



Exchange to MDaemon Migration

By Richard Harber
Decision: DIGITAL
Atlanta, Georgia, USA

Dystrybucja w Polsce



www.econs.pl

Alt-N Technologies, Ltd.
2550 SW Grapevine Parkway
Grapevine, TX 76051
Tel: (817) 601-3222
<http://www.altn.com>

Table of Contents

Can MDAemon Deliver?	3
Exchange Increases MDAemon's Appeal.....	3
Users Behavior Drives the Need for Better Performance.....	3
64-Bit Characteristics.....	4
The Cost of Moving to Exchange.....	4
How to Migrate from Exchange to MDAemon.....	5
Migration Considerations	5
Two-Server Method	6
Server Recommendations	7
Two Email Servers on One Machine	8
Why All This Work on the Old Email Server?	9
Defragmentation	10
Handheld Devices on MDAemon.....	11
Calendar / Contacts / Tasks	12
Email	12
What doesn't MDAemon do that BES does?	13

Can MDAemon Deliver?

Exchange Increases MDAemon's Appeal

Can MDAemon deliver messaging services on par with Exchange? Not too long ago, the answer was 'no'. However, as Groupware was succeeded by Outlook Connector, both the possibility and opportunity seemed to appear. And, Microsoft has recently made it a bit easier with Exchange 2007.

Exchange 07 is an exclusive card-carrying member of the 64-bit club—A bold move from our friends in Redmond. Moving to 64-bit wasn't so much of a 'want-to' as it was a 'have-to'.

Users Behavior Drives the Need for Better Performance

Email continues to exude its voracious appetite on all system resources in its path. As a general rule, people don't practice good email maintenance. For example, they want to add 'walk the dog' and 'take out the trash' as daily recurring appointments within their calendar from now until 2097. They want to have 6,000 contacts, not because they need all of them, but because there is some bragging rite associated with how big your address book is. Ironically, many of them are old and out-of-date. And finally they want instant access to all of it, every single bit, all the time either by way of the Web or shoehorned into a device which weighs less than a box of paperclips. To user, it is just suppose to work no matter how much information is there. For email server developers, it is a delicate symphony comprised of many different instruments that, when played properly, produce beautiful results. We all know that a culture change must take place with email. That is, there must be a conversation all to itself which will need to be reserved for another utopian day.

For now, like everyone else in the messaging business, Microsoft has to deal with the way things are. According to the Redmond roadmap, we were all suppose to be nicely settled into 64-bit processing by now. We're not and won't be for another year or so. This assumption likely led to some pretty interesting conversations around the water cooler some three years ago. So, the decision was made to push forward with the other big Exchange agenda—bolt a new Exchange server front end to a SQL database backend. This makes sense when you think about it. Exchange's issue would have disappeared and storage/indexing would have been handled by a true database engine. This was the plan for Yukon, the code name for Exchange 2007. However, it looks like Yukon was put on the same development train as Vista...and we all know how that has ended up. So, the train was turned around and headed back to 64-bit land. At the end of the day, this was good move for the real Exchange environments; those enterprise customers with thousands of mailboxes. It needed to happen for them. And, Exchange 2007 is really a solid product. For once, Microsoft has decided to change the world a few bits at a time instead of attempting to retrain the world.

64-Bit Characteristics

Basically, to handle more quantity and size, you need more bandwidth. Not just on the Internet side, but within a server as well. 64-bit architecture centers around memory and disk I/O. It pushes beyond the 4GB memory limit many people encounter with Exchange. To get a better idea of how much room Exchange requires, walk up to any Exchange server and open the task manager. There you will see it—the big, bloated Exchange store. The math says it all. To properly size an Exchange server, Microsoft recommends allocating about 5MB of physical memory per mailbox. Then there is the in and out part—the ability for information to get in and out of the CPU. This is where disk I/O enters the picture. In 64-bit architecture, disk I/O is likened to adding extra lanes to a busy highway. Terrific! The problem is most of the business clients we work with are mere mortals in a 32-bit world. Yet, everything is heading towards 64-bit, and the bell you hear tolling is the conductor on the 32-bit train saying, “It’s pretty much the end of the line.” However, as we all know, customers don’t wish to be forced into change unless absolutely necessary, especially when it involves the premature attrition of hardware. Unfortunately, there is no glory for the IT guy who tells a company, “We have reached the limits in Exchange 2003 and...uh...we can’t upgrade to the new version with our current hardware.” As the company decision makers start to put two and two together, they realize that two-year-old server—the one they just made the last lease payment on—won’t run Exchange 2007. What should have been an inline upgrade opportunity is now an equipment-replacement necessity.

Ok, so for most, the four horsemen of the apocalypse are not going to ride down the street due to this bad news.

The Cost of Moving to Exchange

Let’s consider the cost of moving to the new version of Exchange. We are not speaking of the baseline CALs or server software; the ‘real’ cost. For fun, let’s figure on 100 mailboxes and do a little napkin math. For Exchange to truly function properly, you need six items:

1. Dedicated, 64-bit non-domain controlling server; about \$5,000 in the white box world and \$7,000 in the brand world
2. Exchange server software; comes in at about \$700.00
3. Exchange CALs; one for you and one for you and one for you @ \$70.00/each; that’s \$7,000.00
4. Third party, two-level malware protection (store-level and brick level) Ahh, double your pleasure. Symantec is a popular one; \$40.00/user for the server portion; \$40.00/mailbox for brick level protection; call it another \$80.00/each so \$8,000.00
5. Special Exchange agent for data backup and protection; \$1,100.00 now that Symantec is still trying to digest Veritas
6. Postini / Barracuda / Ironport anyone? I am not going to add this option on top and send the whole project into orbit. However, the sheer existence of these products suggests that the Exchange level antispyam programs don’t perform well enough.

Total is \$24,000 once you add in media, shipping & handling, aggravation fees etc. What about Blackberry support? Ok, Blackberry Enterprise Server, about \$4,000 for the server + 20 users for those of you keeping track @ home. One other thing; according to RIM, BES cannot run on the same server as Exchange 2007? http://www.blackberry.com/knowledgecenterpublic/livelink.exe/fetch/2000/8067/645045/7963/7965/1180408/Installation_Guide.pdf?nodeid=1363702&vernum=0RIM

Another server? Fantastic. Let's add another \$10,000.00 for BES and you are knocking on the \$35,000.00 door. \$35K...without labor, without ongoing support and maintenance which as we all know is required by Exchange. I am going to assume that the great minds in this room do not work for free.

There are still a few poor and unfortunate souls running Exchange 2000. And there are many companies who are close to or have already maxed out the 4GB limit on Exchange 2003. When you consider the Microsoft solution, MDAemon can be a very appealing one. This is also true if the company is using handheld devices such as Blackberries or Windows Smart phones. Yes, you can use a Blackberry with MDAemon and enjoy 100% syncing. It is not exactly the same and there are certain things you should consider before suggesting the solution. Blackberry and Windows Smart Phone devices are SyncML and IMAP compliant. So is MDAemon. In fact, MDAemon was one of the pioneers in implementing the SyncML protocol. IMAP is used for mail; SyncML for everything else. The SyncML products of choice are SyncJE for Blackberries and Synthesis for Smartphones. Both work well, yet the Smartphone works a bit better. It runs Windows, remember? Let me emphasize these products are in their infancy and have issues to work thru. Blackberry Enterprise server is a mature product, built from the ground up to integrate into Exchange. Our experience has shown far fewer compatibility issues with Smart Phones than with Blackberries. And although there is a willingness between Alt-N and Nexthaus to work thru what are a scant few compatibility challenges between Outlook Connector and SyncJE, time and additional work is required to achieve the overall integration and reliability levels enjoyed by heavy Blackberry users. I'll be glad to comment more about this piece during Q & A.

How to Migrate from Exchange to MDAemon

Migration Considerations

The biggest hurdle to overcome is the migration. It is not as difficult as it is time consuming. And most of the time is really consumed before the actual migration begins. The process I am about to describe was developed by our team is the safest, and appears to produce the best results and performance with the current iteration of products involved. It begins with the adoption of a prime directive: migrations present an organization with a golden opportunity to perform data cleanups and implement mail server limits. Some important items to consider as you embark on this process. Note, I will come back around to specifics on each item in a few minutes:

1. Exchange compresses and stores all mail objects in a JET engine database. The same engine which powers Microsoft Access. As you migrate to MDAemon, this data will uncompress and explode in size. It is not uncommon to see mailboxes grow by a factor of 1.5 - 2.5 times their original Exchange size.

2. As a general rule, a mailbox should remain below 2GBs for optimal performance. This is true of both Exchange and MDAemon. Fragmentation and the indexing of content are pivotal to a mail system's performance.

Make sure your Exchange data is as clean as possible prior to migration. This is very important. If the integrity of the Exchange data is in question, be prepared to use the EXMERGE utility on the damaged mailboxes.

3. Make sure the Exchange server is properly tuned using Microsoft's Best Practices. Otherwise, the process of moving data will take longer. Microsoft has an excellent KB article and tool for this task.
4. Make sure an Exchange offline defragmentation has been performed prior to the migration

Let me stop here for a moment. If you are not familiar with the last two items, let me know during Q&A. They are must-know items when working with Exchange. If you are not using them, Exchange is NOT running at its best.

5. Archive and prune mailbox data. Additionally, empty deleted items and junk mail folders
6. Make sure there is ample drive space on the Exchange server to accommodate a base MDAemon installation. I think I just got everyone's attention: two mail software systems on one server? Hold that thought.
7. Have additional drive space available for the exported data. Local drive is best; USB II drive will work as well. No mapped network drive unless absolutely necessary.

Assuming your data is in good shape, we now look towards the move. Two choices here: One, you can start fresh with MDAemon, exporting all the old mail into .pst files and letting users open and move data at their leisure. Or two, you can truly migrate all content into MDAemon. There are advantages and disadvantages each way. The method you choose will depend on the environment, the competency of the users, and the overall condition of the existing Exchange data. Since the first choice is likely straight forward for everyone, we're going to focus on the second, more complicated one.

Two-Server Method

The best and safest way to migrate is using two servers and the MD Migrator utility from Alt-N. The old Exchange server and a new MDAemon server. This is especially true when Exchange data stores are above 25GB in total size. I know some of you are saying 'ugh...I have to buy another server for this? Why not re-use the Exchange server?'. As we all know, users are just not good about keeping their mail in check. In our perfect little IT world, this would not be an issue. Nor would there be any problems with any type of migration. Last I checked, none of us are living in that world. Murphy is alive and well. And that little law of his seems to come around at the wrong time. Remember, this is a platform change. It is Exchange. Should something go wrong with the migration, leaving your Exchange server in stasis is the wisest move. You can always restart the migration process, even on another day if you leave the old server alone. You know the VIPs or managers that have no mailbox limits in Exchange? The ones with 6GB mail stores and every email since 1995? There is a

good chance that a few of these rogue email users will have to be handled via EXMerge . Furthermore, once Exchange has been installed on a server, it won't give up its grip. It would be easier to give a woman \$10,000, place her in front of Tiffany's and keep her from shopping. The only way to fully remove its grip from Active Directory and the server itself is a full format and OS re-install, especially if someone was foolish enough to make the Exchange server a domain controller. In most cases, the cost of moving data from the server, reloading it, and moving data back is more money than adding a second server and doing the migration. It is much easier turn up your Exchange server in five minutes vs. pulling the all-nighter rebuilding an Exchange site, eating the time.

Server Recommendations

From our testing on numerous installs, and we have done many of them, here is the base server recommendation:

1. Windows 2003 R2 with Core2Duo technology
2. The server can be a domain controller, however, it isn't necessary. Domain member is fine and preferred.
3. 2GB-4GB of memory; we'd suggest 4GB since memory is very cheap these days
4. This is important: use two RAID1 mirrored sets; one for OS/apps and one for data; You can use SCSI or SAS drives, however SATA II works just as well for less money. Unless there is no other choice, do not use RAID 5. Here is why: Mirroring maintains redundancy & fault tolerance..what we all want. If you have studied the different RAID levels, RAID 5 is slower in delivering data than RAID1, especially the traditional one large container RAID5 approach. By isolating the OS and app on one physical array and the data on another physically different one, the server can talk to each array @ the same time, dramatically increasing performance of both the OS and the app. Additionally, pagefiles can be spread across two physically different arrays, a Microsoft-preferred method. And finally, Microsoft Volume Shadow Copy can be properly utilized; a terrific restore option for MDAemon
5. Install and run the Microsoft Security Wizard. Ever wonder why some of these new, dual core servers don't set the land speed records when you crank them up? Even after turning off services and tweaking, the server still doesn't have that 'umph' you expected. Shaking your head in disappointment, you spit out a few harsh curse words followed by some variation of 'blah blah blah bloated Microsoft code.' What a let-down. Then along comes SCW. Everyone familiar with this tool? Its a little diddy that Microsoft included with Windows 2003 server, to lock down a server's security settings since it is so...well...complicated to do manually. This is especially true of IIS settings. Easily missed, it is not installed by default, although it should be. Yet the title is a bit misleading as it does much more than mere security adjustments. This wizard-driven configuration tool returns performance to a server by changing and tweaking a myriad of settings and services within the OS, some of which your experience and training said couldn't change; some you likely didn't know existed. It is an awesome tool which you should use on every server you deploy. Complete the wizard, save the generated profile, and apply. That's it. If you are unsure of a particular setting affect on the

server, create different profiles and evaluate each one. You can easily rollback, unload and re-load @ will by re-running the wizard and choosing the appropriate profile file. How much do you get back? A measurable 15-20%. Really. Not a hard choice here. Let the CPU get whacked silly by behind-the-scenes, or have the server run like a well tuned Ferrari. The most important settings to disable: digital signing of files, audit logging, pre-Windows LAN Manager authentication

6. Don't let Windows determine the best overall performance level for itself; peg the performance settings within Advance Performance properties to 'Best Performance' and not the default. No one needs all those fade effects, transitions, and fancy graphic overhead on a server. No one.
7. Perform full defragmentation of the data drive before migration is begun. Do it multiple times until it reaches a flat level of 1.0. And don't use the Microsoft defrag utility. Discover and use Diskeeper. If you don't want to buy it, download and install the 30-day trial. Trust me, this program is well worth it and needed on every server you deploy. Windows is a fragmentation beast, especially on initial OS installations. Diskeeper has a nifty set-it-and-forget-it feature which will background defrag and keep the drives in check.
8. Install MDAemon on the server and setup AD integration. Allow the creation of the mailbox accounts
9. Confirm the mailboxes have been created and shut-down the MDAemon service
10. Leave this server and return to the Exchange one.

Two Email Servers on One Machine

This brings me back around to the two-on-one scenario. Two mail server software systems on one physical server: the Exchange server. Yup, we're gonna install MDAemon on the Exchange server to facilitate the migration. Call it the mail mule.

For those of you still following, we're now on the Exchange server:

1. First, begin by downloading and install SyncBack, a free data copy utility; you'll thank me for this one. More in a minute
2. Create an AD account named mdmigrator. This account needs to be in the following groups:
 - Administrators
 - Domain Admins
 - Enterprise Admins
 - Schema AdminsBasically a g-d account
3. The mdmigrator account also needs service account access to all Exchange mailboxes:
 - <http://support.microsoft.com/kb/821897> for Exchange 2003
 - <http://support.microsoft.com/kb/262054/> for Exchange 2000

4. Install MDAemon on the same server as Exchange Management Tools. This is usually the Exchange server but can be another server if room does not permit
5. Log into the Exchange server using the 'mdmigrator' account
6. Now you can follow the steps outlined in the Alt-N MDMigrator whitepaper.

As the saying goes, a watched pot never boils. So, go and get a cup of joe...the migration has begun.

Why All This Work on the Old Email Server?

The best results are achieved when all data exporting and importing is carried out locally instead of across the network; although one extra step will be required afterwards, this is faster and more reliable. Once the data has been successfully exported and dumped into the 'mule' instance of MDAemon (all automatic I might add), you can simply copy the user folder structure directly from the mule over to the new server...if you trust the Windows copy system. A small piece of advice: Don't. You never want to be 90% thru a 50GB copy and have some misfiring process interrupt the transfer. You all know about it. That dreaded 'Cannot copy blah blah blah ' dialog box that gives you one choice...'ok.' Yea, it's not ok. It may as well say, 'sorry pal, you have to start all over.' A loud scream is heard from the server room as you restart the multi-hour copy process. This is where SyncBack becomes your best friend. This may be the best piece of free software on the planet. Why copy, when you can sync. Better yet, why be unsure when you can be sure. As the name implies, SyncBack will perform a scheduled copy or sync of data from one server to another. It is easy to use and has some very crafty rules so that you don't destroy or overwrite the original. And, it can run even while content is changing. Ahhh, I know what you are thinking. The answer is 'yes;' you can start the sync process while the export from Exchange is taking place. Best part, it will alert you when it is done and run a verify to make sure everything made it to the new location. You can't do THAT with Windows copying. You will find a hundred uses for this little jewel. Did I mention it is free?

After the data has been successfully copied to the new server, MDAemon can be started up. As it starts the first time, indexes need to be built for all mailbox items (emails, contacts, tasks etc). The server is going to kick into high gear. This is where the separate RAID1 arrays will really demonstrate their power. In addition, MDAemon will begin to generate log files as usual. Lots of them. Congratulations, your server is now live, albeit heavily fragmented. Don't do any speed or performance tests quite yet or you will be disappointed. Remember all those mailboxes you moved over? Loaded with fragmentation. Loaded. Same is true of the logs.

Unlike Exchange, MDAemon stores information in a flat-file/folder format. This is a distinct advantage for moving data, quickly locating messages, and data protection. For example, you can implement Microsoft Volume Shadow Copy to take snapshots of your MDAemon folders every few hours, restoring a single message or accidentally removed folder, at any time. And, if you have enough storage, you can have weeks or months of snapshots to choose from. This is HUGE. Not so with Exchange. Funny, a Microsoft product that doesn't work with another Microsoft product. Hmmm. Yet, like everything else in life, sometimes an advantage also presents a disadvantage. The byproducts of this design is more indexing and fragmentation.

Defragmentation

Everyone is familiar with fragmentation so there isn't really a need to define it. In the MDAemon universe, with so many messages in so many different files and folders, fragmentation is much more pronounced, and occurs at a quicker pace than Exchange. Unchecked log files are the biggest culprit. As fragmentation levels increase, indexing time increases. Keep in mind that MDAemon creates an individual index file, of all objects, for every user folder called HIWATER.MRK. With high fragmentation levels, the server and hard drives must work harder to 'find' the location of content. This delay manifests itself on both the Outlook Connector and the WorldClient sides in similar ways. You will notice a bit of lethargic behavior, especially when drilling down into customarily large folders such as 'Sent Items.' There are a few checkboxes within MDAemon to tweak these settings, however, the root of the problem is fragmentation and it must be addressed and controlled from the very beginning. So, there is a direct correlation between fragmentation levels, index building speed, and overall performance. How do you overcome it. Four ways:

1. Use Diskkeeper 11 to automatically keep the drives defragmented
2. Turn down the MDAemon text logging unless you need to do diagnostics.
3. Turn off a few of the features within WorldClient

Skip initial mail check

Collapse my nested folders

Empty trash at signout

Hide deleted messages

Limit messages per page to between 10 and 25

4. Implement domain-wide mailbox limits, especially auto-deletion of deleted items

You guessed it, time to kick off a manual defragmentation. It will likely take a few passes before you get to the 1.0 level. After this process completed, schedule a checkdisk with the 'fix' option and give the server a good reboot. Now you can sit back and briefly gloat over your fine work as the server-portion should be stable and running well. Only thing left to do is get the Outlook Connector installed on the PCs. This is very straight-forward so I won't spend a whole lot of time discussing it. A few tips though.

When configuring the Outlook, don't be overzealous with checking mail. That is, set the 'check mail' timer to 2 minutes or more. No one, and I mean no one, needs to check for mail every 30 seconds, or one minute. There is no need to flood your network with all that BS mail checking traffic. For that one nail-biting message that will arrive any second, users can manually click send/receive. If your life hinges on checking mail every 30 seconds, you shouldn't use email. Use another technology, called a telephone, to get what they need. Or, do what we do: deploy ComAgent as a safe, internal IM solution. Lunch plans can be made much quicker this way rather than email.

5. Set the folder checking to 'Show Only Subscribed Folders.' No need to constantly check folders you don't have access to or use, right?
6. Set the Send/Receive to check mail in specific folders, like the Inbox, instead of all folders. Unless you are using crazy rules or have multiple inboxes, this is all you really need.
7. Refrain from changing the location of the local cache file. This was a real issue in previous versions of OC. However, as a general rule, it is best to leave it in the default, protected location.

The first time you launch Outlook, it will build the local cache file. This is a local copy of your server-based mailbox. This may take a good few minutes depending on the size of the mailbox. Don't fret, this is normal. Naturally, the server will 'feel' it if a gang of users with larger mail stores all open Outlook for the first time, at or around the same time. Now, keep in mind that fragmentation also plays a key role in the performance of Outlook. Yea, that local cache file gets fragmented to hell and back since it grows and shrinks all the time. Should you run into someone who appears to have slow Outlook performance, and you eliminate all the obvious stuff, a good defrag will have a profound affect on performance. So, will purging their old email. Oh, one side note: Everyone knows that the archiving system in Outlook is not very good. The great minds in Redmond have based the rules engine on the modified date instead of the received date. Well, the modified date is going to change to the migration date once the migration has completed. Hence the reason you archive BEFORE the migration process.

Handheld Devices on MDAemon

I save the best for last. How to get a handheld device to talk to MDAemon. There are all sorts of variables which affect how well it will work. Variables outside the mail server. Unlike using Outlook on a LAN, or WorldClient over the Internet, handheld devices are dependent on a varying, wireless network. We have seen the best results on the Verizon network. The worst on T-Mobile. It's all about the speed of EVDO and a device that works with it. Most newer Blackberries and Windows Mobile devices leverage the broadband speeds of these data networks. Older devices, especially Blackberries, communicate at a palsy 12k. Anyone know the piece of data which will bring a handheld to its knees? It's not what you think. Email right? Nope. Recurring calendar events or tasks. Keep that in mind as you stare down that big mailbox with 6,000 contacts and 10,000 tasks. All those bits have to fly thru the air and land in that tiny little device. At slow speeds, multiple syncs will likely be required in order for all data to make its way to the device. In cases where the task and contact counts are very high, you may be forced to batch sync. That is, break down a larger sync job into a number of smaller ones. This is a manual, time consuming process and is not preferred. Eventually, you will suffer the affects of information overload in the handheld device. And, you can safely assume it will happen to the user at the worst possible time. Instead, create a new Outlook Connector profile and copy relevant, forward looking data into it, leaving the other profile intact for historical reference. All the more reason to purge, purge, purge before bringing a handheld into the mix.

For all phones, be sure to set the SyncML password for each user within MDAemon. Otherwise, it won't work. Second, make sure that there is an IMAP and WorldClient rule in your firewall, pointing at the mail server. SyncML uses the WorldClient port. IMAP uses port 143. Third, make sure whichever phone you choose to use is up-to-date on firmware and patches. Very important.

Calendar / Contacts / Tasks

1. Download and install the Synthesis SyncML client for Smart Phones; SyncJE for Blackberries
2. Install and configure the respective SyncML client it on the device
3. Use the Alt-N instruction guides to make sure you enter the web address exactly how it is documented.

For calendar and tasks, set the phone to sync forward, not backwards and forwards. That is, tell the phone to grab events going forward for the next 60-90 days. Don't load it down with all of your calendar events and tasks for all of 2007 and 2008. If you need some history, don't go back more than 30 days. This is a common mistake and the biggest reason why the sync experience is a miserable, time consuming one for the user. Remember, a handheld device is supposed to show you what is going on now, and a short distance into the future. It is not suppose to contain your life for the next five years.

4. Check signal levels and kick off the first sync.

Email

1. Configure IMAP account settings
2. Program for 3 days' worth of email
3. Set the interval for checking mail; minimum is 15 minutes; don't worry, this is often enough. Remember, if you need it quicker, use a phone.
4. Check signal level and kick off the first sync

MDaemon has some cool tools to help you monitor and troubleshoot handheld issues. First, there is a SyncML tab in the GUI which allows you to watch the process in real time. Second, there is a change log file which can be viewed at any time. Once MDAemon is aware of a handheld device, it knows to batch all of the information which is supposed to be transmitted. The handheld does the same thing. When the sync process begins, a comparison of sorts is made and the exchange of this change information begins, in both directions. This is the 'magic' of the SyncML protocol. On the server side, these changes are stored in a log file called a change log. Make a change inside Outlook or WorldClient, and the information should be written to this file. From there, it is up to the device to check in and grab it. One cool caveat: the change log stores the date and time of the last sync from the device. You will find this information very helpful as you troubleshoot sync issues...and you will have them.

What doesn't MDAemon do that BES does?

When you talk about comparing handheld devices, the conversation seems to revolve mostly around Blackberry. That's because smart phones work about the same across platforms. Blackberries are different and have their roots in being a real corporate solution. BES is not only a conduit for passing data to Blackberries but a management tool for them as well. Think of it as Active Sync, Group Policy, WSUS and remote device management for Blackberries. Microsoft has built similar features into Exchange 2007 and Exchange 2003 with SP2. They have grown the Exchange Active Sync feature list to include many of the items which used to be exclusively available to BES users only. And, taking a cue from the SyncML world, have implemented a direct push technology to speed up syncing. Why? Because the Windows Mobile platform continues to get more intelligent and now competes with Blackberry. Why have to invest another \$10k in BES if you can get nearly all that functionality in Exchange for free. After all, Exchange is already working in conjunction with Group Policy, and WSUS right?

Currently, the MDAemon/SyncML combo provides a conduit only. The remote management and policy enforcement of devices are not part of the current iteration. However, they are defined in the device management piece of the SyncML initiative. Considering that Microsoft's Exchange Active Sync is built on HTTP and XML, and works natively with Windows Mobile/Smart phones, it is conceivable that these device management features can be implemented in the MDAemon/SyncML space. Product longevity and viability will likely drive its development. Should you consider this a weakness of the product? Not at all. If a customer is dependent on these features or desires to keep them right now, MDAemon is not the right choice. For many companies, this will not be the case. For many companies, Exchange was not the right choice, it was the only one because of BES.

Listen and explore. Listen to your client's needs and frustrations. Explore how they are really using their email system, and what features are important to them. You will be surprised by what you hear. Few companies truly leverage the robust capabilities of their respective mail systems. Nor do they understand what they are missing. What they do understand is the need to send and accumulate email, share their calendar with others, and have it all on their hip. Positioning MDAemon is not about selling against Exchange, it is about understanding it. It is about where Exchange really belongs. It is about choice. MDAemon is a formidable and well-rounded solution with its robust feature set, tight integration, and absence of extensive maintenance. Mix in ongoing innovation and a 66% savings over an equivalent Exchange setup and you have quite a compelling case. No, MDAemon is not the right solution for everyone. Neither is Exchange.