Realtime publishers

November 2008
Volume 1, Number 10

Windows Administration

in Realtime

Letter from the Editor

When Is "Fix on Fail" a Good Idea?

by Greg Shields

A recent computer problem with the United States' air traffic control system brought about a one-day virtual standstill to flights all around the country. The problem, which started in one location and eventually progressed through multiple major U.S. airports, caused many functions of air traffic control to cease functioning properly. From news reports, the IT teams in charge of the system were aware of the impending failure but were unable to proactively resolve the problem due to the department's "fix on fail" policy. Essentially, the technician's good intentions were stunted by bad policy.

Now, I personally am not a fan of the drive towards what some call "small government." I believe that effective industry regulation is a necessary check on the otherwise unrestricted power of corporations. I also believe that while government doesn't always spend money in the most efficient of ways, the institution of government is the organization best suited to tackle the humanity-evolving projects that require big dollars and no eye towards profit.

But at the same time, no one has ever said that government is good at acting *proactively*. If IT technicians were aware a problem was about to occur, then not fixing it is little more than a failure of due diligence.

The same holds true in our IT organizations, but with a slightly different bent. As I travel around the country consulting with various groups, I most often find that simple monitoring is a glaring omission. Most especially in the small shops, often the only way IT finds out about a problem is when the Help desk phone rings. With the price of effective monitoring products dropping into the range of…well…***free*** these days, this omission is another example of a failure of due diligence.

So if you've got some spare server capacity but an empty pocketbook, why not take a look through some of the zero-cost monitoring options currently available on the market today. *The Dude* is a great network-based monitoring platform that alerts when devices go down. *EventSentry Light* and *Spiceworks* are both limited but cost-free tools that alert your pager when system conditions occur. Other tools are out there—some as individual monitoring platforms and others that come as a component of a full systems management suite—that aren't free but are easy on the pocketbook. For any, the very first time you're alerted before your Exchange private store fills up, preventing another 8-to-12 hour outage, you'll feel you've paid back that initial cost.

No matter what you pick, in this editor's opinion "Fix on Fail" is never a good idea. It's little more than a shortsighted way to not do work today in the hopes that it won't break tomorrow.

Answers from the Experts

What is a good Secure FTP server that is Win2008 compatible? What's the difference between Small Business Server and the new Essentials Business Server? And more...

by Don Jones

This month, I'll be answering several shorter questions that have come in over the past few weeks.

Q: What is a good Secure FTP server that is Win2008 compatible?

A: Surprisingly, the one in the box—Microsoft has finally included one in Win2008's version of IIS, after taking so long to do so most of us probably assumed Microsoft didn't know about Secure FTP! Install it by using Server Manager; be sure to drill into the IIS details to find the SFTP option.

Q: I need to import users into AD from a CSV file that my HR department provides. What's the easiest way?

A: Easiest—probably PowerShell, and it'll give you an opportunity to use the shell and start getting used to it. Start by installing PowerShell (it's free), then getting the AD command set ("cmdlets"—also free). You'll need to know the column names from your CSV, and the corresponding AD attribute names. Then do something like this in the shell:

Import-CSV *filename* | ForEach-Object { New-QADUser –organizationalUnit *destination_ou* –samAccountName $_.*'column'* –department $_.*'column'* –fullName *'column'* –city *'column'* }

Each instance of column should be the column name from your CSV file that corresponds to the attribute; you can include as many AD attributes as you want using the pattern:

-attribute 'column'

If your CSV file layout is constant (for example, you always get it with the same column names), save this command in a text file that has a .ps1 filename extension. You can then re-run the command however often you need to in order to import new CSV files.

Q: I can't get IIS to run ActivePerl (or PHP, or any other add-in extension). What's wrong?

A: First, make sure you properly installed the extension and enabled it in the IIS management console—new extensions are typically disabled by default for security. Keep in mind that each Web site's worker processes run in an application pool, and the processes themselves use the pool's configured identity. By default, application pools use the NetworkService account, although you can change it. However, the pool identity needs two important user rights that normal domain users don't have: Replace a process-level token, and Adjust memory quotas for a process. If the pool identity doesn't have those privileges, it won't work properly for many CGI-based extensions. Ideally, you should use ISAPI-style extensions whenever possible anyway.

Q: What's the difference between Small Business Server and the new Essentials Business Server?

A: I hate Microsoft product names sometimes: "Small" I can understand; "Essentials?" Basically, EBS is Microsoft's "Medium-Sized Business Server." It contains most of the same functionality as SBS, but supports as many as 300 users and allows specific functionality to be spread across different servers rather than consolidated onto a single server as with SBS. You can, for example, have a "database server" that is separate from the machine acting as domain controller. EBS costs quite a bit more per server, and per user, than SBS (although the per-user cost for the high-end Premium edition is only slightly higher than SBS).

Microsoft has upgrade paths and pricing from editions of SBS to editions of EBS, meaning your business can grow from being "small" to "medium" (or "essential," I guess) over time. Frankly, 300 users doesn't say "medium" in my book, so maybe the product name should be "Bigger-Than-Little-But-Still-Smallish Business Server." Okay, "Essentials" is probably better.

Q: How much memory does Win2008 Server Core really need?

A: Server Core won't install with less than 512MB installed, although if you're installing it into a virtual machine, you can often reduce the memory after installation. There are reports of 256MB causing no problems without any roles installed (suggesting that simple roles such as DHCP Server might work fine with that, but maybe not a busy Active Directory—AD), although 128MB made it stop working; another reports lowering the memory as low as 64MB with good results. One report lists Server Core humming happily with 256MB running AD, DNS, and DFS installed. Your mileage may vary.

Do you have an IT question you'd like Don to answer? Send it to answers@realtimepublishers.com for consideration!

Don Jones is a co-founder of Concentrated Technology (www.concentratedtech.com), helping to deliver IT knowledge in less time using innovative content techniques. He also serves as CTO and Series Editor for Realtime Publishers. Don is the author of more than 30 IT books, including Windows PowerShell: TFM; VBScript, WMI, and ADSI Unleashed; Managing Windows with VBScript and WMI; and many more. He is a multiple-year recipient of Microsoft's "Most Valuable Professional" (MVP) Award with a specialization in Windows PowerShell.

Product Review

Hyena from System Tools

by Don Jones

If you haven't discovered Hyena, it's possible you're working harder than you need to for Windows administration. Hyena is an all-in-one tool for AD administration, server management, service management, event management, job and task scheduling, printer management, disk management, and even reporting. It's more than a simple Microsoft Management Console (MMC) loaded with snap-ins; each element of Hyena was custom-designed and, in many cases, combines functionality from multiple native MMC snap-ins.

Active Directory (AD) functionality includes full user and group management, including the ability to copy existing objects when creating new ones and have just selected attributes copied. You can even create named templates to speed object creation, and can rearrange AD queries on menus to speed access. Figure 1 shows Hyena's AD management view.

Service management is also made easier: You can access services from any computer, view service dependencies, install or remove services on multiple computers, reconfigure services—on multiple computers at once!—and even remotely view device drivers. This functionality is notably missing in Microsoft's native tools and can enable you to easily, finally, meet password-management policies for service accounts in your environment.

Full support for managing the local users and groups of your domain members is also provided, including modification of multiple users at once, password reset handling, automatic cleanup of user home and profile directories upon user deletion, and more. You can centrally manage task schedules throughout the environment, including viewing and creating tasks on multiple computers at once.

Hyena includes Access-based reporting (Microsoft Access is required), and can copy most selected information to the Windows Clipboard for use in other applications. Hyena integrates with the company's Exporter Pro application, which supports the creation of delimited text files for computers, shares, users, groups, printers, security information, registry keys, Windows Management Instrumentation (WMI) information, and much more. It can even consolidate information from multiple domain controllers, such as the pre-Win2003 "last logon" information, which isn't replicated between domain controllers. Exporter Pro is actually included with Hyena, and is sold separately (handy if you have someone in your environment who can be tasked with reporting but doesn't need Hyena's management capabilities).

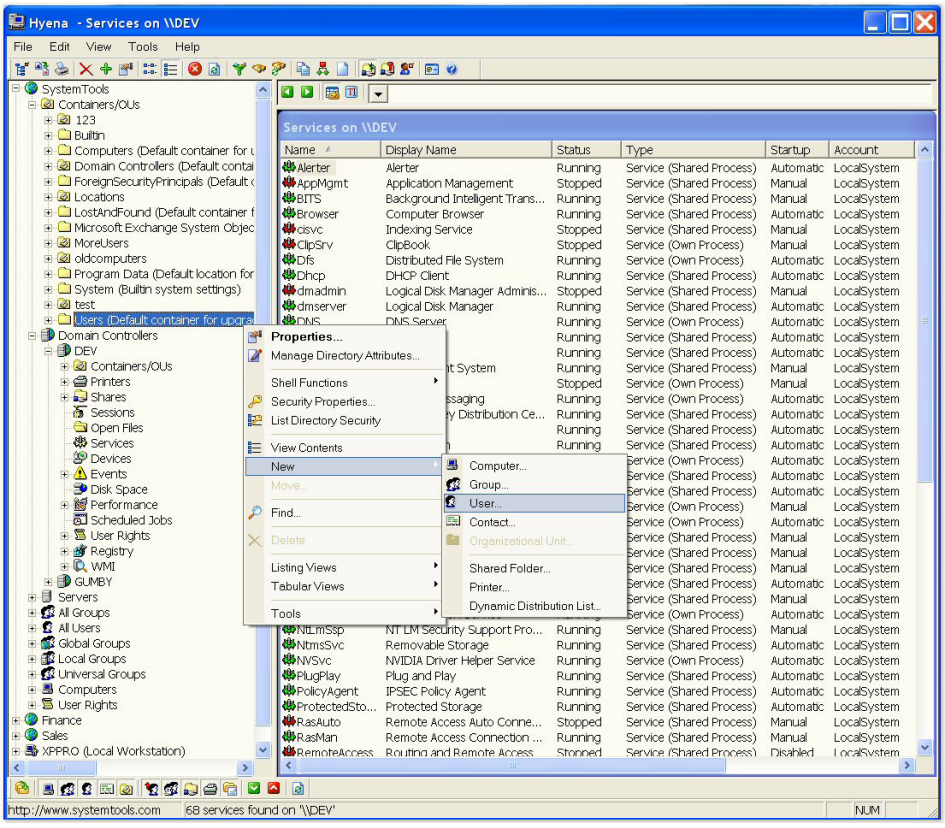

Figure 1: Hyena's AD management view.

User importing is not included; the company's User Commander product handles this (including import templates with no scripting required), but is a separate purchase.

Event log management includes the ability to view one or more event logs—filtering events by several criteria, if desired—from multiple computers, all in a single view. This doesn't use any form of log consolidation; rather, Hyena directly retrieves the information from the desired computers, which can be time consuming depending on how many you specify and where they're physically located. Everything in Hyena is agentless, meaning the workstation running Hyena connects directly to managed resources in order to obtain information. The benefit of this approach is that there is no need to deploy and maintain agents; the downside is that it can sometimes take a while for Hyena to collect the information you need—especially in the case of collecting WMI information.

Numerous conveniences are included: You can supply alternative credentials for connections to any domain to which you've pointed Hyena, eliminating the need to use RunAs or to maintain a synchronized set of credentials across domains; you can save event log filtering options for frequently used filters; you can clone printer configuration across computers (useful when you're migrating printers from one server to another); and much more.

The product's Enterprise edition also supports direct connectivity to Exchange Server for mailbox management, access to Windows Terminal Server user settings, and integration with WMI for gathering inventory information and, in some cases, executing remote configuration changes. A single license for the Enterprise edition runs $269. You can also find a 30-day trial available at www.systemtools.com.

The Top-5 Reasons to Switch to Hyper-V

Should You Switch to Virtualization?

by Steve Giovanni

Now that Microsoft has released Hyper-V to the public, many are questioning whether to make the switch to virtualization. This topic is certainly worthy of investigation because Microsoft has done an excellent job in providing a fast, stable product with near feature parity with VMware. Designed to aid in your decision-making process, the following list will provide you with the top-5 reasons to make Hyper-V your virtualization solution of choice.

1. Cost

This one is a no-brainer. Microsoft is basically giving their product away while VMware charges several thousand dollars for their solution. But what of the oft-said phrase "You get what you pay for?" Well, don't be too quick to judge. During my testing on identical hardware, Hyper-V's performance more than sufficiently matched that of ESX. Formatting a hard drive I had created inside of ESX took about 5 seconds, and in Hyper-V, it took about 30 seconds.

For example, take the topic of server migration. Virtualization provides much greater flexibility for managing patches and performing system maintenance. If you need to add RAM to a physical server hosting several virtual machines (VMs), you have the opportunity to move the virtual guests onto another physical host while you shut down the server to add the memory, then move them back when the maintenance is complete.

Some will say that for your extra dollars, you are buying the maturity that comes with a product that has been on the market for years, which is certainly true. With VMware, you have VMotion, a tried-and-true method that can literally move a VM while never dropping a single ping packet. With Microsoft's Quick Migration, you have a newer solution that does not achieve the same level of functionality. Because of this, your VMs become unavailable during the migration process. The amount of downtime will vary depending upon the size of your VM and the specifications of your hardware, but on average, the migration takes between 8 seconds to 2 minutes.

On the one hand, 8 seconds really isn't that long, relatively speaking. If you have a scheduled maintenance window once a month and perform the migration during this time, that downtime works out to be merely 1.6 minutes a year. On the other hand, 8 seconds might be 8 seconds too long for mission-critical servers.

With VMotion, the VM configuration is created on the destination side. A memory map is created first on the source side, then transferred to the destination side. This process actually happens multiple times during the transition, until finally the destination side takes control of the VM. Quick Migration fully suspends a VM, copies its memory image to disk, and then reloads and resumes the VM on a new host. To sum it up, ESX's feature set may be more mature, but Hyper-V's features are adequate for most environments.

2. Guest Licensing

Cost is worth a second look in this case due to the money you can save in licensing. It is important to understand your licensing rights and obligations when running Microsoft Server and other Microsoft applications in a virtual environment. Before virtualization, each software license you purchased from Microsoft allowed you to install and use one copy of the software on a server. You needed a license for each installation of the software, regardless of whether it was running.

With Microsoft's updated licensing plan, however, the use rights no longer specify the number of times software may be installed and used on a server. Although you can actually have an unlimited number of non-running instances, each license gives you the right to have a certain number of concurrently running instances of the software on a particular server at a time.

For example, each license for Exchange grants you the right to create however many installations of Exchange you want, but the right to run one instance of Exchange at any given time. The server chosen as the 'licensed' server, however, may be run in either a virtual or physical environment.

To help you take advantage of the deployment flexibility that VM technology offers, Microsoft has become more liberal with Server 2008 licensing. Windows Server 2008 is available in eight versions, three of which include Hyper-V:

- Windows Server 2008 Standard—With Windows Server 2008 Standard Edition, customers can run one physical and one virtual instance per license—a 1:1 physical-to-virtual ratio.
- Windows Server 2008 Enterprise—With Windows Server 2008 Enterprise Edition, customers can run one physical and four virtual instances per license—a 4:1 physical-to-virtual ratio.
- Windows Server 2008 Datacenter— With Windows Server 2008 Datacenter Edition, customers receive unlimited virtual instances per license—a 1:U physical-to-virtual ratio.

Additionally, if you're a Microsoft Certified or Gold Certified Partner, you are entitled to at least four copies of Windows Server 2008 Datacenter Edition for internal-use purposes. This means that many companies have the opportunity to run their entire internal infrastructure on Datacenter machines and enjoy free licensing for all the Windows guest operating systems (OSs). This could apply to server types including domain controller, DNS, email, print, patch, monitoring, remote access, and many more.

3. Availability

Although serving many of the same functions, these two products are in many ways aimed at different markets. Microsoft's solution is designed for the majority, and ESX will soon begin to become more of a niche solution for customers who need the features lacking in Hyper-V.

There's no question that VMware is market leader today, but by bundling Hyper-V with Server 2008, Microsoft's market share will increase exponentially because now most everyone who purchases a copy of Server 2008 is going to at least have the potential to use Hyper-V. Thus, with the increased number of people that have access to Hyper-V, you can expect to see more and more features in Hyper-V, third-party add-ons for Hyper-V, and widespread use and documentation of Hyper-V.

A good analogy for this situation is Terminal Services versus Citrix Presentation Server. Terminal Services is adequate for most scenarios and is bundled with Windows already, so more people use it. After all, why pay for the features you don't need? If you need the more advanced features that Terminal Services doesn't yet offer, you might decide to go with Citrix's solution. Likewise, Hyper-V aims to satisfy the needs of most customers, and stands to be more prevalent due to the large existing install base of Windows Server.

Microsoft's strategy is a good one, and a proven one to boot. They're giving their product away. They're bundling it with Server 2008 and therefore it will become more widely used, regardless of whether it is better or worse, simply because it is more ubiquitous. It's not unlike when they gave Windows away in order to get their foot in the door, and now as a result they dominate the market.

Microsoft has worked very hard to develop a broad feature set for Hyper-V and to integrate this form of virtualization technology with its other offerings. For example, to get higher-level functionalities, you would use Microsoft's System Center Virtual Machine Manager (VMM), one of the offerings in the System Center suite of products. It's within VMM that you will gain access to features such as:

- Physical to Virtual Conversion (P2V)
- Virtual Server or VMware to Hyper-V Conversion (V2V)
- PowerShell Scripting
- VM Templates/Cloning
- Failover Cluster Integration

And Microsoft is committed to adding functionality to VMM. Currently, their 2008 version is in beta and offers several new features. Among the updates are improved utilization of Server 2008 foundational features; comprehensive support for VMware VI3, including VMotion; and a re-engineered administrative permissions engine. The new permission engine includes the ability to create the delegated administrator role, which maintains the management abilities of a full administrator but on a reduced scope of responsibility, such as a designated sub-set of virtual hosts.

4. Space

The one technical area where Hyper-V outshines its rival is when it comes to disk space. Unlike ESX, Hyper-V supports differencing disks. The differencing disk can be used to create multiple environments derived from the same base image with much less disk space. ESX doesn't support differencing disks due to performance concerns, and therefore, to derive from a base image, a full clone is needed, which doubles the disk usage.

This is extremely noteworthy because chances are that a lot of your servers could be built from a standard base image, which is a particularly common configuration. A single virtual hard disk can be cloned and used multiple times over while only marginally increasing your disk usage.

The way it works is the differencing disk creates a set of modified blocks in comparison to a parent image. The differencing disk, the child, stores a record of all changes made to the parent disk and provides a way to save changes without altering the

parent disk. In other words, by using differencing disks, you ensure that changes are made by default to the differencing disks and not to the original virtual hard disk. You can, however, elect to merge changes from the differencing disk to the original virtual hard disk when it is appropriate to do so. The differencing hard disk expands dynamically as data intended for the parent disk is actually written to the differencing disk.

It is important to note that you should definitely write-protect or lock the parent disk. Otherwise, if the parent disk is modified by some other process that does not recognize the parent/child relationship of the differencing disk, all differencing disks related to the parent disk become invalid, and any data written to them is lost. By locking the parent disk, you can mount the disk on more than one VM, similar to a read-only CD-ROM. Although performance may not be on par with a full-fledged virtual disk, differencing disks are a great option to have available and a great way to dramatically improve space saving on your SAN over time.

5. Manageability

The final reason to switch to Hyper-V is because it is likely that you are already better at managing it. Why? Because even though it is a new product, at the end of the day, it's Windows, and you know Windows. VMware is its own entire operation system, and one could easily spend entire weeks if not months diving into how to properly set up and configure ESX and Virtual Infrastructure 3, particularly if you have little or no experience with Linux.

To perform such common tasks as manage permissions, copy an existing VM, or transfer files to your host involve either basic Linux administration knowledge or learning VMware's or a third-party's tools. Whereas, on the Hyper-V side, if you've been doing Windows administration for a while, these tasks should be quite easy for you. If you're already familiar with managing Windows permissions, copy/pasting a VHD is no different than copying/pasting any other file format, and you can access the hosts' file structure just like any other server in your domain.

A final consideration when deciding on a virtualization solution is integration. Do not overlook how well the solution you choose will integrate into your existing IT infrastructure.

Steve Giovanni has been in the IT industry for more than 10 years, but he has been around computers his entire life. He is an IT consultant, an active member of the IT community at large online, and an all-around technical evangelist at heart. Steve has earned a B.S. in Computer Science, and has several IT certifications including CCNA, MCP, and an MCTS in Exchange 2007. Steve has written a number of articles on a variety of IT topics, specializing in solutions involving Windows Server, Exchange, and Hyper-V. Steve is a feature contributor to the Windows Administration in Realtime eJournal *from Realtime Publishers.*

The Deep Dive

Automating Software Deployment for the Small IT Shop, Part II

by Greg Shields

Last month's article discussed the sometimes complex steps involved with packaging software for automated distribution. That article was rightfully titled *The Art of Software Packaging* because the process of finding either the right set of "silent switches" or the best "diff" can sometimes take a little slight-of-hand. Yet while packaging software to enable it to run silently requires a bit of intuition along with the ability to construct a good Google search string, once complete, the rest is easy.

In this, part two of the series, let's assume that you've completed the repackaging process for the software you're interested in automatically deploying. You've either located the necessary silent switches or you've repackaged the software using "diff" tools such as WinINSTALL LE. Now you need to deploy it to desktops around your environment.

Large enterprise organizations often have the staffing, the funding, and the process maturity already in place to acquire high-end systems management solutions such as Altiris or Microsoft's System Center Configuration Manager. These enterprise-class management solutions easily enable the distribution of software to highly targeted collections of computers with rich reporting and troubleshooting functionality already baked in. However, small IT shops rarely have the budget to purchase these tools, let alone the available time for administrators to learn their nuances.

Yet the small IT shop isn't left completely out in the cold. A number of low-cost systems management platforms from companies such as Kaseya, KBOX, Hyena, and others are quickly making names for themselves. Although these platforms may not scale to the level of the enterprise solutions, their installation, maintenance, and use are designed with the small shop in mind. These tools incorporate graphical interfaces for scripting client actions, packaging and delivering software and updates, and inventorying computer hardware and software with just a few mouse clicks.

Freeware solutions are available too. Two in particular are worth looking at because both involve no added cost for the organization with an Active Directory (AD) infrastructure in-place. The first, *Group Policy Software Installation*, leverages AD's native Group Policy for the distribution of MSI files. The second, a freeware tool called PSExec from the former Sysinternals team, enables a slick way to remotely install software to individual machines across the network. For either of these, you'll need a file share somewhere on your network where packages are stored and can later be accessed.

Group Policy Software Installation

Software installation using Group Policy automatically takes advantage of your existing AD infrastructure. If Group Policy is successfully functioning in your network today, you've already got the infrastructure in place to immediately begin distributing software. The process starts with creating a new Group Policy Object (GPO) and navigating to *Computer Configuration* | *Policies* | *Software Settings* | *Software Installation*. Right-click the node and choose New | Package. In the resulting dialog box, navigate to the file share that contains your repackaged .MSI installation and select the software to be installed. Ensure that you're connecting to your .MSI file through a commonly accessible file share or clients may not be able to access the file share when they attempt to later install the software.

Group Policy Software Installation can only be used for installing .MSI files. It is possible to work with other types of installations using special .ZAP files, which are beyond the scope of this article.

When prompted to select the deployment method, choose *Assigned*. Finally, apply the GPO to an Organizational Unit (OU) of computers. The next time those computers reboot, they will automatically download and install the software as part of the initial Group Policy processing during startup.

By viewing the properties of the installation, you will find a number of choices that determine how the software installation behaves and works with the targeted client systems. One configuration in particular to be aware of is under the *Deployment node* in the box marked *Uninstall this application when it falls out of the scope of management*. Be careful with selecting this box. By checking this box, any time the computer finds that it is no longer under the scope of management of this particular Group Policy the software will automatically uninstall. Although this allows for a quick reconfiguration of computers as they change OU memberships over time, it can have unexpected consequences when not properly prepared for.

Group Policy Software Installation has a number of additional properties and mechanisms for configuration other than what is described here. As an example, it is also possible to configure the software installation under User Configuration instead of Computer Configuration, a change that offers installation options to user rather than an enforced install.

For details about other ways to deploy software, see the Microsoft knowledgebase article at http://support.microsoft.com/kb/816102.

Sysinternals PSExec

AD and Group Policy are excellent ways to automatically and rapidly distribute software across large swaths of your environment when the need exists. But there are times when you need only to install software to one or two computers in the environment. Perhaps those computers are outside the scope of an existing Group Policy where software is otherwise installed. Or, perhaps installation through Group Policy isn't behaving properly. In either case, you may find yourself occasionally in the need to install software to small numbers of computers.

Doing this through Group Policy is challenging because an individual computer object can only exist in one OU at a time. In this case, the command-line Sysinternals tool PSExec can assist. PSExec is a freeware tool that is part of the PSTools. These tools can be downloaded from http://technet.microsoft.com/en-us/sysinternals/bb896649.aspx. Once downloaded to your local desktop, you'll need to manually copy them to a location in your computer's path such as *C:\Windows*.

PSExec enables the launching of processes on remote computers around your environment. For example, to launch a command prompt on a remote computer named \\computerName, use the syntax:

```
psexec \\computerName cmd
```

The msiexec.exe process is the executable used to launch all .MSI installations. Like any executable, it can be remotely launched using PSExec. Launching msiexec.exe with the right set of switches and pointing it to a commonly accessible server share that contains a repackaged .MSI file will start an installation. Doing so with PSExec can start that installation remotely over the network. This is usually done when the user is logged out of the computer, so as not to conflict with any work being done. You can minimally accomplish this with the command:

```
psexec \\computerName "c:\windows\system32\msiexec.exe" /i \\server\packageShare\setup.
msi
```

Obviously, if you've determined that the installation requires other switches or parameters for a successful installation, add them to the simple previous example. Once launched, you can use the tool PSList, also from the PSTools, to monitor the process list of the computer that is installing the software. To see a constantly updating list of processes in order by processor use that are being run on the remote computer, use the command:

```
pslist \\computerName /s
```

In that list, keep an eye on the msiexec.exe process. Once its process returns to zero use of the processor, you can then have a good expectation that the software installation is complete.

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.

Practical PowerShell

Managing Active Directory Users with Windows PowerShell

by Jeffery Hicks

If you've been a regular reader, by now you've come to recognize the value Windows PowerShell brings to your administrative toolkit. Unfortunately, as great as PowerShell v1.0 is, there are limitations. For example, Microsoft did not ship any cmdlets for managing Active Directory (AD) and probably won't for some time yet. But that doesn't mean you can't use PowerShell for this task.

> You can download the New-User.txt and Import-User.txt files from the Realtime site at http://www.realtime-windowsserver.com/code/v1n10_Practical_PowerShell.zip.

The [ADSI] Way

First, because PowerShell is based on the .NET Framework, it can access the .NET classes designed for directory services such as AD. The primary class is a System.DirectoryServvices.DirectoryEntry. You can create a new object of this class using the **New-Object** cmdlet.

```
PS C:\> $user=new-object system.directoryservices.directoryentry "LDAP://CN=Jack Frost,OU
=Payroll,OU=Employees,DC=BigCompany,DC=local"
PS C:\> $user
distinguishedName
-----------------
{CN=Jack Frost,OU=Payroll,OU=Employees,DC=bigcompany,DC=local}
```

That's a bit much to type, so the PowerShell team added an [ADSI] type accelerator. By casting a variable as an [ADSI] object, PowerShell knew to treat a distinguished name as a DirectoryEntry object.

```
PS C:\> [ADSI]$user="LDAP://CN=Jack Frost,OU=Payroll,OU=Employees,DC=bigcompany,DC=local"
```

Now, you can use the $user object and manage different properties of this user account.

```
PS C:\> $user.title
Accountant
PS C:\> $user.department
Payroll
```

You can pipe $user to **Get-Member** to see all the populated properties. You can change most properties, even ones that don't show, as long as you know the name.

```
PS C:\> $user.telephonenumber=555-1108
PS C:\> $user.SetInfo()
```

When using ADSI, remember that you need to call the **SetInfo()** method to commit any changes to the directory service. You won't see this method if you pipe $user to **Get-Member**, but it exists and you need to use it.

Another method you will likely want to use is **SetPassword()**. Again, **Get-Member** doesn't display it, but you can call it:

```
PS C:\> $user.SetPassword("P@$$wrd")
```

This method commits the change to AD immediately, so there is no need to call **SetInfo()**.

Creating a user account in AD using ADSI takes a few steps, and while you can type each command at the PowerShell command line, I think you'll find it easier to use a script or function like the following example.

```
#New-User.ps1
#You will want to change the default container path
    Param([string]$OUPath="CN=Users,DC=bigcompany,DC=local",
          [string]$Name=$(Throw "You must enter a first and last name"),
          [string]$SAM=$(Throw "You must enter a SAMAccountname"),
          [string]$password="P@ssw0rd",
          [string]$description="Created "+(get-date),
          [string]$upnsuffix="bigcompany.com",
          [boolean]$enable=$True,
          [boolean]$forcepwd=$True
    )

    #validate that a first and last name was entered
    if ($Name.split(" ").Count -ne 2) {
        Write-Warning "You must enter a first and last name for the name parameter."
        Return
    }

    [ADSI]$OU="LDAP://$OUPath"

    #if OU doesn't exist quit the function
    if (!$OU.DistinguishedName) {
        Write-Warning "Failed to connect to $OUPath"
        return
    }
```

```
#Add the user object as a child to the OU
 $newUser=$OU.Create("user","CN=$name")
 $newUser.Put("sAMAccountName",$SAM)

 #commit changes to Active Directory
 $newUser.SetInfo()

 #set a password
 $newUser.SetPassword($password)

 #Define some other user properties
 $newUser.Put("DisplayName",$name)
 $newUser.Put("UserPrincipalName","$sam@$upnsuffix")
 $newUser.Put("Description",$description)

 #first name
 $newUser.Put("GivenName",$name.split()[0])
  #last name
 $newUser.Put("sn",$name.split()[1])

 #enable account = 544
 #disable account = 546
 if ($enable) {
     $uac=544
     }
 else {
     $uac=546
     }

 $newUser.Put("UserAccountControl",$uac)

 if ($forcepwd) {
     #flag the account to force password change at next logon
     $newUser.Put("pwdLastSet",0)
     }

 #commit changes to Active Directory
 $newUser.SetInfo()
```

This might look like of lot of PowerShell commands, but the number of required steps to create a new user is very small. You first need an ADSI object for the parent container or OU where you will create the account.

```
 [ADSI]$OU="LDAP://$OUPath"
```

The $OUPath variable is a script parameter that you can specify when running the script or use a default setting. To create the account, call the Create() method, specifying the class of object you are creating, a user, and its name:

```
    $newUser=$OU.Create("user","CN=$name")
```

You'll also want to specify a sAMAccountname for the new user as well, and save the changes to AD:

```
    $newUser.Put("sAMAccountName",$SAM)

    #commit changes to Active Directory
    $newUser.SetInfo()
```

The rest of the script sets some additional user properties, such as the user principalname, password, a description, and more. You can add or modify as many properties as you'd like. Here's how you might use this from a PowerShell prompt:

```
PS C:\> c:\scripts\New-User -OUPath OU=Testing,DC=mycompany,DC=Local" -Name "Wes East"
-sam "weast" -upn "mycompany.com"
```

Any values that I don't specify will use the default script settings.

There is a great deal you can do using the [ADSI] type adapter and the other .NET DirectoryService classes; however, it requires a lot of work. Plus, unless you develop your scripts and functions accordingly, you can't easily leverage the pipeline. The better solution for managing AD users is with a set of cmdlets.

The Quest Cmdlet Way

Quest Software has released a set of free cmdlets for managing AD. Go to http://www.quest.com/powershell/activeroles-server.aspx and download the Active Roles Management Shell for Active Directory. There are 32- and 64-bit versions. Install it on your desktop or wherever you will be running PowerShell. It does not need to be installed on a domain controller.

The installation will add a new program item to your start menu that will launch a PowerShell window using the Quest snap-in, but I prefer to add this line to my PowerShell profile and use my regular PowerShell session:

```
add-pssnapin Quest.ActiveRoles.ADManagement
```

Open a new PowerShell session and run this command to see all the cmdlets included in the Quest snapin:

```
PS C:\> get-qadcommand
```

Let's use some of these cmdlets to create and manage AD user accounts. To create a new user, you'll naturally use the **New-QADUser** cmdlet. If you look at the cmdlet help, you'll notice that most account properties are exposed as parameters.

```
PS C:\> new-qaduser -parentcontainer "OU=Temp,OU=Employees,DC=bigcompany,DC=loca
l" -name "Jim Shortz" -firstname "Jim" -lastname "Shortz" -samaccountname "JShortz"
-userprincipalname "jshortz@bigcompany.com" -title "Sales Agent" -Department "Marketing"
-phonenumber "555-9800" -description "contract employee" -displayname "Jim Shortz"
-userpassword "P@ssw0rd123"

Name                           Type           DN
----                           ----           --
Jim Shortz                     user           CN=Jim Shortz,OU=Temp,OU=Employees,DC=bigc
ompany,DC=local
```

With a one line, albeit long, PowerShell expression, I created a new user account and populated a number of user properties all at the same time. The only required parameters are the parent container and the name. But I always specify at least a sAMAccountname and user principal name.

By default, the account is disabled, but that it is easily changed with another cmdlet:

```
PS C:\> enable-qaduser "Jim Shortz"
```

The last step you might want to take is to force the user to change password at next logon. You can modify the user account using Set-QADUser:

```
PS C:\> Set-QADUser "Jim Shortz" -UserMustChangePassword $True
```

Remember, these are cmdlets, so you can leverage the pipeline and anything you can do for one account, you can do for 10, 100, or 1000 accounts. Let's import a CSV file and create new user accounts. For the sake of simplicity, we'll create all the users in the same OU, and the CSV file's header line uses the New-QADUser cmdlet's parameters such as FirstName, LastName, Title and sAMAccountname. My CSV file is using this header:

```
OU,Name,Firstname,Lastname,SAMAccountname,Telephone,Office,Department,Title,City
```

Armed with this CSV file, I'll use **Import-CSV** and pipe each imported object to **ForEach-Object**, which uses **New-QADUser** to create the new account.

```
Import-Csv newusers.csv | ForEach-Object {
    New-QADUser -parentcontainer $_.OU -name $_.Name `
    -samAccountName $_.samaccountname `
    -firstname $_.FirstName -lastname $_.LastName `
    -title $_.title -department $_.department -company "BIGCompany" `
    -phonenumber $_.Telephone -userpassword "P@ssw0rd" `
    -office $_.Office -userprincipalname ($_.samaccountname+'@bigcompany.com') `
    -displayname ($_.FirstName+' '+$_.LastName) `
    -City ($_.City) -description "Created by New-QADUser" `
 | Enable-QADUser | Set-QADUser -UserMustChangePassword $True
  }
```

Each new user account is then piped to **Enable-QADuser**, which then pipes the new user account to **Set-QADUser**, which flags the account so that the user must change their password at next logon. If you need to find or search for user accounts, it doesn't get any easier than using **Get-QADUser**. You can find a specific user account and display specific properties:

```
PS C:\> get-qaduser "Jim Shortz" | format-list Name,Title,Department,Phonenumber


Name        : Jim Shortz
Title       : Sales Agent
Department  : Marketing
PhoneNumber : 555-9800
```

Or find users based on one or more criteria:

```
PS C:\> get-qaduser -department sales -city boston | Select Name,Title

Name                                                    Title
----                                                    -----
Brenda Wright                                 Account Executive
Karen Green                                             Rep
```

I've found the user accounts that are in the Sales department and in Boston and displayed the name and title properties. As you see from this example, I can pipe user objects from between cmdlets. This makes it very easy to make mass changes to user accounts. Use the **Get-QADUser** cmdlet to retrieve one or more user accounts and pipe them to **Set-QADUser** to modify.

Let's suppose the Chicago office has been relocated to Kansas City. This one-line command quickly updates the city and telephone number for all user accounts in the entire domain:

```
PS C:\> get-qaduser -city Chicago | set-qaduser -city "Kansas City" -phonenumber "555-
9000"
```

As you work with the Quest cmdlets you'll realize that they present many AD properties into a more user-friendly format. For example, the Password age property is stored in AD as the number of elapsed seconds, but the Quest cmdlets translate this into something more meaningful:

```
PS C:\> get-qaduser -sizelimit 0 -enabled -passwordneverexpires $false | sort PasswordAge
-desc | Select -first 50 | Select Name,Distinguishedname,PasswordAge

CN=Charles Dickens,OU=Legal,OU=Employees,DC=bigcompan... 203 days
CN=Fred Flintstone,OU=Temp,OU=Employees,DC=bigcompany... 141 days
CN=Rock Hudstone,OU=Temp,OU=Employees,DC=bigcompany,D... 141 days
CN=Wilma Flintstone,OU=Temp,OU=Employees,DC=bigcompan... 141 days
CN=Cary Granite,OU=Temp,OU=Employees,DC=bigcompany,DC... 141 days
CN=Eugene Slate,OU=Temp,OU=Employees,DC=bigcompany,DC... 141 days
CN=Anne Sample,OU=Employees,DC=bigcompany,DC=local       141 days
CN=Tom Sawyer,OU=Sales,OU=Employees,DC=bigcompany,DC...  133 days
CN=Roy Biv,OU=Executive,OU=Employees,DC=bigcompany,DC... 129 days
CN=Sam Apple,OU=Finance and Investments,OU=Employees,... 127 days
CN=Olga Jimenez,OU=Development,OU=Employees,DC=bigcom... 126 days
CN=Sean Gonzales,OU=Physical Plant,OU=Employees,DC=bi... 126 days
CN=Felicia Sutton,OU=Payroll,OU=Employees,DC=bigcompa... 126 days
CN=Philip Bryant,OU=Physical Plant,OU=Employees,DC=bi... 126 days
CN=Tracey Barrett,OU=Product Development,OU=Employees... 126 days
CN=Alan Butler,OU=Physical Plant,OU=Employees,DC=bigc... 126 days
CN=Nina Soto,OU=Public Affairs,OU=Employees,DC=bigcom... 126 days
```

If your query will return more than 1000 objects, use the –SizeLimit parameter set to 0 to return all objects. The rest of the Get-QADUser expression is to return all enabled user accounts with passwords that can expire. A report like this might be useful because maybe you want to disable all these accounts with a password age greater than 100 days.

```
PS C:\> get-qaduser -sizelimit 0 -enabled -passwordneverexpires $false | where {$_.
passwordage.value.days -ge 100} | disable-qaduser -whatif
```

The Quest cmdlets support –WhatIf and –Confirm, which I encourage you to use when running an expression like this.

Finally, let's look at adding and removing users from groups using the **Get-QADGroupMember**, **Add-QADGroupMember**, and **Remove-QADGroupMember** cmdlets. The **Get-QADGroupMember** will return all group members of a specified group:

```
PS C:\> get-qadgroupmember "Las Vegas Staff" | select name

Name
----
Cass Ino
```

Now look how easy it is to add users to a group. This one-line command gets all user accounts with a city property of Las Vegas and adds them to the Las Vegas Staff group:

```
PS C:\> get-qaduser -city "las vegas" | Add-QADGroupMember "Las Vegas Staff"
Name                           Type           DN
----                           ----           --
Joe Friday                     user           CN=Joe Friday,O...
Sam Apple                      user           CN=Sam Apple,OU...
Mark White                     user           CN=Mark White,O...
Clarice Starling               user           CN=Clarice Star...
Elmer Fudd                     user           CN=Elmer Fudd,O...
Randy Redd                     user           CN=Randy Redd,O...
Roy Brown                      user           CN=Roy Brown,OU...
```

Now suppose Sam Apple gets transferred from Las Vegas to Miami. With a single PowerShell expression, I can change his city property, add him to the Miami Staff group, and remove him from the Las Vegas Staff group:

```
PS C:\> get-qaduser "Sam Apple" | set-qaduser -city "Miami" | add-qadgroupmember "Miami
Staff" | remove-qadgroupmember "Las Vegas Staff" | select Name,City

Name                                                          City
----                                                          ----
Sam Apple                                                     Miami
```

Sure, you could manually have made the change using Active Directory Users and Computers, but what if you had 1000 accounts to change? PowerShell and the Quest cmdlets would get this task done in seconds.

As you might imagine, there is much, much more that you can accomplish with these cmdlets. I wanted to give a sample of what is possible in PowerShell. I encourage you to look at the cmdlet help and examples and to download the Administrator's Guide when you download the install file. I also explore these cmdlets and ADSI in great detail in *Managing Active Directory with Windows PowerShell: TFM* (SAPIEN Press, 2008). ◆

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.

What's the Deal with Resource Mailboxes?

by J. Peter Bruzzese

You go to create a mailbox, start the new mailbox wizard in Exchange 2007, and you are presented with a whole list of options (see Figure 1). Two that stand out are the Room and Equipment mailboxes. The concept that inanimate objects should be given mailboxes just doesn't make sense, perhaps. But the use of a mailbox to represent a resource is an ingenious way of providing the ability to schedule "time" with those resources when needed.

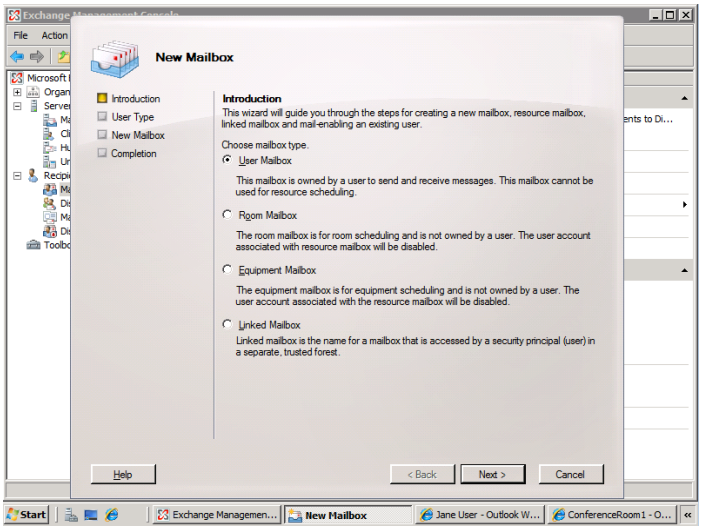

Figure 1: The many different mailbox types you can create.

The Room and Equipment Mailboxes

You can use the resource mailboxes in the following ways:

- Room—Conference rooms, auditoriums, special AV rooms, executives-only rooms, and/or training rooms
- Equipment—Projectors, laptops, company cars, AV carts, and so forth

When you create these mailboxes, you also create an account in Active Directory (AD) for those resource mailboxes. The accounts, however, are disabled by default. Thus, a person cannot logon with a resource mailbox account. But because there is a mailbox, you can manage the account.

You can make configuration additions to the properties of the account once the account is created. To do so, locate the mailbox and go into the properties from the Exchange Management Console (EMC). You'll notice with resource mailboxes that there is a Resource Information tab (see Figure 2).

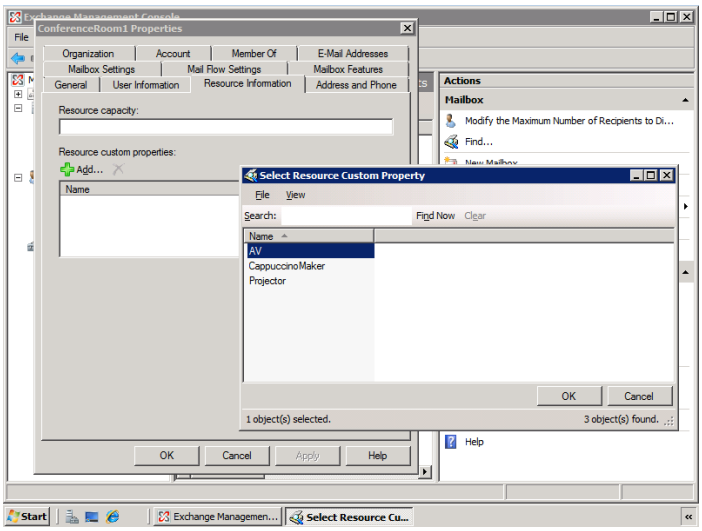

Figure 2: Resource properties.

You can indicate a resource capacity (for example, with a room, the number of people that are legally allowed in the room). You can also indicate resource custom properties by clicking Add and choosing from a list of options that you create in the Exchange Management Shell (EMS) using the

cmdlet Set-ResourceConfig. So, in the example that Figure 2 shows, you see the Resource Custom Properties with a variety of options that I included within the EMS using the following command:

```
Set-ResourceConfig
-ResourcePropertySchema ("Room/
Projector", "Room/CappuccinoMaker",
"Room/AV")
```

Full Access Permissions

Your next step, either through the EMC or the EMS, is to provide another user with the ability to control the resource. Consider an illustrative scenario: there are five conference rooms that are usually organized by one person. That person isn't you (hopefully) because you are the network administrator. Whoever that person is, that is the one you want to give your resource mailboxes full access permissions over so that the person can view, alter, and configure schedules and scheduling conflicts for those rooms or that equipment.

To provide permissions over a resource mailbox, you locate and select that mailbox in the EMC. Then, from the actions pane, select Manage Full Access Permission. Click Add, and add the user you to whom you want to provide full control. Then click Manage and Finish, and that user will be able to access the mailbox.

Have the user open their own mailbox in Outlook or Outlook Web Access (OWA); then have them open another mailbox (the resource mailbox) from their account. Once the other mailbox is open, go into the options for the mailbox. Two primary areas of concern are the Calendar Options (to establish when the resource is available for use) and Resource Settings (to configure a whole slew of options on how the resource can be utilized and by whom).

It might take some getting used to, but once you have it organized, the resource mailboxes are really going to prove helpful, especially when scheduling meetings. Notice that the scheduling assistant (in Figure 3) allows you to

view rooms that are available on any given day and shows you the availability for those rooms. So, when booking a meeting location, you can quickly see which conference room you are going to be able to use and, depending on the number of people, whether the room is big enough for your meeting or you will have to pick a different time and location.

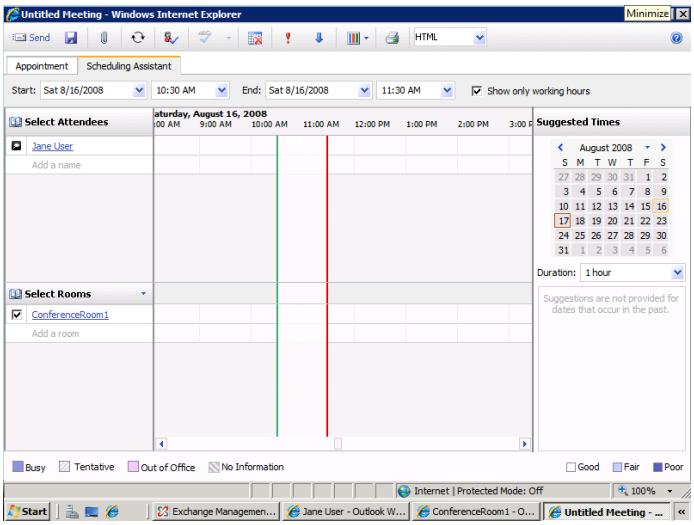

Figure 3: The Scheduling Assistant helps show you what is available for both people and resources.

Coming Into the 21st Century

It's true, some people prefer the old-fashioned way of keeping track of conference rooms—a human with a calendar and lots of pencil marks. I liked the 70s too. Maybe we can type it all up on a typewriter and get some of that carbon copy paper to give others the schedule. Jokes aside, the resource mailboxes offer the ability not just to keep the schedule of your resources but also for others to clearly see what is available to them, as well as to know whether a room is available, when it will be available, what size the room is, and whether that room has the projector you need or other AV equipment. Resource mailboxes are a great way to keep your people informed.

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. In harmony with the joy of writing Exclusively Exchange for Realtime Publishers, he has created a free Exchange training site at www.exclusivelyexchange.com. He is co-founder of ClipTraining.com, a provider of short, educational screencasts on Exchange, Windows Server, Vista, Office 2007 and more. You can reach Peter at jpb@cliptraining.com.

Copyright Statement