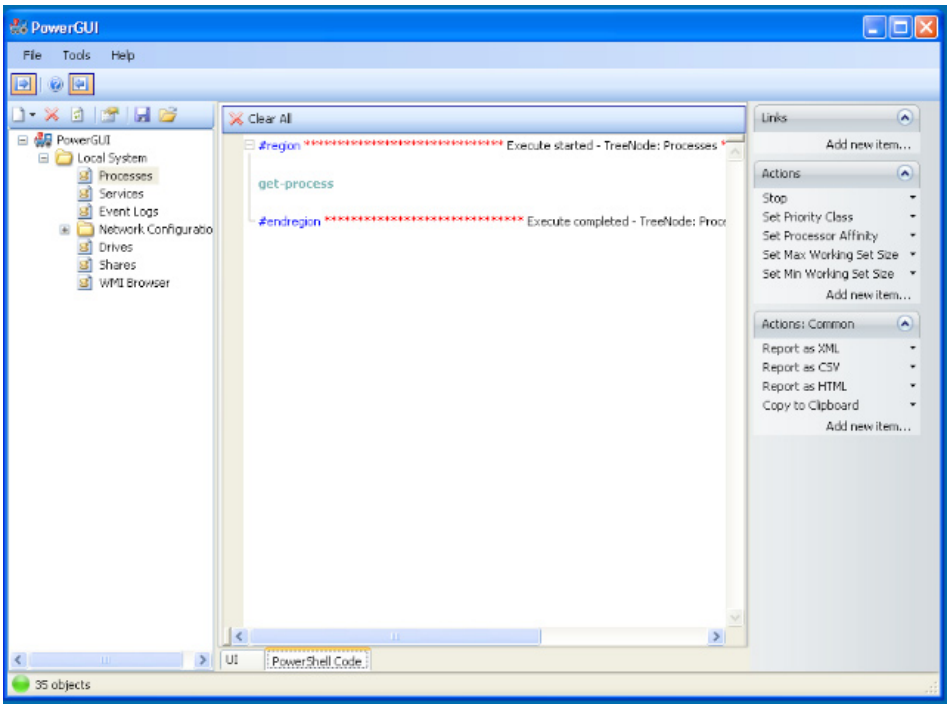


*Figure 2: Learning from the code created by PowerGUI.*

the actual PowerShell command line used to generate the graphical display.

This is a fantastic way to learn PowerShell: Allowing the GUI to generate the code for you, and then looking to see what code was generated. Of course, you have to remind yourself to do that—PowerGUI's designers clearly didn't intend PowerGUI to simply be a crutch, and so it doesn't shove the PowerShell code in your face. Personally, I'd prefer to see the code displayed in a pane underneath the graphical view so that it was more readily accessible; I truly believe that strong command-line skills are going to be a significant differentiator in job position and salary in a few years, and a tool that can help drive command-line familiarity is a great help.

PowerGUI isn't limited to simple tasks such as getting processes and event logs; it ships with PowerPacks for Active Directory (AD) management, Exchange Server 2007, and System Center Operations Manager 2007, giving it the ability to produce incredibly complex commands through a fairly simple-to-use GUI. For example, one incredibly useful—and quite complex—function is PowerGUI's WMI Browser (see Figure 3).

The code for this complex task, as Figure 4 shows, is pretty significant, and this is absolutely a case where the code—which is honestly something you'd never write yourself—is less important that the functionality of the tool. It's with this type of complex task that PowerGUI really shines, and you find more such tasks as you start adding PowerPacks to manage functionality beyond simple Windows elements.

PowerGUI also packs in a simple PowerShell script editor, providing syntax highlighting and basic code hinting. It's not as robust as commercial

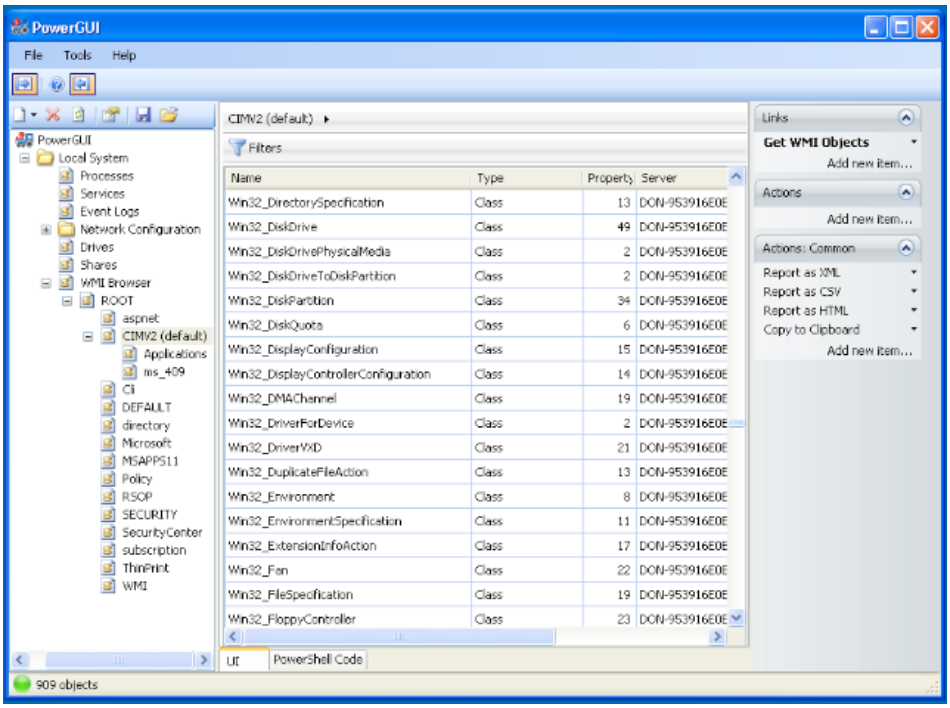


*Figure 3: PowerGUI's WMI Browser.*

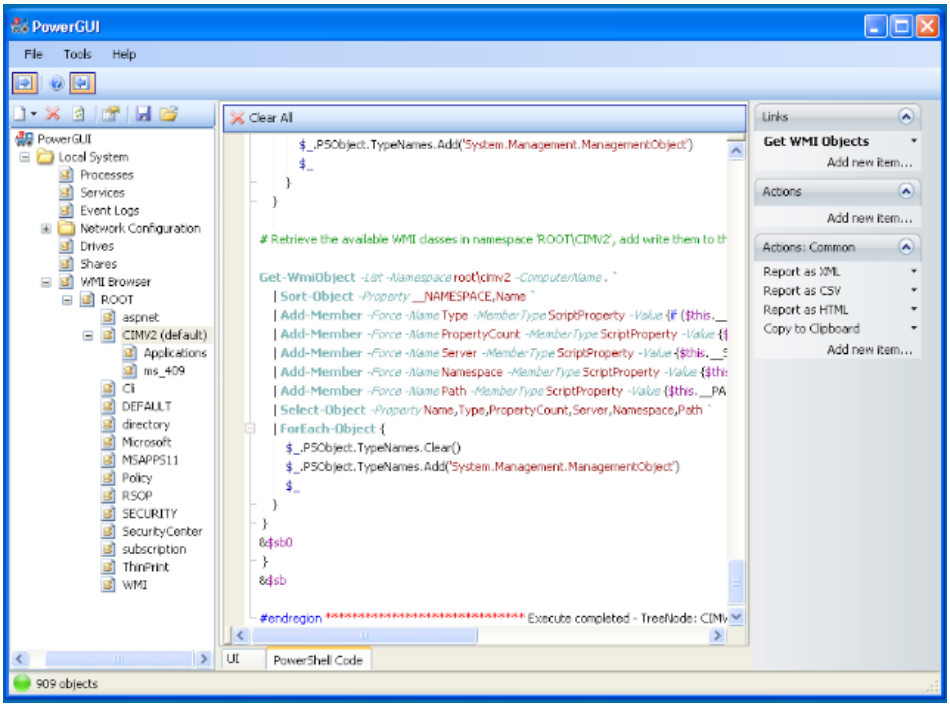


*Figure 4: PowerGUI offers powerful management of complex tasks.*

editors such as PrimalScript or PowerShell Plus, but it's far better than Notepad—and the price is certainly right. In fact, the editor provides features similar to the editor that will be bundled with PowerShell v2 once that is released.

All in all, PowerGUI is a great tool to have on your computer if you're working with Windows PowerShell. If nothing else, it's a nice way to learn more complex PowerShell commands, especially with Exchange, AD, and SCOM. It provides a nice set of extras, such as its WMI Browser and script editor. Once you move into the creation of more complicated scripts that automate complex business processes and include decision-making logic, you'll find yourself outgrowing

PowerGUI and thirsting for a more script-oriented workflow with Visual Studio-like features—all of which can be found in commercial script editors like those mentioned earlier. Until that time, however, PowerGUI is a great way to dip your toes into PowerShell and overcome any fear of the CLI.

*Don Jones is a Series Editor for Realtime Publishers, and the Director of Training and Publishing for SAPIEN Technologies. Visit him online at* www.ScriptingAnswers.com.

# Practical PowerShell

# *Local Password Management with PowerShell*

*by Jeffery Hicks*

The Problem: All your member servers have local administrator accounts that require periodic password changes. Or you might have a number of local user accounts on a member server that require management.

The PowerShell Solution: Because PowerShell supports Active Directory Services Interface (ADSI), managing local user accounts can easily be done from the console, although for more complicated tasks, you'll want to wrap the functionality into a PowerShell function. But don't panic, I've done the wrapping for you. You can download the code samples from the Realtime Publishers web site.

You don't need to install PowerShell on any of the servers or desktops that you'll be managing; everything I'm going to show you can be done from your Windows XP or Vista desktop. However, you'll need to make sure you are running your PowerShell session with credentials that have administrative permissions on those computers. One alternative is to establish a secure channel to the server or servers before you attempt any management tasks. A quick way to do so is to map a drive to the remote computer using alternate credentials such as this:

```
PS C:\> net use z: \\server12\c$ /user:mydomain\administrator
```

You'll be prompted for the password, but once this mapping is set up, any ADSI connection will use this secure channel and these credentials.

So how does all this work? Using the [ADSI] type adapter, you can "get" the local administrator account on any computer, again assuming you have appropriate privileges:

```
PS C:\> [adsi]$admin="WinNT://$env:computername/administrator"
```

This expression creates an ADSI variable, $admin, for the administrator account on the specified computer. In this situation, the computer is whatever is returned from

```
$env:computername
```

which will be your local computername. You can substitute any computername on your network. If you've renamed the administrator account, simply use the appropriate name. The ADSI provider I'm using is the Windows NT provider (WinNT) and the name is case sensitive. If all went well, you should be able to try:

```
PS C:\> $admin | Select *

UserFlags                  : {66049}
MaxStorage                 : {-1}
PasswordAge                : {10554947}
PasswordExpired            : {0}
LoginHours                 : {255 255 255 255 255 255 255 255 25…
FullName                   : {}
Description                : {Built-in administration account
BadPasswordAttempts        : {0}
LastLogin                  : {3/16/2008 6:10:29 PM}
HomeDirectory              : {}
LoginScript                : {}
Profile                    : {}
HomeDirDrive               : {}
Parameters                 : {}
PrimaryGroupID             : {513}
Name                       : {Administrator}
MinPasswordLength          : {0}
MaxPasswordAge             : {3628800}
MinPasswordAge             : {0}
PasswordHistoryLength      : {0}
AutoUnlockInterval         : {1800}
LockoutObservationInterval : {1800}
MaxBadPasswordsAllowed     : {0}
objectSid                  : {1 5 0 0 0 0 0 5 21 0 0 0 32 71 2…
```

See the PasswordAge property? The value you see is the password age in seconds. To get the value in days, divide this number by 86400:

```
PS C:\> ($admin.passwordage)[0]/86400
122.163738425926
PS C:\>
```

The [ADSI] type adapter makes a best effort for creating the underlying .NET System.DirectoryServices.DirectoryEntry class but isn't perfect. Thus, I've had to fuss a little to get the object's PasswordAge value. As you can see, the administrator password was set a little more than 122 days ago.

That's all there is to the basic functionality. But because this column is called Practical PowerShell, let's make it so. Take a look at the Get-PasswordAge function, which Listing 1 shows.

You can download the code from the Realtime Web site at http://www.realtime-windowsserver.com/code/v1n4_Practical_PowerShell.zip

```
Function Get-PasswordAge {
    Param([string]$computer=$env:computername,
    [string]$account="Administrator"
    )

    $errorActionPreference="SilentlyContinue"

    [adsi]$user="WinNT://$computer/$account,user"

    if ($user.name) {
    [int]$age=($user.passwordAge[0])/86400
    [datetime]$lastchanged=(Get-Date).addDays(-$age)

    $obj=New-Object System.Object

    $obj | Add-Member -MemberType NoteProperty -Name "Computer" `
    -Value $computer.ToUpper()
    $obj | Add-Member -MemberType NoteProperty -Name "Account" `
    -Value $account.ToUpper()
    $obj | Add-Member -MemberType NoteProperty -Name "PasswordAge" `
    -Value $age
    $obj | Add-Member -MemberType NoteProperty -Name "LastChanged" `
    -Value $lastchanged

    write $obj

    } else {
        $msg="Failed to connect to "+$computer.toUpper()
        Write-Warning $msg
        }

}
```

*Listing 1: The Get-PasswordAge function.*

To use this function, you'll need to add it to your PowerShell profile, copy and paste it into your current PowerShell session, or dot source the script file you downloaded.

The function takes two parameters: a computer name and account name. I've set the defaults to the local computername and administrator, respectively. To use locally, all you need to do is run the function:

```
PS C:\> get-passwordage

Computer    Account         PasswordAge    LastChanged
--------    -------          ---------      -----------
PUCK        ADMINISTRATOR          122      11/15/2007 6:39:01 PM


PS C:\>
```

Because the function writes an object to the pipeline (which I'll cover in a moment), you can also execute expressions like this:

```
PS C:\> (get-passwordage).passwordage
122
PS C:\>
```

Here's how the function operates: the first thing I do is set the $errorActionPreference variable to SilentlyContinue. If there is an error, such as inadequate permissions or the server is unavailable, I want the function to keep going because it has its own error handling. By setting this variable to SilentlyContinue, I'm turning off any error messages to the error pipeline.

Next, the function creates the ADSI object like the one we looked at earlier based on the parameters passed to the function:

```
[adsi]$user="WinNT://$computer/$account,user"
```

If the expression was successful, $user will exist, so I can use an If statement to check:

```
if ($user.name) {
```

If there is no value for $user.name, the function jumps to the Else statement, which displays an error message and exits:

```
} else {
   $msg="Failed to connect to "+$computer.toUpper()
   Write-Warning $msg
   }
```

I use the Write-Warning cmdlet to send the message to the warning pipeline, which for me, is just as good as an error. If I had used Write, the message would have gone to the success pipeline, which wouldn't be good if I was going to pipe the function's output to something like Out-Printer.

But let's assume I made a connection. The function defines a variable $age:

```
[int]$age=($user.passwordAge[0])/86400
```

This variable holds the password age in days as an integer. By casting $age as an integer, a value such as 122.163738425926 is saved as 122. I use this value to set the next variable, which is the date the password was last changed:

```
[datetime]$lastchanged=(Get-Date).addDays(-$age)
```

I use the [datetime] object's addDays() method and add the $age variable as a negative number, in essence subtracting that number of days from the current date and time.

Because I want the function to write to the pipeline, I use the New-Object cmdlet and create an empty generic object:

```
$obj=New-Object System.Object
```

I'll give my object properties by piping it to Add-Member, which creates a property with the name and value I specify.

```
$obj | Add-Member -MemberType NoteProperty -Name "Computer" `
Value $computer.ToUpper()
$obj | Add-Member -MemberType NoteProperty -Name "Account" `
-Value $account.ToUpper()
$obj | Add-Member -MemberType NoteProperty -Name "PasswordAge" `
-Value $age
$obj | Add-Member -MemberType NoteProperty -Name "LastChanged" `
-Value $lastchanged
```

The last thing the function has to do is write the object to the pipeline:

```
write $obj
```

That's all fine and dandy, but what about all the servers I want to manage? Well, you could use an expression like this if you have only a few servers:

```
PS C: \> "server1","server2","server3" | foreach { get-passwordage $_ }
```

Remember that PowerShell treats any comma separated list as an array, so each computername is passed one at a time to the ForEach construct, which calls the Get-PasswordAge function passing it the current pipeline object as the computername. Assuming you can connect to all the servers, you'll get a simple table. If you have a text list, you can pipe its contents to the function:

```
PS C:\> Get-Content servers.txt | foreach {get-passwordage $_}
```

But don't forget the function is writing to the pipeline, so you can use expressions like these:

```
PS C:\> Get-Content servers.txt | foreach {get-passwordage $_} |
 sort PasswordAge -desc
PS C:\> Get-Content servers.txt | foreach {get-passwordage $_} | where {$_.passwordage
-ge 60 } convertto-html | out-file pwdreport.html
```

I'm sure you can come up with other ways to use this function. By the way, if you've changed the administrator account name, simple specify the name:

```
PS C: \> get-content servers.txt | foreach {get-passwordage $_ zippy}
```

### *Changing Passwords*

Now that you have a handle on where you need to change passwords, let's use PowerShell to get the job done. Again, we're going to use ADSI:

```
PS C:\> $admin.SetPassword("P@$$w0rd")
```

The new password is set immediately. If you refresh the object and check the password age again, it should be only a few seconds old:

```
PS C:\> $admin.psbase.refreshcache()
PS C:\> $admin.passwordage
18
PS C:\>
```

Again, let's wrap up everything into a function, which Listing 2 shows.

```
Function Set-Password {
 Param([string]$computer=$env:computername,
         [string]$account="Administrator",
         [string]$password="P@ssw0rd"
       )

    $errorActionPreference="silentlyContinue"

    [adsi]$user="WinNT://$computer/$account,user"
    #get current password age
    [int]$oldage=$user.passwordage[0]

    if ($user.name) {
        $user.SetPassword($password)
        sleep -milliseconds 500
        }
    else {
        $msg="Failed to get $account on "+$computer.toUpper()
        Write-Warning $msg
        return
    }

    #refresh object and check to make sure pass
    #word was changed
    $user.psbase.refreshcache()

    if ($user.passwordage -gt $oldage) {
        $msg="Failed to change password for $account on "+$computer.toUpper()
        Write-Warning $msg
    }

}
```

*Listing 2: The Set-Password function.*

This function requires parameters for the computer, account name, and the new password to set. I've set a default new password, but you can change it. The function takes essentially the same approach as the Get-PasswordAge function. I turn off the error pipeline so that I can display my own messages. Again, I use ADSI to get the user object:

```
[adsi]$user="WinNT://$computer/$account,user"
```

The function grabs the user's current password age, so we can check later to verify the change.

If the user object doesn't exist or there was some other error connecting to the server, $user.name won't have a value, so the function displays a message to the warning pipeline and exits:

```
$msg="Failed to get $account on "+$computer.toUpper()
Write-Warning $msg
return
```

Otherwise, the SetPassword() method is called. I have the function sleeping for 500 milliseconds to give the remote server a chance to update the account. You might need to adjust this based on your network and server conditions.
To verify the password change was successful, the function refreshes the object:

```
$user.psbase.refreshcache()
```

If the password was changed, the passwordage property will only be a few seconds.

To modify the password on a list of computers, we can use the same approach we employed to retrieve the password age:

```
PS C:\> Get-Content servers.txt | foreach {set-password $_}
```

An expression like this will use whatever default values you've specified in the function for the administrator account name and new password. If you prefer to specify a new password value, use this expression:

```
PS C:\> Get-Content servers.txt | foreach {set-password $_ "MyP@$$w3rd"}
```

It doesn't matter whether servers.txt has 5, 50, or 500 servers, the local administrator password will be changed on all of them and with practically no effort on your part. OK, you had to download and get the function loaded, but once you understand this process, implementing it is extremely simple.

I've run out of time for now, but next month, I'll continue our discussion with additional PowerShell expressions for managing all your local accounts.

*Jeffery Hicks, MCSE, MCSA, MCT, and Microsoft PowerShell MVP, is a Scripting Guru for SAPIEN Technologies. Jeff is a 16-year IT veteran. He has co-authored and authored several books, courseware, and training videos on administrative scripting and automation. His latest book is WSH and VBScript Core: TFM (SAPIEN Press 2007). You can contact him at jhicks@sapien.com.*

# Exclusively Exchange

# *Upgrading to Exchange 2007*

*by J. Peter Bruzzese*

Let's be honest—there is no real *upgrade* to Exchange 2007. Not in the traditional sense of the word. You cannot take a server running any prior version of Exchange and insert a DVD and upgrade that server in-place. What you can do is transition or migrate your existing organization and server structure over to Exchange 2007. Why the semantics? And, more important, what does the move to Exchange 2007 entail? Let's find out!

### *Migrations and Transitions*

The reason you can't upgrade directly is that Exchange 2007 can be installed only on x64 hardware and off a 64-bit OS. Being that such isn't the case for Exchange 2003/2000, you must install Exchange 2007 on a separate machine (running Windows Server 2003 or 2008); you can then concern yourself with transitioning your organization.

You'll note the absence of any mention of Exchange 5.5 in the previous paragraph. This is because a transition is not possible with Exchange 5.5 or older, nor is it possible with other non-Microsoft messaging infrastructures (that is, Lotus Notes). In these cases, you are looking at migrating over. A migration involves a change in the messaging infrastructure in which the data itself is migrated without configuration data being carried over and coexistence is not supported. Keep this in mind with Exchange 5.5 as well: Coexistence is not supported, so you must remove all Exchange 5.5 Servers before even considering the deployment of your first Exchange 2007 Server within your environment. To migrate Exchange 5.5 to 2007 actually requires you to move your organization to an Exchange 2000/2003 environment and then transition to Exchange 2007.

> Microsoft provides transition tools called the Microsoft Transporter Suite for migrating from Lotus Domino. The tools will help you move over Directory information, Free/Busy data, users, groups, mailboxes, and more. To download the suite, go to http://www.microsoft.com/downloads/details.aspx?FamilyId=35FC4205-792B-4306-8E4B-0DE9CCE72172&displaylang=en.

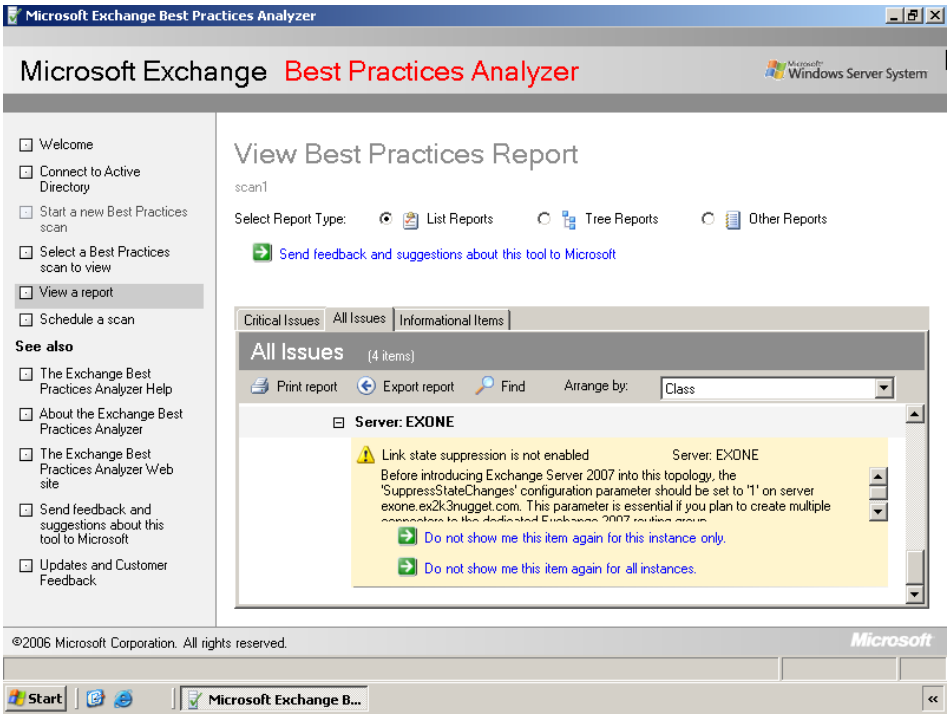


*Figure 1: Working with the Exchange Best Practices Analyzer.*

The logical question likely on your mind at this point is "How do I perform a transition?" Well, again, being that an in-place upgrade to Exchange 2007 is not a possibility, a transition that includes some form of coexistence is required. Coexistence implies a period of time (with length of time varying according to your needs) during which you have both Exchange 2000/2003 Servers running alongside Exchange 2007 Servers. So where do you begin the transition/coexistence process?

## Preparing For Exchange 2007

Preparation for that first Exchange Server involves eliminating all Exchange 5.5 servers from the environment. Next, make sure that your Exchange 2000 and 2003 systems have the correct service packs applied (Exchange 2000 systems require SP3 and Exchange 2003 systems require SP2). To prepare your Active Directory (AD) for the inclusion of the new Exchange environment, you will need to enhance the schema. Before doing so, ensure that the schema master domain controller is running Windows 2003 with SP1 or later. In addition, you should locate Global Catalog (GC) servers within the site that will run the Exchange Server and ensure these are also running Windows 2003 Server with SP1 or later. Next, there are two levels you need to raise to the proper setting—the Domain Functional Level and the Exchange Organization level. Make sure both are operating in Native Mode.

> The Exchange Best Practices Analyzer is a useful tool to ensure that your environment is ready for the deployment of your first Exchange 2007 server. This tool, available at http://www.exbpa.com/, will scan your directory and configuration environment to ensure you are ready for the transition. As Figure 1 shows, the tool will highlight any errors that you need to address pre-deployment.

Once you have all service packs installed and the levels are set, you are ready to prepare AD. You can do so using switches and the setup program for Exchange 2007. There are switches you can run one at a time for the smallest impact on your environment with each step (including the /PrepareLegacyExchangePermissions and /PrepareSchema switches). However, simply running the /PrepareAD switch with the setup program will make all the necessary changes (so that it is not necessary to run the other two switches). Your next step is to run the /PrepareDomain or /PrepareAllDomains switches to configure permissions.

## Installing and Moving Data

Front-end servers under Exchange 2000 and 2003 cannot access an Exchange 2007 Server, you need to replace those front-end servers with an Exchange 2007 Client Access Server (CAS) role. This server should be your first server if you are going to break up the installation of Exchange onto multiple systems where your Mailbox Server is separate from other roles. The CAS should be installed first and the Hub Transport second (because a Mailbox Server cannot send or receive email without the CAS and Hub Transport in place). Once these are in place and you ensure the front-end servers have been dropped, you can install the Mailbox Server role.

> You could simply perform a typical Exchange 2007 Server install where all three major roles are installed in one shot.

The moment you have an Exchange 2007 Server in your environment, you have entered the state of coexistence, which can be as short or long as you need. To begin the process of transition, you need to use the Move Mailbox wizard or the Move-Mailbox cmdlet (both from the Exchange 2007 system). If you try to use the System

Manager tools in Exchange 2003/2000 to perform these tasks, it will appear to begin the process but will fail.

In addition to moving mailboxes, there are additional items you need to prepare for and consider transitioning before you can remove the last vestiges of Exchange 2000/2003 from your organization. Public Folders will need to be factored into the transition equation. In addition, you must make sure that your clients can access their mailboxes from their MAPI or POP/SMTP clients and/or Outlook Web Access (OWA) or Outlook Anywhere, if you use these features in your environment. You should ensure the Offline Address Book and the Schedule+ Free Busy information has been replicated over to your Exchange environment as well. There is much to consider.

## Conclusion

Once the dust settles, and all the pieces are in place, the time for coexistence is over. You can begin the decommissioning phase to the 'upgrade.' This process requires a great deal of thought and preparation, but when you consider the great features that come with Exchange 2007, you won't be disappointed you made the move.

*J. Peter Bruzzese is an MCSE (NT,2K,2K3)/MCT, and MCITP: Enterprise Messaging Administrator. His expertise is in messaging through Exchange and Outlook. J.P.B. is the Series Instructor for Exchange 2007 for CBT Nuggets. His latest book is "Tricks of the Vista Masters". He is co-founder of ClipTraining.com, a provider of short, educational screencasts on Exchange, Windows Server, Vista and Office 2007. You can reach Peter at jpb@cliptraining.com.*