Realtime
publishers

June 2009
Volume 2, Number 6

Windows Administration
in Realtime

2 **Letter from the Editor**
Fixing "The People Who Fix"

4 **Break the Habit**
Logging Onto the Console

6 **Product Review**
SecureCopy

8 **Improving Collaboration with Microsoft Office Communications Server**
Are Voice, Video, and Messaging Part of Your Corporate Growth Plan?
By: Jim Varner - Employ MOC to leverage your existing VoIP investment and provide a higher level of productivity across your organization.

12 **The Deep Dive**
Dealing with Multi-Vendor Logs
By: Jan Kanclirz Jr. - Discover a tool that will improve your organization's security, efficiency, and compliance.

14 **Practical PowerShell**
Giving You an Out
By: Jeffery Hicks - Use PowerShell to capture output to a real Microsoft Word document.

28 **Exclusively Exchange**
Exchange 2010 Databases and High Availability Adjustments
By: J. Peter Bruzzese - Discover a major change in Exchange 2010.

Letter from the Editor

Fixing "The People Who Fix"

by Greg Shields

Incentives are great. When I'm buying a new car, the incentives for a lower APR or better stereo are sometimes just enough to get me to make that purchase. But you can argue that bad incentives are worse than no incentives at all.

Nowhere is this more pronounced than with the incentives I sometimes see in IT organizations. I've had my share of visiting the IT organizations of numerous businesses both large and small. In each of those opportunities, I see IT professionals merrily going about their daily tasks, solving problems and fixing computers when they break. They tell me about their work order tracking systems, and how they delight in resolving incoming tickets when they show up in their queue.

Sound like your job? It's a common sight in companies everywhere.

In some of those organizations, I find well-meaning managers creating incentives for their teams based on the number of tickets they can resolve. "The person who resolves the most tickets this week gets a $50 gift certificate at Best Buy!"

Yet sometimes I wonder if incentives like this are actually counterproductive in the long run. A few years ago, I found myself explaining what I did for a living to a friend who doesn't work in IT. When asked that age-old question of, "So, what do you do for a living?" I prepared my usual response that centered around the facts and perils of fixing computers. But in talking about just that response, I remembered that my job wasn't really to "fix" them but to *"keep them running"*.

I worry sometimes about those incentives for enforcing good staff behaviors. Incentivizing the resolution of problems indeed resolves work orders, but at the same time has the tendency to drive IT down an exclusive break/fix mentality. "If it breaks, we fix it." But in these organizations, what happens when no one's job is to focus on prevention? Who keeps the problems from happening in the first place if everyone in IT is focused only on fixing?

I was consulting not long ago for a large investment bank. This bank had grown in size to the point where its reactive-minded IT organization was actually creating problems with bank operations. In one specific instance, their Exchange Server found itself running out of space on a regular basis, each time requiring a day-long resolution process due to a dirty shutdown of the private store.

While there, I started asking some very pointed questions, "You really need to consider implementing monitoring as well as configuration control to prevent this from happening again. How can we do this?"

To which the response was, "We've really been meaning to get something like that implemented, ***but we just haven't had the time with all the work orders coming in***." This organization incentivized fixing problems so much that no one in the organization ever had or made time to actually step back and think about preventing them.

Sound like your job? If so, consider a suggestion for changing how you do business. Start today by finding one individual in your organization that you can remove entirely from the daily grind of resolving work orders. If you can't afford an entire person, start by giving them half their day back. Challenge that person with spending their time looking at ways to prevent problems from happening. This can happen through improved change control, the implementation of configuration management and documentation, as well as automation through scripting and other tools. Incentivize this person through the number of work orders that ***don't*** get created over a particular period. Above all, draw a hard line between those that are responsible for the daily grind of dealing with customers and work orders and the job of this individual. Insulate them from the rest of the world while giving them the charter to improve it.

You won't find results overnight. But over the course of a few months, or a year, you might find that your sheer volume of requests actually goes down.

Let me know how it goes. The best story gets written up in a future edition of this eJournal. Drop me a line at gshields@realtimepublishers.net. ◆

Break the Habit

Logging Onto the Console

by Don Jones

In this new monthly column, I'm going to focus on *worst* practices—starting with the ones I see happening in some of my consulting clients, but eventually looking at those bad practices you send in (visit the "worst practices" discussion at http://concentratedtech.com/groups/show/13 to tell me your horror stories). The idea is to recognize that none of us are perfect, and that we've got a few bad habits, administration-wise, that we could afford to get rid of.

Logging onto the Console

This is probably one of the worst things I see happening in some of my clients. It happens in a variety of ways, too:

- Some admins just waltz into the data center, hit Ctrl+Alt+Delete, and start running admin tools—which they've installed onto the server, of course.
- Other admins use those IP-based Keyboard-Mouse-Video (KMV) switches so that they can type right on the console without the hassle of making a trip into the data center.
- Still others rely on good old Remote Desktop Services (formerly Terminal Services) to remotely spin up a console session. This practice is seen as "less bad" because those sessions are virtualized and aren't usually the "real" console session.

They *all* drive me nuts. See, I started back in the NetWare days, when a server was a server and not a workstation. Servers didn't have a GUI or even much of a command line. They had enough local functionality to get things up and running, and that's about it—and honestly, that's the way things *should* be.

Don't get me wrong: Microsoft got Windows where it is today because they created a server OS that looked and worked a lot like the desktop OS everyone was familiar with. This let departments and other small groups within a company deploy their own file or print server without having to hire an expensive server support person—in many cases, a smart desktop user could keep things maintained. But it set a bad precedent, and a lot of those "smart desktop users" became the "paper MCSEs" that many of today's administrators get so frustrated with.

Wait, why is logging onto the console so bad? Because every time you do it, the server—which really should have better things to do—has to spin up an entire graphical desktop session for you. When you log off, it has to de-allocate all that memory. Logging onto the server nearly always means you're logging on as an administrator, too, so you're opening up enormous opportunities for wrongdoing in the form of malware. Yeah, I know—your servers have anti-malware software installed. That didn't stop *hundreds* of servers—yes, *servers*—from being infected with the Conficker worm, in many cases through vectors that could only have worked *in a console session*. Stay off the console and the server doesn't execute nearly as much code, and the server offers fewer opportunities for code to execute—meaning it stays safer. Fewer graphical sessions being started and stopped also means more efficient memory use, and we all know that Windows does *much* better with lots of efficiently-used memory, right?

So why are console logons so common? In some cases, sheer ignorance. I've run across newer admins who simply don't realize that they can install their admin tools—mainly MMC snap-ins—on their desktops. One argued with me for hours over Exchange Server, insisting that we couldn't install Exchange on Vista. I agreed, but pointed out that the Exchange installer knows that, and if you run it on Vista, it will only offer to install the management tools. In other cases, console logons are Microsoft's fault, such as the huge delay in releasing admin tools that ran on Vista. In other cases, it's a lower level of technical proficiency, such as admins who don't know how to make use of existing remote management tools—especially those that require command-line expertise. Finally, in still other cases, it's Microsoft's fault again, for not making some critical management capabilities—such as easily setting up network connections and other core stuff—available remotely.

You can kick this habit by carefully examining every task you do on the server console, and looking for ways to do those remotely. Windows PowerShell v2 (coming in Win7 and Win2008R2, and which will be made available for older versions of Windows) will help tremendously because it—for the first time—offers a supported "remote command-line" capability for servers. Newer Microsoft products continue to expand remote management options, too, including MMC snap-ins and PowerShell snap-ins. In many cases, some of the hundreds of existing command-line tools can do the job just fine—if you take the time to hunt them down and learn how to use them.

Kick the console habit: Let your servers be servers, and manage things from your workstation. ◆

Don Jones is a co-founder of Concentrated Technology. Join him and cohort Greg Shields for intense Win2008 and Windows PowerShell training—visit ConcentratedTech.com/class for more details. Ask Don a question by visiting ConcentratedTech.com and using the "Contact" page.

Product Review

SecureCopy

by Eric Schmidt

In large enterprises, storage migrations can be one of the most challenging tasks that need to be accomplished every time existing hardware is retired. Often, this task requires moving terabytes of data between systems in a way that is transparent to the users. The other requirement is that the integrity of the data and the associated access control lists (ACLs) must also be maintained. Although some administrators are fortunate enough to have sufficient funding to have the migration performed by the storage vendor, others must rely on more affordable tools to get the job done. These tools usually come in the form of scripts or batch files or perhaps by leveraging free tools such as RoboCopy. There is an alternative—a product called SecureCopy from ScriptLogic.

SecureCopy is a file copying tool that offers a significant number of features that simplify file server migrations. SecureCopy has the ability to copy files, folders, NTFS permissions, file shares, local users and groups, and compression settings.

Unlike command-line utilities like RoboCopy, SecureCopy offers a very simple, easy-to-read user interface (UI) that starts by establishing the source and destinations for files and folders. Once those have been identified, the interface organizes all the options into tabs, starting with basic options for controlling how the copy will take place.

One of the best features is the mirror option. On large file system migrations, it's best to perform the copy over time and allow the mirror functionality to keep everything in sync. On the day that the cutover will take place, a synchronization can be run before the old server is turned off. Using this method, the final cutover will be much faster because the bulk of data has already been copied; all that needs to be copied are the files/folders that changed since the last synchronization took place.

The other features available include copying permissions, encrypted files, and file timestamps. Timestamps are the attributes that are particularly important to copy because they can help with overall data management. If timestamps are not copied over, all files on the new server will have the time that they were copied instead of the actual last time they were opened or modified.

A couple of benefits that this product has over tools such as RoboCopy include the ability to verify files after they have been copied and the ability to migrate SID history. Verification is an important safety mechanism to insure that files are the same at both the source and destination. SID history may also be important if domain migrations have taken place and users are still relying on SID history to access files.

In addition, SecureCopy allows file shares and local groups to be migrated. When doing a large file server migration in which there are a significant number of shares and local groups, the ability to copy them with a tool instead of recreating them on the destination can be a tremendous time saver.

SecureCopy also has the ability to filter the files that will be copied either by extension or path. In situations in which data is being restructured either as part of a migration or for consolidation, this filter feature can be useful in controlling what data gets put where. One of the possible uses of filtering is to exclude files that aren't supposed to be stored on the server, such as .mp3 music files.

From a performance perspective, SecureCopy is able to leverage multiple threads. This enables large migrations to better take advantage of network and system resources in order to make the migrations run as fast as possible. The number of threads that are used is a user-controlled option to help ensure that resources are not overloaded.

The final benefit that SecureCopy provides is the ability to easily schedule and manage jobs. With free tools such as RoboCopy, the only way to schedule a job is to create a batch file and a scheduled task. With SecureCopy, jobs can be scheduled, enabled, and disabled within the application. The layout of the scheduled jobs is also very easy to read and provides quick access to all aspects of the jobs that have been scheduled, including the last run time and the current status.

Overall, SecureCopy can be a valuable resource for organizations that lack the funding for vendor-provided solutions to perform large-scale file server migrations. SecureCopy's usefulness is not limited to large file migrations; it can also be effectively leveraged for most situations where files have to be moved from one location to another. The UI is very clean and intuitive, which enables administrators to quickly create jobs and avoid making the mistakes prone to command-line tools.

Eric Schmidt works as Enterprise Microsoft Security Technologist, with Honors, for Raytheon Company and has worked in Information Technology for 13 years. Eric has a Masters degree in Computer Information Technology and has developed extensive experience in systems administration, engineering, and architecture specializing in Microsoft Active Directory and Systems Management. Eric has been well recognized throughout his career for his contributions to designing and implementing enterprise-wide solutions using Microsoft Windows-based technologies.

Improving Collaboration with Microsoft Office Communications Server

by Jim Varner

Microsoft Office Communications (MOC) Server is the logical next step in Microsoft's evolution. MOC integrates instant messaging, voice, and other collaboration features into the existing Microsoft suite. By providing this functionality, Microsoft is making it easier for end users to work more efficiently and communicate directly and quickly with the right people.

One of the key design considerations that went into MOC is both simple and profound. MOC was designed on the concept that we communicate with people not "dial a device". Building on that idea, MOC lets us collaborate with people who are available via the communication method that they choose. These features are readily available to businesses of all types and sizes.

There are two versions of MOC, Standard and Enterprise. MOC Standard is designed for companies with a single location and fewer than 1000 employees. As a result, most of the features require a single server. In the Enterprise edition, the different functions need multiple servers, similar to the architecture of Microsoft Exchange 2008. This article will outline the available features of both the Standard and Enterprise editions, and suggest how they may be useful to your business.

MOC Standard Edition

MOC is designed to provide instant messaging, conferencing, presence information, and voice capability to end users. Although it is possible to use this product for internal communication only, you can see greater returns by integrating MOC with people outside your network. Imagine how much simpler life would be if your salespeople, external project managers, and Help desk contacts were available in a chat window at your convenience.

The main difference between MOC and free Internet-based chat systems is the ability to log all your chat sessions using an Archiving and CDR server. This is important for any organization that is concerned with regulatory compliance. You can also integrate Presence information, which tells you the best method for contacting someone at any given time, and provides a "click to dial" feature via Microsoft Outlook.

Setting up messaging internally is a simple deployment, which can be handled without additional hardware or software. MOC includes a Communicator Client, which offers both Instant Messaging and Presence functionality. However, if you need to use instant messaging outside the network, a Microsoft Access Edge Server must be deployed.

Internal Web and A/V conferencing is also part of the MOC Server Standard Edition, and can be run on the main hardware platform. This may seem like a frivolous feature, but the time savings can add up, even in a small company. Before MOC, users had to print documents and take them around the office for meetings and discussions. Now, with just a few mouse clicks, you can share your desktop and even video with anyone at a moments notice.

Presence displays your availability with a simple red/yellow/green indicator light and integrates with your Outlook calendar. So, when you are in meetings or on scheduled phone calls, your status will automatically be set to away. Users can still send you an instant message while your status is red, which you can reply to when you return.

The final feature of MOC is voice integration. Using the MOC software, users can control their incoming calls and even set up voice and video conferences on the fly. A Microsoft soft-phone can be used in place of a regular handset but is not required. MOC can interface with your existing phone system and allow users to control their communications through a familiar Microsoft interface.

MOC Enterprise Edition

MOC Enterprise Edition is designed for larger companies with multiple locations. This version of the product can be deployed as a standalone system or integrated with an existing PBX. Because most companies have existing phone systems, integration will be the most common of these two deployment methods.

Integrating either edition of MOC requires some of the functions to be handled by additional servers. The Mediation Server and Media Gateway

Appliance are the two basic starting points of any integration. The Mediation Server allows MOC to communicate via SIP with either a Media Gateway Appliance or a telecomm service that offers SIP trunking. The Media Gateway Appliance allows you to communicate with PSTN providers who don't offer SIP trunking as an option.

Several other features of MOC Enterprise Edition require their own hardware as well. If you need to provide Web Conferencing services, you will need both a Web Conference and Web Components Server. These servers together can provide internal and external Web conferencing as well as access to the corporate address book.

One requirement that drives this level of collaboration is the need to access not just data but people from any location at any time. Most users want access to the corporate address book from any location. Although MOC Standard Edition offers this functionality with additional hardware, the Enterprise Edition is designed to integrate multiple address books from many locations. A Microsoft Web Components Server is required to provide this access.

Other features including A/V Conferencing, Message Archiving, Call Detail Recording, Clientless Presence, and Messaging require their own servers as well. By breaking out these features, Microsoft has allowed companies to scale their deployments and use only the components that fit their business.

The A/V Conferencing Server allows users to place calls and give A/V presentations from their desktops. This server also controls A/V sessions and allows admins to specify media ports and conference properties. Although this is not a replacement for WebEx or other services, MOC A/V Conferencing provides basic collaboration functions and lays the groundwork for a more robust solution as the product matures.

Archiving messaging conversations and phone call details is possible in both the Standard and Enterprise editions of MOC. However, these functions require an Archiving and CDR server. This server provides a central location for all messages and logs, which allows you to easily access and sort conversations via a time/date/user stamp. Conversations and call details are stored and accessed via a SQL database that can reside on the CDR server or in a separate location.

In some situations, you may not want to install a MOC client on a PC. Both presence information as well as messaging can be provided with a browser-based client. However, this requires a Communicator

Web Access Server. This server is specifically designed to provide access to messaging, call management, and desktop sharing capabilities without anything more than a browser and an Internet connection.

The final component required to provide services outside of the corporate network is the Access Edge Server. This server allows external users to communicate with any MOC server on the network. For example, if your organization needs easy access to contracted Help desk personnel, this server will provide the connectivity.

Thus, the MOC Enterprise Edition provides the same features and functions as the Standard Edition. However, because of its modular architecture, the Enterprise Edition can scale to meet the needs of larger organizations. Now that we have a basic understanding of MOC, let's discuss how it might fit in certain scenarios, and where you can go to get more detailed information.

MOC Server Scenarios

The level of collaboration required in today's environment continues to grow and become more complex. IT administrators and directors need instant access to their vendors and partners, while other internal users need to share documents and ideas from any location at any time. There are several features in MOC that allow this level of collaboration and at the same time let you take advantage of your existing investments.

Many companies currently use, or are considering the purchase of, collaboration software such as WebEx or Adobe Acrobat Connect. These services are well established and provide desktop sharing and voice connectivity using a hosted model. As a result of their maturity and feature richness, they can be quite expensive. Companies considering these products should examine a MOC server and the collaboration tools it can provide. Conferencing and desktop collaboration may not be adopted fast enough to justify investing in a service. With MOC, you can enable these features within your organization, and expand them slowly as the adoption rate increases.

Instant messaging is another technology that is rapidly gaining market share. When you need help in the current environment, it usually involves several phone calls, long waits on hold, and possibly even voicemail. With MOC instant messaging, both your internal and external Help desk personnel can respond instantly to your questions and get you the help you need, when you need it.

The messaging feature is even more beneficial when coupled with Presence. If you can view a list of your Help desk personnel or even your vendor reps who are currently available and the best method to communicate with them, you can avoid the hassles of phone tag and trading emails. This level of collaboration results in less downtime and less wasted time while you and your users wait for the information you need.

Finally, managing the client software and agents for collaboration services can be a nightmare. Certainly we have all spent time talking a salesperson through a VPN client installation while they are on the road. With MOC, access to your collaboration services can be Web-enabled so that your road warriors don't spend time trying to install and configure client software. This level of access and ease of deployment are all part of the MOC suite.

Resources

So whether you are supporting a robust IP Telephony deployment already or building a new communications system for your employer, MOC should be investigated as a possible solution. Microsoft has provided many tools to educate and assist with collaboration planning and rollouts.

Obviously, planning is the most important step in any project. With that in mind, Microsoft has provided the MOC Server Planning guide to take you through this process. This guide will walk you through every step of planning and deploying a MOC Server, and is available for download.

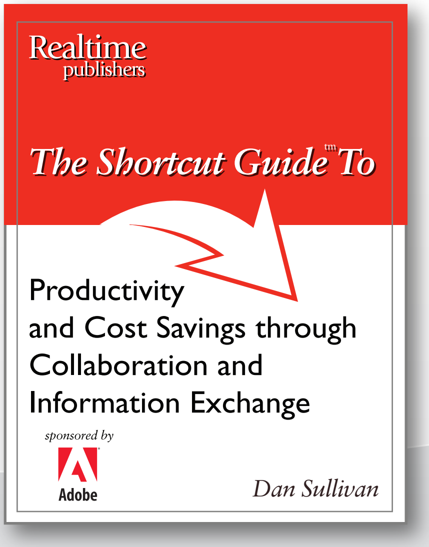
Another excellent source of information is the OCS R2 Enterprise Telephony Integration white paper. This white paper explains Microsoft's approach to design and helps you understand deployment methods and product limitations. The OCS R2 Enterprise Telephony Integration white paper can be downloaded.

Integrating MOC Enterprise Voice services can be a little intimidating, as this is a new approach to communications. Many companies are reluctant to include third-party software in enterprise voice deployments. To reduce the risk and take you through the complexities of enterprise voice, the Enterprise Voice Planning and Deployment Guide is available.

One final point: MOC has been built on open standards, so there are many hardware vendors in the market claiming that their hardware is certified for use with MOC. However, many vendors have elected to forego the Microsoft certification process. In order to insure that your hardware is fully compatible with MOC and supported by Microsoft, you should search the list.

Although Microsoft is relatively new in the VoIP space, they have years of experience providing collaboration software. MOC can be used to leverage your existing investment and provide a higher level of productivity across your organization. It's definitely worth investigating if voice, video, and messaging are part of your corporate growth plan.◆

Jim Varner is a technology consultant with more than 14 years of exeprience. Jim has been involved with the design, implementation, and support of complex environments for Lockheed Martin, Kroenke Sports, Great West Life, and numerous others. He has contributed to several publications including Redmond Magazine and The Northern Colorado Business Report and has written chapters for several software installation guides. Jim holds certifications in Citrix, EMC, VMware, and Cisco, including the Cisco Lifecycle Management Certification (Prepare, Plan, Design, Implement, Operate, and Optimize).


Realtime publishers
The Shortcut Guide To™
Productivity and Cost Savings through Collaboration and Information Exchange
sponsored by
Adobe
Dan Sullivan

Dealing with Multi-Vendor Logs

by Jan Kanclirz Jr.

You set it and forget. Many of us are guilty of it—we configure our network devices, servers, databases, firewalls, intrusion detection system (IDS), and just about any networked device out there that sits in our company environment to send logs over to some type of central repository. But then what? Well, we usually forget about it or just give up in the process of trying to piece individual raw log entries together out of millions of lines from a flat syslog file.

The challenges created by this situation are many:

- Variety of devices generating logs—The number of network and system devices is on the rise: In response to increased technology needs and security threats, we are deploying more and more security devices.
- Volume of data—A fair-size company can generate terabytes (TB) of raw logs each month, if not each week. Storage requirements and the need to maneuver through these enormous log files become unmanageable and time consuming on already overworked staff.
- Regulatory requirements—Over the past few years, governmental and industry regulations flooded the market with regulations. PCI, HIPAA, and SOX, just to name a few, demand and hold us legally accountable to protect information and prove that our security measures are adequate.

Army of Soldiers and One General

The army of networked devices continues to grow, which we are responsible to monitor and protect. Luckily, they all can speak a common language format, which we can interpret and store. This common format is called the syslog protocol. Syslog is a client/server logging protocol that is most likely found riding on top of UDP port 514. Having this type of standard helps this enormous army of networked devices to communicate back to the General in a common way.

A correlation event engine is just that—"the General." It is an application that is capable of receiving raw syslog data from a variety of devices and is able to store each event in organized fashion and correlate between raw log events—piece them together, if you will. Imagine that your external IDS detected an attack, your firewall access list allowed it through, your internal routers routed it, your switch port switched it, and your Windows server accepted it on HTTP port, but because your Windows server had an updated patch level, the attack was not effective. A correlation event engine, "the General," can receive any type of raw syslog data from all these vendor-agnostic devices, correlate each of these events, and assign priorities based on your corporate security policy. So, the next time that high HTTP vulnerability comes along, you can ignore it or at least not worry about it as much because it really isn't effective down the line. Sound cool? It really is. Many vendors, such as RSA and Cisco, have been providing such solutions for some time now.

Correlation engines come with pre-defined templates to support syslog messages from a variety of vendor devices. The good news is that if your vendor or level of code isn't pre-defined, you have the capability of creating your own templates to support any syslog formats.

Volume

Storing large amounts of data gets very expensive very quickly—not to mention the organizational challenge of storing millions of lines of raw log data in a readable format. Size your correlation engine based on number of events per second and your retention storage requirements. Some correlation engines support the capability to interconnect external storage; this functionality comes in very handy if you are required to keep usable data for a long time and offers the improved availability built-in to external storage.

The number of syslog events or flows also matters and will determine the type of device you will need to purchase. Some correlation engines can scale by dividing multiple Generals into multiple site locations and splitting the effort of storing and processing events. These location Generals are then managed by the chief management General that interconnects each location, providing a central management solution.

Ask your vendor for an Events per Second (EpS) storage calculator to calculate your storage requirements. The following example is a sample calculator used to scale Cisco's MARS:

1000 EPS with 30days of retention will require about 907GB of disk space.

The size of your correlation engine will depend on the number of EpS and your retention requirement, which will be dictated by your security policy and compliance needs. An additional consideration is that you might want to put an engine in each geographical location to avoid sending millions of logs over pricy WAN connectivity. You can then have your centralized manager manage them all.

Regulatory Compliance

Regulatory compliance is a very critical requirement for many businesses. Regulations such as PCI, SOX, and HIPAA are forcing corporations to properly secure, store, and manage sensitive information. A correlation engine is an auditor's dream. Correlation engines can help with secure storage and management of logs. Vendors provide readily-available SOX, PCI, and HIPPA templates as well as custom queries that auditors can use to prove an organization's compliance with many different requirements.

A discussion of all regulatory compliance is out of the scope of this brief article; let's take a peek at one of the most recognized—PCI compliance—and how a correlation engine can help. Compliance with this standard is necessary for all customers who process, transmit, or store credit card information. The PCI Data Security Standard (DSS) is made up of 12 key areas that are then sub-divided into more than 100 categories. Let's take a look at some of these requirements and how a correlation engine can ensure compliance is clear to an auditor:

- Install and maintain a firewall configuration to protect data—The auditor is looking for all activity and any user changes to firewalls within the environment. Every change or activity on a firewall is sent via raw log into the correlation engine, which can then report on it.
- Track and monitor all access to network resources and cardholder data—The auditor is looking for proof of ongoing monitoring of the cardholder database, its access, and any modifications of devices, including wireless. Again, all of these changes, such as Oracle raw logs, are logged and sent to the correlation engine, which stores and creates reports as needed.
- Do not use vendor-supplied defaults for system passwords and other security parameters—Correlation engines can interface and receive reports from vulnerability scanners, such as those from McAfee and Qualys, and correlate and report on settings such as default system passwords and other security vulnerabilities such as patch levels.

Last Word

Without a doubt, when configured correctly, a correlation engine helps organizations with compliance, better security, and, most of all, efficiency. Your administrators' time is wasted if they are looking at IDS logs, firewall logs, vulnerability logs, and patch level logs individually, trying to piece it all together. In already lean economic times, it makes sense to invest in a product that helps address your organization's security, efficiency, and compliance needs.◆

Jan Kanclirz Jr. (CCIE #12136-Security, CISSP, RSA CSP, CCSP, CCNP, CCIP, CCNA, CCDA, INFOSEC Professional, Cisco WLAN Support/Design Specialist, DCASI, DCASD) is currently a Senior Network Information Security Architect at MSN Communications. Jan specializes in multi-vendor designs and post-sale implementations for several technologies, such as VPNs, IPS/IDS, LAN/WAN, firewalls, content networking, wireless, and VoIP. Beyond network designs and engineering, Jan's background includes extensive experience with open source applications and Linux. Jan has contributed to several Syngress book titles on topics such as wireless, VoIP, security, operating systems, and other technologies.

Practical PowerShell

Giving You an Out

by Jeffery Hicks

You can download a zip file with all these scripts from http://www.realtime-windowsserver.com/code/v2n6_Practical_PowerShell.zip.

Windows PowerShell offers a number of cmdlets such as Out-File and Out-Printer that are useful when you need to save the output of a PowerShell expression.

```
PS C:\ get-service | where {$_.status -eq “Running”} | Out-file services.txt
```

This will create a text file containing the pipelined output. But maybe you’d prefer a different format, say Microsoft Word? You could use an expression like this:

```
PS C:\ get-service | where {$_.status -eq “Running”} | Out-file services.doc
```

But when you go to open the file, you’ll likely have to configure Word to properly format the document. I thought it might be more useful to have a tool for capturing output to a real Microsoft Word document; something that would allow you to create colorized output and perhaps with a footer. The function I developed is called Out-MSWord. I suppose I should mention that Microsoft Word must be installed in order for this function to work. I tested it with Microsoft Word 2007, but I don’t see why it won’t work with more recent versions as well. The function has been tested with PowerShell v1.0 and the latest PowerShell v2.0 CTP; although I can’t guarantee compatibility with future CTP releases. You can download a script file with the function and examples here.

```
Function Out-MSWord {
 Param ([string]$filepath,
        [switch]$tee,
        [switch]$append,
        [switch]$noclobber,
        [string]$font=”Consolas”,
        [double]$fontsize=8,
        [string]$fontcolor=”Auto”,
        [string]$footerfont=”Consolas”,
        [double]$footersize=8)
 BEGIN{
  #define some MS Word variables
  $wdSeekMainDocument = 0
  $wdSeekPrimaryFooter = 4
  $wdSeekPrimaryHeader = 1
  $wdAlignPageNumberCenter =  1
  $wdAlignPageNumberInside =  3
  $wdAlignPageNumberLeft = 0
  $wdAlignPageNumberOutside = 4
  $wdAlignPageNumberRight =  2

  $wdStory = 6

  $Auto = 0
  $Black =  1
  $Blue = 16711680
  $Green =  32768
  $Red = 255
  $Teal = 8421376
  $Violet = 8388736
  $Yellow = 32896

  #select font color
  switch ($fontcolor) {
    “Auto” {$color=$Auto}
    “Black” {$color=$Black }
    “Blue” {$color=$Blue }
    “Green” {$color=$Green }
    “Red” {$color=$Red }
    “Teal” {$color=$Teal }
    “Violet” {$color=$Violet }
    “Yellow” {$color=$Yellow }

    default {
      Write-Warning “Invalid color choice: $fontcolor. Using Default. Valid color choices
are: Black,Blue,Green,Red,Teal,Violet and Yellow.”
      $color=$wdAuto}
  } #end Switch

  #create the MS Word COM object
  $word=New-Object -ComObject “Word.Application”
  #get document if -append
  if ($append)
  {
    #verify file exists and if so, open it
    if ((Get-Item $filepath -ea “silentlycontinue”).Exists)
    {
        $doc=$word.documents.open($filepath)
        $blnNewFile=$False
        #Write-Host “Appending to $filepath” -foregroundcolor Cyan

    }
    else
    {
        #you asked to append to a file that doesn’t exist so create a new one
        $doc=$word.Documents.add()
        $blnNewfile=$True
        #Write-Host “-Append called but file doesn’t exist. Creating $filepath”
-foregroundcolor Cyan
    }
  }
  else
  {
    #create a new document
    $doc=$word.Documents.add()
    $blnNewFile=$True
    #Write-Host “Creating $filepath” -foregroundcolor Cyan

   }

  $selection=$word.Selection

  #get the footer
  $doc.ActiveWindow.ActivePane.view.SeekView=$wdSeekPrimaryFooter

  #set the footer
  $printed=(“printed {0}” -f (Get-Date))
  $selection.HeaderFooter.Range.Text=$printed
  #add page numbering
  $selection.HeaderFooter.PageNumbers.Add($wdAlignPageNumberRight) | Out-Null

  #get the footer and format font
  $footers=$doc.Sections.Last.Footers
  foreach ($footer in $footers) {
    if ($footer.exists) {
       $footer.range.font.name=$footerfont
       $footer.range.font.size=$footersize
       }
   } #end Foreach

   #return focus to main document
   $doc.ActiveWindow.ActivePane.view.SeekView=$wdSeekMainDocument

  #initialize an array to hold incoming objects
  $data=@()
  } #end BEGIN scriptblock

 PROCESS {
   #save incoming objects to a variable
   $data+=$_

   #write piped object is -Tee
   if ($tee)
   {
    write $_
   }

 } #end PROCESS scriptblock

END {
     #convert data to string
    $text=$data | Out-String
   #only write the file if $text exists
```

```
    if ($text) {
      if (!$blnNewFile)  #the file has been opened before
      {
        #jump to the end
        $selection.Endkey($wdStory) | Out-Null
        #insert blank lines
        for ($i=1;$i -lt 4;$i++) {
            $selection.TypeParagraph()
        }
       } #end if !$blnNewFile

       #set font and paragraph settings
       $selection.font.name=$font
       $selection.font.size=$fontsize
       $selection.font.color=$color
       $selection.paragraphFormat.SpaceBefore = 0
       $selection.paragraphFormat.SpaceAfter = 0
```

```
#add the text to the document
$selection.TypeText($text.Trim())

if ($filepath)
 {
  #don’t overwrite if -noclobber was specified
  if ($noclobber)
  {
      #does file exist?
      if ((Get-Item $filepath -ea “silentlycontinue”).exists)
      {
          Write-Warning “-NoClobber specified and $filepath exists”
          #show the document so you can review and/or save
          $word.visible=$True
      } #end if item exists
      else
      {
          #check PowerShell version because v1.0 requires special
          #handling
          if ($host.version.major -eq 1)
          {
              $doc.SaveAs([ref]$filepath)
          }
          else
          {
              $doc.SaveAs($filepath)

          }
          $word.quit()
      }
  } #end if $noclobber
  else
  {
      #check PowerShell version because v1.0 requires special
      #handling
      if ($host.version.major -eq 1)
      {
          $doc.SaveAs([ref]$filepath)
      }
```

```
           else
           {
               $doc.SaveAs($filepath)
           }
           $word.quit()
       }
      } #end if $filepath
      else #no filepath
      {
       #jump to the beginning of the document
       $selection.Homekey($wdStory) | Out-Null

       #show the document so you can review and/or save
       $word.visible=$True
     }
  } #end if $text
  else
  {
  Write-Warning “No data”
    #turn off alerts
    $word.displayAlerts=0
    if ($host.version.major -eq 1 )
    {
        #close without saving
        $doc.close([ref]$Word.WdSaveOptions.wdDoNotSaveChanges)
     }
    else
    {
        #close without saving
        $doc.close($Word.WdSaveOptions.wdDoNotSaveChanges)
    }
    #exit Microsoft Word
   $word.quit()
  }
 } #end END scriptblock
} #end Function
```

After you load the function into your PowerShell session, I find it helpful to also create an alias:

```
PS C:\> Set-Alias ow Out-MSWord
```

The function uses the Begin/Process/End script blocks, so you can pipe objects to it:

```
PS C:\ ps | where {$_.workingset -gt 50MB} | ow
```

When you run this command, a Microsoft Word document will open with the output from **Get-Process**. You can manually save the file if you want. Notice the footer displays the date and time the command was run as well as page numbering. These are hard coded into the function, but you can change them of course.

The function can be customized by specifying a number of parameters, which are outlined in Table 1.

Parameter	**Description**
-Filepath	A file name used to save the document
-Tee	If specified, the output will also be displayed to the console
-Append	Append output to an existing file; if the file doesn't exist, it will be created
-NoClobber	Don't overwrite existing files
-Font	The font used in the document's main body; the default is Consolas
-FontSize	The font size using the document's main body; the default is 8
-FontColor	The font color used in the document's main body; valid color choices are: Black, Blue, Green, Red, Teal, Violet, and Yellow, and the default is Black
-FooterFont	The font used in the footer; the default is Consolas
-FooterSize	The font size used in the footer; the default is 8

Table 1: Parameters you can use to customize the script.

Here's the previous command customized:

```
PS C:\ ps | where {$_.workingset -gt 50MB} | ow c:\data\bigprocs.doc -tee -noclobber
-fontcolor Red
```

The parameters are positional or you can use the parameter names. For the switch parameters like –Tee and –Append, you have no choice.

This function demonstrates how to use COM objects within Windows PowerShell as well as how to create a function that accepts pipelined input. Let me show you.

First, the BEGIN script block is executed before any pipelined objects are processed. I use this script block to set up the Word document. The Microsoft Word COM model relies heavily on variables. I find it easier to simply define the ones I need:

```
BEGIN{
  #define some MS Word variables
  $wdSeekMainDocument = 0
  $wdSeekPrimaryFooter = 4
  $wdSeekPrimaryHeader = 1
  $wdAlignPageNumberCenter =  1
  $wdAlignPageNumberInside =  3
  $wdAlignPageNumberLeft = 0
  $wdAlignPageNumberOutside = 4
  $wdAlignPageNumberRight =  2

  $wdStory = 6

  $Auto = 0
  $Black =  1
  $Blue = 16711680
  $Green =  32768
  $Red = 255
  $Teal = 8421376
  $Violet = 8388736
  $Yellow = 32896

Next, I need to determine what font color to use. This is accomplished with a Switch
construct:
  switch ($fontcolor) {
    “Auto” {$color=$Auto}
    “Black” {$color=$Black }
    “Blue” {$color=$Blue }
    “Green” {$color=$Green }
    “Red” {$color=$Red }
    “Teal” {$color=$Teal }
    “Violet” {$color=$Violet }
    “Yellow” {$color=$Yellow }

    default {
      Write-Warning “Invalid color choice: $fontcolor. Using Default. Valid color choices
are: Black,Blue,Green,Red,Teal,Violet and Yellow.”
      $color=$wdAuto}
  } #end Switch
```

If an invalid color was specified, a warning message is displayed and the document uses the Auto color, which is essentially Black. Now it’s time to create the Microsoft Word COM object:

```
  #create the MS Word COM object
  $word=New-Object -ComObject “Word.Application”
```

If –Append was specified, the function checks to see whether the file exists and if so, opens it:

```
if ($append)
{
  #verify file exists and if so, open it
  if ((Get-Item $filepath -ea “silentlycontinue”).Exists)
  {
      $doc=$word.documents.open($filepath)
```

Otherwise, Microsoft Word creates a new document:

```
      $doc=$word.Documents.add()
```

This same expression is also used to create a new document. Now it’s time to define the footer. First I need to give it focus:

```
$selection=$word.Selection

#get the footer
$doc.ActiveWindow.ActivePane.view.SeekView=$wdSeekPrimaryFooter
```

Then I can set the text to display:

```
#set the footer
$printed=(“printed {0}” -f (Get-Date))
$selection.HeaderFooter.Range.Text=$printed
```

Followed by adding page numbering:

```
$selection.HeaderFooter.PageNumbers.Add($wdAlignPageNumberRight) | Out-Null
```

Lastly, I enumerate the footers and set the footer font and size.

```
#get the footer and format font
$footers=$doc.Sections.Last.Footers
foreach ($footer in $footers) {
  if ($footer.exists) {
     $footer.range.font.name=$footerfont
     $footer.range.font.size=$footersize
     }
 } #end Foreach
```

It’s possible to have different footers on different pages. This code gets the proper formatting accomplished.

The BEGIN script block wraps up by returning focus to the main document:

```
   $doc.ActiveWindow.ActivePane.view.SeekView=$wdSeekMainDocument
```

Then initializes an array to hold incoming objects:

```
  $data=@()
```

The reason for this is that I don't want to create a new Word document for every object. I want to write all the pipelined objects at once. So, during the Process script block, each pipelined object is added to the array:

```
PROCESS {
   #save incoming objects to a variable
   $data+=$_
```

This is also where each pipeline object is written to the console if –Tee was specified:

```
   if ($tee)
   {
    write $_
   }
```

After all the pipeline input has been processed, the END script block is executed. The first step is to convert the objects in $data to strings:

```
END {
     #convert data to string
    $text=$data | Out-String
```

Now it's possible that there will be no data. If that's the case, a warning message is displayed and the Word document that had been created is closed without saving:

```
  Write-Warning "No data"
    #turn off alerts
    $word.displayAlerts=0
    if ($host.version.major -eq 1 )
    {
        #close without saving
        $doc.close([ref]$Word.WdSaveOptions.wdDoNotSaveChanges)
     }
    else
    {
        #close without saving
        $doc.close($Word.WdSaveOptions.wdDoNotSaveChanges)
    }
```

But assuming data does exist, the function needs to know where to insert data, especially if appending data. In that case, the selection jumps to the end of the file and inserts a few blank lines:

```
if ($text) {
     if (!$blnNewFile)  #the file has been opened before
     {
       #jump to the end
       $selection.Endkey($wdStory) | Out-Null
       #insert blank lines
       for ($i=1;$i -lt 4;$i++) {
           $selection.TypeParagraph()
       }
      } #end if !$blnNewFile
```

Now that the insertion point is selected, I define the font size, color, and paragraph spacing:

```
      #set font and paragraph settings
      $selection.font.name=$font
      $selection.font.size=$fontsize
      $selection.font.color=$color
      $selection.paragraphFormat.SpaceBefore = 0
      $selection.paragraphFormat.SpaceAfter = 0
```

Finally, the data is written to the file:

```
     #add the text to the document
     $selection.TypeText($text.Trim())
```

At this point, all that is left is to close up the document and quit Microsoft Word. However, I need to accommodate the –NoClobber parameter if specified. If the file exists, Word will display the document, so you can manually save it with a different name if you wish:

```
     if ($filepath)
      {
       #don’t overwrite if -noclobber was specified
       if ($noclobber)
       {
           #does file exist?
           if ((Get-Item $filepath -ea “silentlycontinue”).exists)
           {
               Write-Warning “-NoClobber specified and $filepath exists”
               #show the document so you can review and/or save
               $word.visible=$True
           } #end if item exists
```

Otherwise, I invoke the SaveAs() method using the specified file name:

```
            else
            {
                #check PowerShell version because v1.0 requires
                #specialhandling
                if ($host.version.major -eq 1)
                {
                    $doc.SaveAs([ref]$filepath)
                }
                else
                {
                    $doc.SaveAs($filepath)

                }
                $word.quit()
            }
        } #end if $noclobber
```

If –NoClobber was not specified and a filepath was specified, the function uses essentially the same code to save the file. If no filepath is specified, the file won't be saved, so I jump to the beginning of the document and make Microsoft Word visible:

```
       {
        #jump to the beginning of the document
        $selection.Homekey($wdStory) | Out-Null

        #show the document so you can review and/or save
        $word.visible=$True
```

The only time you'll see a Microsoft Word window is if you do not specify a file path. Otherwise, the function open and closes Microsoft Word as necessary "behind the scenes."

Let me wrap up this month with some other examples for you to try. Here's another basic example:

```
PS C:\>  gsv | where {$_.status -eq "running"} | Out-MSWord
```

A modification that saves the results to a file:

```
PS C:\>  get-service | sort status,displayname | ow c:\test\my.doc
```

Assuming the file already exists, my3.doc won't be overwritten:

```
PS C:\> gwmi win32_bios | ow -filepath "c:\test\my3.doc" -noclobber
```

Assuming the file already exists, this will append to it. You may need to experiment with different fonts and sizes:

```
PS C:\> ps | sort workingset | ow c:\test\my3.doc -append -fontsize 9
```

You aren't limited to using the function in a simple one-line expression. Here's a more complex example that gets all services and groups them by status. The grouped object is then enumerated by a For construct that evaluates the name of each group with a Switch construct. If the group name is "Running," the service objects are written to a file in green text. If the group name is "Stopped," the objects are written in red. Any remaining objects are written in yellow:

```
$g=gsv | sort status | group status
for ($i=0;$i -lt $g.count;$i++) {
  switch ($g[$i].name) {
    “Running” {
          $g[$i].Group | ow c:\test\svc.doc -fontcolor Green -append
    }
    “Stopped” {
          $g[$i].Group | ow c:\test\svc.doc -fontcolor Red -append
    }
  Default {
      $g[$i].Group | ow c:\test\svc.doc -fontcolor yellow -append
     }
  }
}
```

Here's one last example that first gets all running processes and saves them to a variable. The variable is then filtered three times, writing output to a Microsoft Word document with a different font color depending on the workingset size:

```
$p=ps | sort workingset -desc
$p | where {$_.workingset -gt 100mb} | ow c:\test\ps.doc -fontcolor red -font “Lucida
Console” -fontsize 10.5
$p | where {$_.workingset -lt 100mb -and $_.workingset -gt 25mb} | ow c:\test\ps.doc
-fontcolor yellow -font “Lucida Console” -fontsize 10.5 -append
$p | where {$_.workingset -lt 25mb} | ow c:\test\ps.doc -fontcolor green -font “Lucida
Console” -fontsize 10.5 -append
```

If you have problems getting the function to work or would like to enhance it, I'll be more than happy to help in the PowerShell forum at ScriptingAnswers.com.

Jeffery Hicks (MCSE, MCSA, MCT) is a Microsoft PowerShell MVP and Scripting Guru for SAPIEN Technologies. Jeff is a 17 year IT veteran specializing in administrative scripting and automation. Jeff is an active blogger, author, trainer and conference presenter. His latest book is Managing Active Directory with Windows PowerShell: TFM (SAPIEN Press). Follow Jeff at Twitter.com/JeffHicks and blog.sapien.com. You can contact Jeff at jhicks@sapien.com.

Exclusively Exchange

Exchange 2010 Databases and High Availability Adjustments

by J. Peter Bruzzese

Well, the Beta of Exchange 2010 was finally released in April 2009, and we have a chance to get our first glimpse into the new features. Upon first glance, you may be surprised to see that the overall interface hasn't changed from the Exchange Management Console (EMC) of Exchange 2007. That is not a bad thing, actually. There isn't always a reason to change the entire look of our management tools and move things around so that we are absolutely lost with each new release of a product. There are some little tweaks to note and a major change in underlying storage architecture that is sure to catch your attention.

We will save the little tweaks and policy settings for later. Let's address the 600-pound elephant with the words 'Databases and High Availability' painted across its chest first.

Storage Groups Are Gone!?

This may be a big shock to those who recall the introduction of a 'storage group' with Exchange 2000, but after three released products (2000/2003/2007), the concept of a storage group has been pushed aside and instead the database itself becomes the management unit to focus on. Now, you may have seen this coming because consistently in Exchange 2007, the rules were to only put one database into one storage group; in fact, you couldn't perform high availability without it. More than one database to a storage group caused logs to be intertwined between the databases for that storage group (which was confusing, to say the least).

Another change in Exchange 2010 is that the database is not managed on a server level (as it was in Exchange 2007 under the Server Configuration branch of the EMC) but rather at the Organization Configuration Level under Mailbox.

> Note: The Server Configuration branch under Mailbox will show Database Copies.

The real change here in Exchange 2010 is how databases are viewed in Active Directory (AD). Now they are peer objects with Server objects. What this means is that, like servers, databases are now global objects and, as a result, the location for database management is performed under the Mailbox node of the Organization Configuration node (as opposed to under the Server Configuration node like you may be used to).

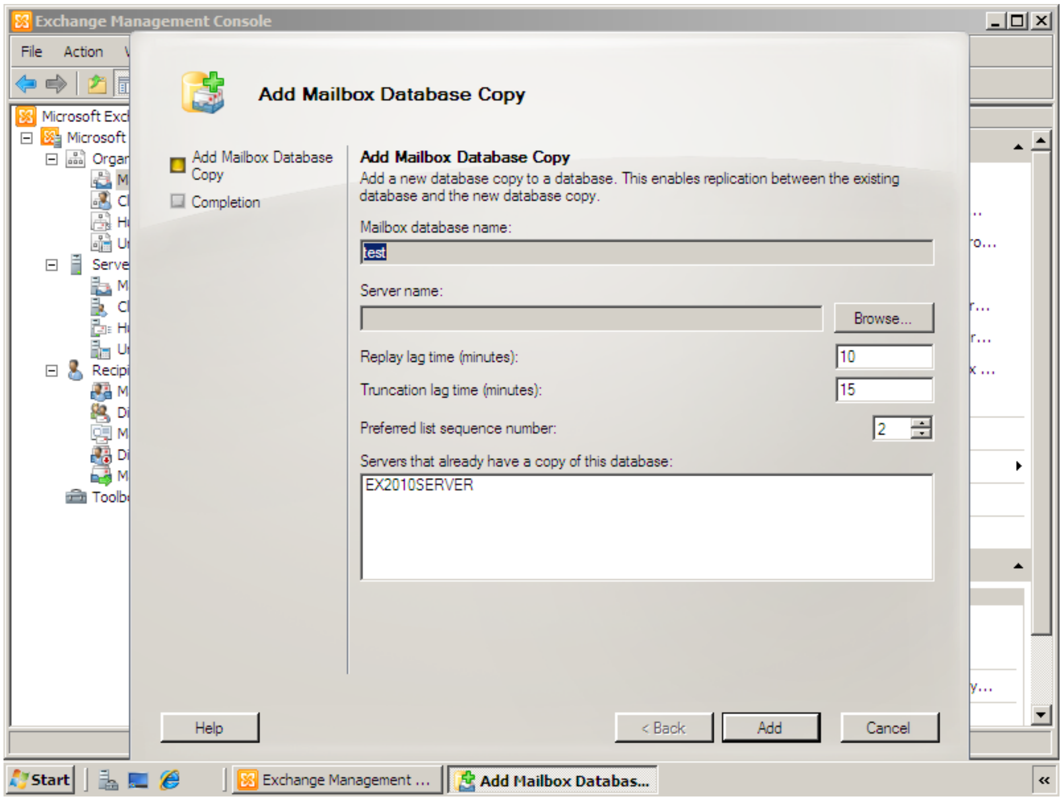
> Note: Initial reports regarding the new structure is that IOPS performance is much improved with 2010. Typically, an Exchange admin looks for expensive disks with fast spindles for storage, but the claim is that you can use lower-cost SATA drives or JBOD (Just a Bunch of Disks) storage—although it is important to note that JBOD is only recommended for very specific configurations; for example, an HA configuration with at least three mailbox database copies.

LCR, CCR, SCR, and SCC Are Gone—What?!

After you have read article after article (I know, I've written many of them that you may have read) regarding the new high-availability features in Exchange 2007—they are just gone!? Well, yes and no. Remember, that SCC was a carry-over from traditional clustering solutions for Exchange that used shared storage for the databases; these are no longer supported in Exchange 2010. LCR/CCR/SCR all use log shipping and replay to keep the databases up to date asynchronously (meaning the log file on the active node would still need to complete and close out before being shipped, providing a slight bit of data loss in the event of a crash). However, with CCR, you had clustering services and the transport dumpster to assist with failover and mitigating the amount of loss suffered in the event of a crash.

Now, with Exchange 2010, take the best parts of the 'continuous replication' solutions and mash them together into a new concept that works in conjunction with Database Availability Groups. DAGs allow for as many as 16 servers using some of the failover clustering components (such as heartbeats and the file share witness you might recall from CCR clusters) to connect all members of the DAG. These servers can span physical locations as well.

There is also the ability to lag the log replay during the configuration (see Figure 1) so that you can augment the time that a corruption or virus might hit and when it gets replayed (allowing you time to stop it), and you can perform point-in-time restores without going back to tape.



Add Mailbox Database Copy

Add Mailbox Database Copy

Add a new database copy to a database. This enables replication between the existing database and the new database copy.

Mailbox database name: test

Server name:

Replay lag time (minutes): 10

Truncation lag time (minutes): 15

Preferred list sequence number: 2

Servers that already have a copy of this database: EX2010SERVER

Figure 1: Note the replay and truncation time lag options in minutes.

One of the coolest aspects of DAGs is that you can decide, on the database level, where you want a database to be replicated towards. So, for example, you might have a server in Florida with three databases and replicate one database to a server in California, one to New York, and one to London. Or all three to each. Rather than using the concept of active/passive, the terms master and live master are used. The database that is 'live' is called the master and if something goes wrong, the Active Manager (which is a Client Access component that is watching the databases) will change the live master to another database. If a change is made, the Client Access Server is actually responsible for redirecting clients.

So, just to recap here: LCR and SCC are gone. CCR and SCR (which were used in Exchange 2007 to allow for recovery at the server and data center levels, respectively) have been combined into a new high-availability solution that works with DAGs, which use continuous replication to provide recovery from a failure at the server and data center levels.

One thing that still seems to be the same, however, is the need for you to be using Server 2008 Enterprise Edition (or Datacenter Edition). You may be wondering why 2008? Exchange 2010 only uses Server 2008, so your options are limited in that sense. The reason for the Enterprise (or Datacenter) Edition is that Windows Failover Clustering is still used. Now, it may be that some form of DAG will be available in the Standard Edition to equal what was allowed with LCR and Exchange 2007, but we will have to wait and see (so far that isn't the case).

One other interesting feature is that you can use the move-mailbox feature with 2010 to move the mailbox while users are still working. In a Vmotion/Live Migration sort of move, the user can continue to work (email send and receive) while you move the mailbox. Haven't seen this feature just yet, but I'm looking forward to looking at it and seeing if there are any glitches.

The changes in structure make life easier on a few other levels:

- Deployment of these solutions can be handled without a special installation of an active or passive mailbox server. In addition, you can install the other server roles with your mailbox server that supports DAG groups.
- You don't need to work through failover clustering services, so although you should still be aware of some of the concepts in clusters (and in previous high-availability options with Exchange 2007), you don't need to be a clustering expert to get this up and running.
- The failover aspects are automatic regardless of whether it is server/site failover. Under the 2007 options, you only had automatic failover with CCR and SCC.

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. His most recent book "Exchange 2007 How-To" was published by Sams in January 2009. 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.

ExclusivelyExchange.com Free Training Videos

Would you like to learn more about Exchange 2007 and 2010? Check out the free training videos at www.exclusivelyexchange.com. And if you want to learn about other subjects like SharePoint, Server Core, Hyper-V, and more…check out www.cliptraining.com.

Copyright Statement