Lunch Break
SERIES

Exchange Server 2010 Tips & Tricks

Learn best practices for high availablity, how to avoid archiving pitfalls and the best way to sync Windows phone with this free guide!

Redmond
Redmondmag.com

SPONSORED BY

yousendit™

RVP
REDMOND VENDOR PROFILE

Important Issues to Consider Before Migrating to a New Version of Exchange

A white paper from Osterman Research

Mail server management is a balancing act. IT must limit email storage and attachments to keep servers running, yet still give users an easy, secure way to share large documents and files.

In this white paper from Osterman Research, find out how to improve mail server performance and boost user productivity by integrating a secure, cloud-based file sharing solution into your Microsoft Exchange migration. By incorporating file-sharing services into your migration plans, you can offload IT staff from time-consuming mail server management, plus improve work processes—and user satisfaction.

However, not all cloud file-sharing solutions are the same. Find out why integration plays a key role, and how choosing a solution that works seamlessly with Outlook can support how users want to communicate, and reduce the burden on IT. The Osterman white paper is offered courtesy of YouSendIt, the first business content collaboration service to offer an integrated approach for sending, sharing and signing documents online. Download the white paper (https://www.yousendit.com/).

YouSendIt, Inc. is the first cloud collaboration service to offer an integrated approach for sending, sharing and signing documents online. With over 20 million registered users in 193 countries, YouSendIt helps enterprises and business professionals streamline collaboration by enabling them to instantly sync and access content in the cloud and easily send files, share folders, and sign documents from anywhere—the desktop, Web or mobile devices. YouSendIt offers a suite of productivity tools (http://www.yousendit.com/applications) that integrate seamlessly into common desktop and mobile environments. With YouSendIt, companies can alleviate ever-expanding email inboxes and overages, improving performance and productivity while reducing costs and IT security risks.

https://www.yousendit.com/

yousendit™

BY JAAP WESSELIUS

Exchange Server High Availability

E-mail is an essential business communication tool, so ensuring high availability with Microsoft Exchange Server is important.

There are several layers in Microsoft Exchange Server 2010 you can configure as a high-availability (HA) solution. The Database Availability Group offers HA on the Mailbox Server role. If you want a full HA solution, you can configure the Client Access Server and Hub Transport Server as an HA solution as well.

It's worth bearing in mind that Exchange Server 2010 Standard Edition also supports replication technologies, just like the Exchange Server 2010 Enterprise Edition. The only difference is that the Standard Edition only supports up to five databases per server, while the Enterprise Edition supports up to 100 databases per server. This is a perfect development for organizations that don't have thousands of mailboxes, and don't need to create a lot of Mailbox Databases.

Exchange Server Databases

Configuring HA for other server roles hasn't changed significantly since Exchange 2007. Exchange Server 2010 uses a database to store the primary data—the messages you send and receive. This database technology is a transactional system, which is pretty common, but Exchange Server uses its own technology built on the Extensible Storage Engine (ESE), sometimes referred to as a JET database.

When installing an Exchange Server 2010 Mailbox Server, the initial mailbox database is, by default, stored on the local C:\ drive; more specifically on C:\Program Files\Microsoft\Exchange Server\V14\Mailbox\Mailbox Database <<random number>>\. This random number is generated by Exchange Server during the initial configuration because the database names on Exchange 2010 and higher servers must be unique within the Exchange organization.

A number of files make up the Exchange 2007 database environment, all of which play a crucial role in the correct functioning of Exchange server:

- "mailbox database 0242942819.edb"
- E00.log
- E00000003a.log, E00000003b.log, E00000003c.log and so on
- E00.chk
- E00res00001.log and E00res00002.log
- E00tmp.log
- Tmp.edb

To understand Exchange database technology is to understand the flow of data between the Exchange Server and the database itself. Data is processed in 32KB blocks, also called "pages." When Exchange is finished processing such a page, it immediately writes it to a log file if it was updated. The page is still kept in memory until Exchange needs this memory again, but when the page isn't used for some time, or when Exchange needs to force an update during a checkpoint, the page is written to the database file. So, the data in the log files is always in advance of the data in the database.

This is an important step to remember when troubleshooting database issues.

As data is written to the database, a pointer called the checkpoint is updated to reflect the new or updated page that was written to the database. The checkpoint is stored in a special file called the checkpoint file. Exchange Server uses this to make sure it knows what data has been written to the database and what data is in the log files and not yet written to the database. So, in short:

1. Mail data is initially processed in memory and separated into pages.
2. Updated pages are written to the log file.
3. If pages are no longer needed by Exchange these pages are written to the database.
4. The checkpoint file is updated to reflect the new location of the checkpoint.

Extensible Storage Engine

The database engine Exchange Server uses is built on the ESE. The ESE exists in several flavors:

- ESE97 for Exchange Server 5.5
- ESE98 for Exchange Server 2000/2003
- ESENT for Active Directory
- ESE for Exchange Server 2007 and Exchange Server 2010

The ESE is a low-level database engine. This means it knows all about "base types," such as short, string, long, longlong, systime and so on. However, it has no knowledge of any structure or schema. The schema is defined by the Information Store in the application. This is in contrast to a relational database like Microsoft SQL Server, where all the database structures are just meta-data, or part of the database itself.

The ESE is optimized for handling large amounts of semi-structured data, as it's impossible for an Exchange Server to predict what kind of data will be received, how large the data will be or what attachments messages will have.

Log Files

When Exchange Server is working with a page, and that page's status changes from dirty to clean, the page is written to the log file almost immediately. Data held in memory is fast to access, but volatile. All it takes is a minor hiccup in the server, and data in memory is lost. When it's saved in the log file, the whole server could burn down, and as long as you keep the disk, you also keep the data.

Thankfully, saving to the log file is normally a matter of milliseconds. The log files are numbered internally, and this number (referred to as the lGeneration number) is used for identifying the log files, and for storing them on the disk when they're completely filled with data.

The current log file, or the "log file in use," is E00.log; while Exchange is filling this log file with data, a temporary E00tmp.log file is already created (or is in the process of being created) in the background. When the E00.log is eventually filled with data, it's saved under another name. The name is derived from the log file's prefix (E00, E01, E02 and so on) and the lGeneration number, which is a sequential hexadecimal notation.

For example, when the lGeneration number is 1, the E00.log is saved as E0000000001.log. Alternatively, the last time this process happened, the lGeneration number was 3E, so the log file was saved as E000000003E.log. Because the lGeneration number is a sequential number, we know that the next lGeneration number of the E00.log must be 3F, and the next time this log file rollover process takes place, the log file will be saved as E000000003F.log.

Although it's not directly visible, the lGeneration number is stored inside the log file, and can be checked by dumping the header information of the log file with the ESEUTIL utility. The first few lines of the log file's header should read something like:

```
Base name: E00
Log file: E00.log
lGeneration: 63 (0x3F)
Checkpoint: (0x3F,8,16)
```

The lGeneration number is listed on the third line, both in decimal and hexadecimal notation. Unfortunately, this is very confusing, and there will be a day that an Exchange administrator mixes up these notations and starts working with the wrong log file.

The "Lazy Writer" Mechanism

After the pages are written to the log file, they're kept in memory, thereby saving an expensive read from disk action when Exchange Server needs the page again. When the Mailbox Server needs that memory for other pages, or when the page stays in memory for a long time, it is written to the database file. This is also known as the "lazy writer mechanism."

A common misconception is that data is read from the log files and written to the database file, but this is not the case. It's written directly from memory to the database, and log files are only read in recovery scenarios, such as after an improper shutdown of the server. Under normal circumstances, the log files are 100 percent write, whereas the database is a random mix between read and write actions.

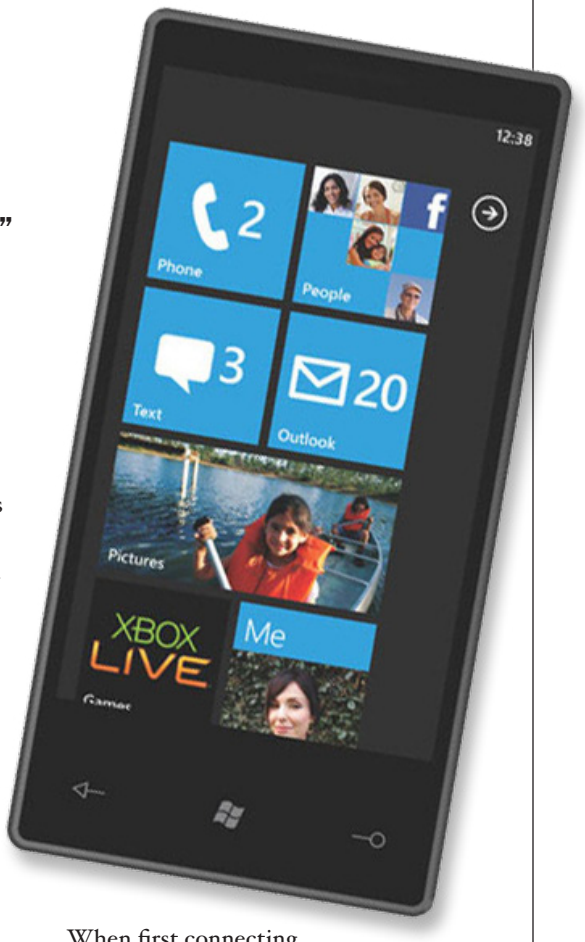
You can learn more about "Exchange 2010—A Practical Approach" (Red Gate Books, 2009) at red-gate.com/our-company/about/book-store. R

Jaap Wesselius is the founder of DM Consultants, a company with a strong focus on messaging and collaboration solutions. After working at Microsoft for eight years, Wesselius decided to commit more of his time to the Exchange community in the Netherlands, resulting in an Exchange Server MVP award in 2007. He is also a regular contributor at the Dutch Unified Communications User Group and a regular author for Simple-Talk.

BY BRIEN M. POSEY

Get Your Windows Phone 7 in Sync

The new Windows mobile platform, code-named "Mango," brings back some of the synchronization features important for enterprise use.



MICROSOFT

When Microsoft released Windows Mobile 6.1 and 6.5, it seemed like an earnest attempt to make Windows Mobile the enterprise-grade mobile computing platform of choice. With both of these versions, you could enroll devices in a domain and manage them with Group Policy.

Microsoft also introduced dozens of new ActiveSync mailbox policies in Exchange Server 2007. Many of these new policies you could only use with fully provisionable mobile devices. That meant for a while, Windows Mobile was the only mobile OS that would support them.

Then late last year, Microsoft released Windows Phone 7. With this new platform came a completely new philosophy. Microsoft removed many of the features you might find useful (if not essential) as an IT professional. The intent seemed to be to make the device more consumer-oriented.

One of the most surprising of these changes was the lack of support for ActiveSync mailbox policies. You can still connect Windows Phone 7 to Exchange by way of ActiveSync, but many of the ActiveSync mailbox policies that had previously received so much attention no longer work. Windows Mobile 6.1 supported 43 different ActiveSync mailbox policies when used with Exchange Server 2007 SP1 (read more about that at bit.ly/oSUPf5). However, Windows Phone 7 only truly supports seven of those policies. The same holds true when you use Windows Phone 7 with Exchange Server 2010 (read more about that at bit.ly/biZpOs).

Whither ActiveSync?

So what happened to the remaining ActiveSync mailbox policies? Some of them are supported to some degree, but not really. For example, you can still use the DisableRemovableStorage policy setting, but it will always return a value of True because Windows Phone 7 doesn't support using removable storage.

The lack of support for ActiveSync mailbox policies isn't the only area in which Windows Phone 7 is less suitable for enterprise use. Another sorely missed component is the certificates management console.

In Windows Mobile 6.1 and 6.5, Microsoft provided a console you could use to manage digital certificates. This console doesn't exist in Windows Phone 7. This is a problem because ActiveSync uses SSL encryption, which is based on certificate use. Windows Phone 7 devices have built-in support for certificates issued by well-known commercial certificate authorities (CAs), such as VeriSign or Go Daddy. However, certificate management can be a challenge if your organization is using its own enterprise CAs.

When first connecting my Windows Phone 7 device to Exchange through ActiveSync, the connection wouldn't work because my Windows Phone 7 device didn't trust my enterprise CA. To get around this, I set up a Hotmail account on my Windows Phone 7 device, e-mailed myself the necessary certificate, and installed the certificate by opening the e-mail attachment. Only then was I able to connect my Windows Phone 7 device to Exchange Server.

Besides the reduced ActiveSync support in Windows Phone 7, other enterprise features are missing as well. For example, you can't enroll a Windows Phone 7 device in a domain or manage it with Group Policy. There have already been a couple of updates, one of which added copy and paste

functionality. However, Microsoft has a major update, code-named "Mango," slated for release this fall.

Messaging with Mango

Mango, which will officially be known as Windows Phone 7.5, will add more than 500 new features to the Windows Phone platform. While many of these new features are consumer-oriented, Microsoft has also revealed a number of new features that IT professionals have been requesting.

A lot of the new features pertain to messaging. The original Windows Phone 7 included one major messaging improvement. Unlike previous versions of Windows Mobile, you could configure Windows Phone 7 to connect to multiple e-mail accounts. One of the ways Microsoft built onto this functionality in Mango is by allowing multiple Exchange ActiveSync connections from a single device. This new feature should prove to be helpful if you need to monitor multiple mailboxes.

time out after failing to connect. At that point, you often had to work through the advanced setup process. In Mango, however, the AutoDiscover service promises to greatly simplify the process of connecting Windows Phone 7 devices to Exchange Server mailboxes.

Of all the new messaging features, built-in support for Out of Office is a huge help. If you've ever been out of the office and forgotten to activate an Out of Office message, you'll be happy to know that Mango should solve this problem once and for all. You'll be able to turn Out of Office messages on or off directly from the phone.

The ability to search an Exchange server for messages that aren't saved on the phone is another helpful new feature. Windows Phone 7 stores a limited amount of mail locally on the device. On my own phone, for example, I store the messages from the last five days. Every once in a while, I need a message I've purged from my phone.

Another improvement to Windows Phone 7 messaging is that the phone now supports conversation view for e-mail messages. This gives you an experience that's more like what you're used to on the desktop.

Another improvement to Windows Phone 7 messaging is that the phone now supports conversation view for e-mail messages. This gives you an experience that's more like what you're used to on the desktop.

Support for the Microsoft Exchange AutoDiscover service is more important from an IT perspective. To connect a Windows Phone 7 device to an Exchange Mailbox, you had to enter the user's e-mail address and password. After doing so, the device would try to connect to the user's mailbox, but it would almost always

Previously, the only way to access a purged message was to temporarily reconfigure the phone to retain a longer e-mail history. In Mango, you'll be able to perform a mailbox search and retrieve messages directly from your Exchange mailbox, even if those messages are older than what would normally be saved on the phone.

"Pinnable" folders are another handy messaging feature. Windows Phone 7 has always let you access any of your mail folders (although some folders are not automatically synced), but the process for doing so was cumbersome. If you have a specific mail folder you frequently use, then Mango will let you pin that folder to the Start screen. This gives you one-touch access to the folder.

Security Upgrades

The original Windows Phone 7 release was a bit lacking when it came to security. While Windows Phone 7-based devices are not inherently insecure, there were many organizations that did not permit their use due to security concerns.

Microsoft has made several security improvements in Mango. The most urgently needed of these improvements is probably support for complex passwords. Windows Phone 7 lets you lock your device, but you're limited to a simple numeric PIN.

Mango will let you lock down your Windows Phone 7-based devices using complex passwords, and will control device password settings through Exchange ActiveSync policies. Some of the password-related policy settings Mango will support include:

- Password Required
- Password Expiration
- Password History
- Allow Simple Passwords
- Minimum Password Length

In addition to these password-related settings, you'll also be able to enforce device locks after periods of inactivity. In addition, you'll be able to automatically wipe a device after a password has been repeatedly entered incorrectly. Of course, you can still wipe lost or stolen devices remotely.

As mentioned earlier, Mango will let your devices simultaneously connect to multiple Exchange Server mailboxes through ActiveSync. ActiveSync policies are applied at the mailbox level, so devices connected to multiple mailboxes could be subject to multiple (and possibly contradictory) ActiveSync policies.

Windows Phone 7-based devices are often used in enterprise environments dealing with sensitive information, so Microsoft has taken steps to prevent data leakage.

Microsoft gets around this by using a formula in which the most restrictive policy setting takes precedence. For example, if one ActiveSync policy sets the inactivity period to five minutes and another policy sets the inactivity period to 10 minutes, then the five-minute inactivity timeout would be the effective policy setting because it's the more restrictive of the two. Keep in mind that Mango looks at each policy setting individually and applies the most restrictive policy settings even if that means mixing and matching settings from multiple policies.

Another feature that can help improve security is that Mango is Information Rights Management (IRM) aware. As such, Mango will let you open IRM-protected e-mail messages and Microsoft Office documents directly on your mobile device.

Windows Phone 7-based devices are often used in enterprise environments dealing with sensitive information, so Microsoft has taken steps to prevent data leakage. For example, you can only synchronize e-mail messages over a cellular connection or Wi-Fi connection. Users can't synchronize their mail simply by plugging their phone into a desktop PC as they could in the past.

Similarly, you can only transmit data files over a cellular or Wi-Fi connection. In some previous versions of Windows Mobile, you could transmit data using Bluetooth or Infrared Data Association. That's not allowed under Mango.

Windows Phone 7-based devices don't let you store data on removable storage cards. You can only store data within the device. It's great that Microsoft has taken so many steps to prevent data from being extracted from a Windows Phone 7-based device. It would have been even better if Microsoft had taken it one step further and enabled device-level encryption.

Networking Connected

Mango offers several badly needed networking-related improvements. Most importantly, it will support connecting to wireless access points that use hidden Service Set Identifiers. Besides a free Lync Mobile app, the devices will also support availability information. The Microsoft Lync Mobile app will let you search the corporate contact list and chat with multiple people simultaneously. Mango also adds broad support for SharePoint and Microsoft Office 365.

As you can see, Windows Phone 7 was initially a bit lacking, but Microsoft seems to be making a serious effort to make Windows Phone 7.1 the enterprise mobile platform of choice. R

RELATED CONTENT

- "Enterprises Go with Windows Phone 7" bit.ly/hzC61U
- "Managing Mobile Devices" bit.ly/nnDNsG
- "Manage Mobile Devices with System Center" bit.ly/gHViSb

Brien Posey, MVP, is a freelance technical writer with thousands of articles and dozens of books to his credit. You can visit Posey's Web site at brienposey.com.

BY DEREK SCHAULAND

Avoiding the Potential Pitfalls of Archiving in Microsoft Outlook

E-mail archiving has changed in a few important ways since the release of Office 2007. Here's how you can stave off some confusion and avoid help-desk calls.

Sometime this year, my organization is planning to make the jump from Office 2003 to Office 2010. I'm not really sure what to expect. As a longtime Office 2003 user, Office 2010 seems a bit foreign to me in a lot of ways—but mainly in Outlook. Some of the functions in Outlook 2010 behave differently than they did in previous versions of office.

Sure there's the Ribbon and the mouse-friendly interface, but those aren't the changes I'm talking about here. The thing I've come across as behaving very differently is message archiving in Outlook 2010. My goal here is to point out the differences—sometimes minor, but still important—between the way archiving worked in Office 2003 and the way it works in Office 2007 and 2010.

The Normal Process of Archiving E-Mails

When mail comes into Outlook, it has a date stamp on it that allows you to see when you received a message. So, when your boss comes running down the hall asking you about the message he sent to you on whichever date, you can peek into your inbox and see, "Hey, he did send me an e-mail on Friday at 2 p.m." It also provides a really good way for you to organize, sort and find messages. However, for consolidation reasons, many organizations—my own included—impose limits on mailbox size. In this scenario, the user is allowed to keep, say, 500GB of storage in his mailbox for what I would consider "live" e-mail. Once this limit is reached, incoming messages will start bouncing and being returned to sender, which in many cases is not a good thing.

To combat this limit, Microsoft has provided a way for users (or the system administrators who support them) to archive messages and other items in a mailbox so that they're moved to a PST file or something like it stored on the network. From there, the messages can be backed up to tape or disk or whatever the medium du jour happens to be. Because I'm trying to do right by my users, I like to put the ownership of the mailbox on them and make them partially responsible for archiving their messages and keeping their mailboxes clean. In most cases, this works pretty well.

Users can create archive scenarios that allow them to have messages older than so many days or weeks or months

SHUTTERSTOCK.COM

moved to their offline archives or out of their mailbox archives. When this happens, these messages no longer take up space in their actual mailboxes.

It's All About the Date

In Outlook 2003, archiving happens by the received-on date or sent date. This makes a lot of sense, because if I receive an e-mail on June 1, 2011, and my archive settings are configured to archive mail older than a month and run every three days, when July 2 hits this e-mail should get archived. It seems that this is the logical way of thinking about how messages should be moved, and in Outlook 2003 (and older versions) this is how it works. But in Outlook 2007 and later, messages get archived by their modified date. Just like other files, messages have a modified date, which is not exposed in the interface in Outlook. Users won't be aware of it right away, but they can expose the date by selecting a column header called "modified date."

Field Chooser

In most cases, the modified date and received date or the sent date will be the same if you process your mail out of your inbox and don't move it around after the fact. Archiving will function just as you would logically expect it to. However, some people like to use folders to manage their mail and will drag messages into folders for safekeeping as they've read or processed them. They'll do this without using rules or anything else of the sort. Once a message is dragged to a new folder, the modified date changes, just as it would with any other file. If you move it or touch it or open it or do anything to it, it changes the modified date. When this happens, that message may not land on the same archiving schedule as you think it will.

I'm not trying to point out a way to use Outlook differently just yet; I simply want to make you aware that archiving may behave in a new manner than what you'd expect.

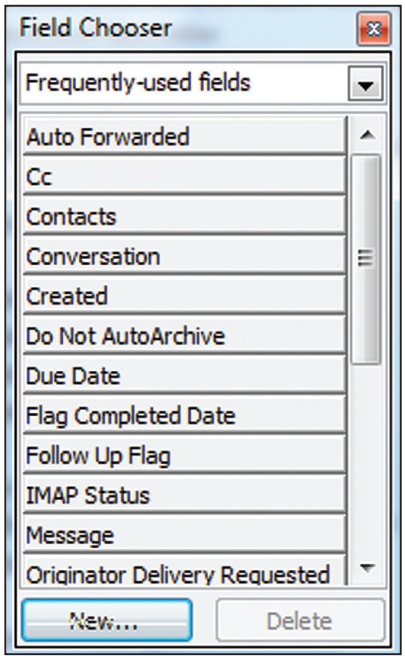


Figure 1. *The field chooser dialog box.*

When working with messages in Outlook 2007 and 2010, take the time to consider the modified date and how people might manage their mail if you (or those you work with) are folder users. This might be something that could cause some help-desk calls in your organization.

Why Does Outlook Behave This Way?

Any mail in your Outlook mailbox is just like any file on your hard disk or a network share that you access regularly. The files that are hot or accessed frequently may be handled differently than files that are cold or accessed once a year. Newer methods of backup can take this into account, keeping the newer files more readily available than the older ones. Outlook is doing the same thing by basing archiving on messages that are touched very often rather than on just the date on which they were received. Mail that you might need to read or reference because you use it a lot is less likely to be archived and moved out of your way.

It seems rather odd that this was not mentioned in any documentation I've seen. I'm OK with not being able to change the archive within Outlook to use different dates from a user perspective, but I wish this information would be presented when Outlook is configured. I didn't discover it until I was having trouble with a mail archive and couldn't figure out why no messages were being moved. Perhaps including an option to opt in or opt out of this new method of archiving would've been a good start, just to make everyone aware of it.

However, once people get used to this new feature—as is the case with the Ribbon in Office—I think they'll like it a little better. At first it was sort of confusing and took considerable time to troubleshoot, but now that I'm aware of this change to archiving I don't think it's quite so bad. It could be useful in allowing users to determine which messages are frequently used and which messages are ready for archiving.

One thing I got out of finding all of this out the hard way is that it's always good to reevaluate your archiving strategy, even if only for your own mailbox. This method of archiving will make me think hard about how often I want my mail archived, and it will likely be a much more frequent occurrence for me even though I don't tend to move mail between folders very often. In the event that I do move any messages around, keeping the archiving frequent should produce similar results to those of older versions of Outlook.

Changing User Behavior (or Administrator Thinking)

Because the method of archiving can't be changed, one thing that can be done is to train users and administrators on

In Outlook 2003, archiving happened by the received-on date or sent date. But in Outlook 2007 and later, messages get archived by their modified date.

how Outlook works. This should include more frequent archiving discussion and possibly the use of an archive folder set as a working folder rather than simply archiving directly out of the inbox (or mailbox).

When I began considering using an archive as working storage for my e-mail, I found the idea completely foreign. I think of archives as stored material, much of which may never be heard from again, so using this as primary mail was quite strange. Because the e-mail needs to be moved to the archive almost as soon as it arrives in order for the working-storage archival method to work, rules, third party add-ons or manual drag-and-drop organization might work best in this scenario.

As mail arrives, it gets shuffled off to the archive as soon as possible, perhaps by a rule that has been created to move mail to an archive folder. This way, there's rarely mail left in the inbox, and all replying, forwarding and general use happens from the archive folders. Using a rule or manual method circumvents the need to run auto-archiving. Because of this, there's no worry about issues surrounding the modified date.

Another possibility would be completely manual. As messages are processed, they could be moved to an open archive folder with a simple drag and drop. This method works similarly to Outlook's archive method but would put control completely in the hands of the user. For those who keep folders inside their inboxes or mailboxes, this might be a decent alternative. This way, the messages still get moved and the user still gets folder organization. In this case, auto archiving is likely to be disabled for all folders. I might leave sent items alone and let auto archive crank those to PST files every week or so, as those don't change much once sent.

Modified	Received
Fri 4/22/2011 11:01 AM	Fri 4/22/2011 ...
Fri 4/22/2011 10:01 AM	Fri 4/22/2011 ...
Fri 4/22/2011 10:01 AM	Fri 4/22/2011 ...
Fri 4/22/2011 8:54 AM	Fri 4/22/2011 ...
Fri 4/22/2011 8:47 AM	Fri 4/22/2011 ...
Fri 4/22/2011 8:45 AM	Fri 4/22/2011 ...
Fri 4/22/2011 8:44 AM	Fri 4/22/2011 ...
Fri 4/22/2011 8:43 AM	Fri 4/22/2011 ...
Fri 4/22/2011 8:09 AM	Fri 4/22/2011 ...
Fri 4/22/2011 7:56 AM	Fri 4/22/2011 ...

Figure 2. *The modified date in Outlook 2007 and later.*

The third alternative way to handle archiving is mostly beyond the scope of this article, as third-party add-ons do exist to help manage mail. The one I've seen, ClearContext from ClearContext Corp. (clearcontext.com), makes use of its own rules engine and management tasks to help manage mail into and out of an inbox. In looking at the rules method and the previously described manual method, this is much like a combination of the two. Looking into the particulars of an application like ClearContext is another article altogether, but for now I'll say it's certainly worth checking out.

Remember: Move Mail Carefully

Just remember to be mindful when moving e-mail around in your mailbox in Office 2007 or 2010. Mail that doesn't move should be minimally affected, but messages dragged into or out of folders for organizational purposes may not behave as you need them to when the prompt for auto archiving next appears.

The use of the modified date in archiving does make message handling a bit more consistent with how other files are handled—but making this known to the user, possibly in the "What's New" documentation, might have been a good idea. I suspect there are many users who don't move their mail around before archiving, but there are likely to be just as many who do. Those users might start calling the help desk if e-mail is not getting moved as expected, if at all. The stated size limit on PST and OST files in Outlook 2010 is 20GB. It's likely that this will be large enough for all the mail your users store; however, creating the habit of cleaning out archive folders on a yearly basis couldn't hurt, just to ensure that the 20GB ceiling stays high enough to prevent failing and corrupt archive files. The 2GB limit in Outlook 2003 has caused quite a few issues with corrupt files (certainly enough for me), and the increased size allowed for archives—with some better management habits—might be just the thing to get archiving exactly where it needs to be.

I'm sure that the issues with archiving are many, and as data increases in size and attachments grow, the limits and methods used to safely back up and store e-mail will change drastically. But for now, I think Microsoft is doing a pretty good job, even if Outlook users need to be aware of how archiving works and even if it takes some effort on their part to help things along. R

Derek Schauland has worked in technology for 15 years, doing everything from a help-desk role to being a Windows systems administrator. He has also worked as a freelance writer for the past 10 years.