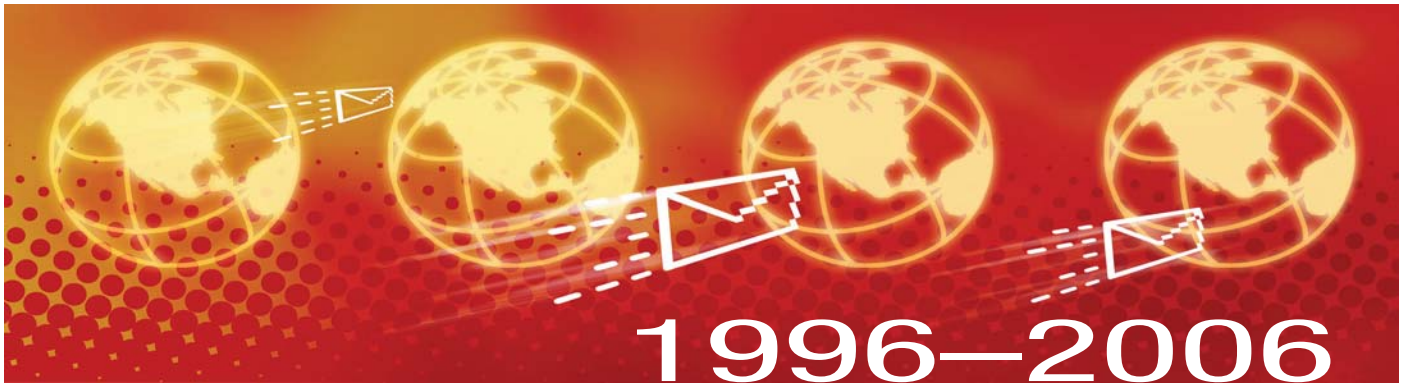
A Decade of Exchange



Windows IT Pro

CONTENTS

A DECADE OF EXCHANGE: 1996–2006

This special section was produced by *Windows IT Pro* and appears as an insert in the April 2006 issue of *Windows IT Pro* magazine and the May issue of *Exchange & Outlook Administrator*.

Working With Exchange Has Been a Blast!

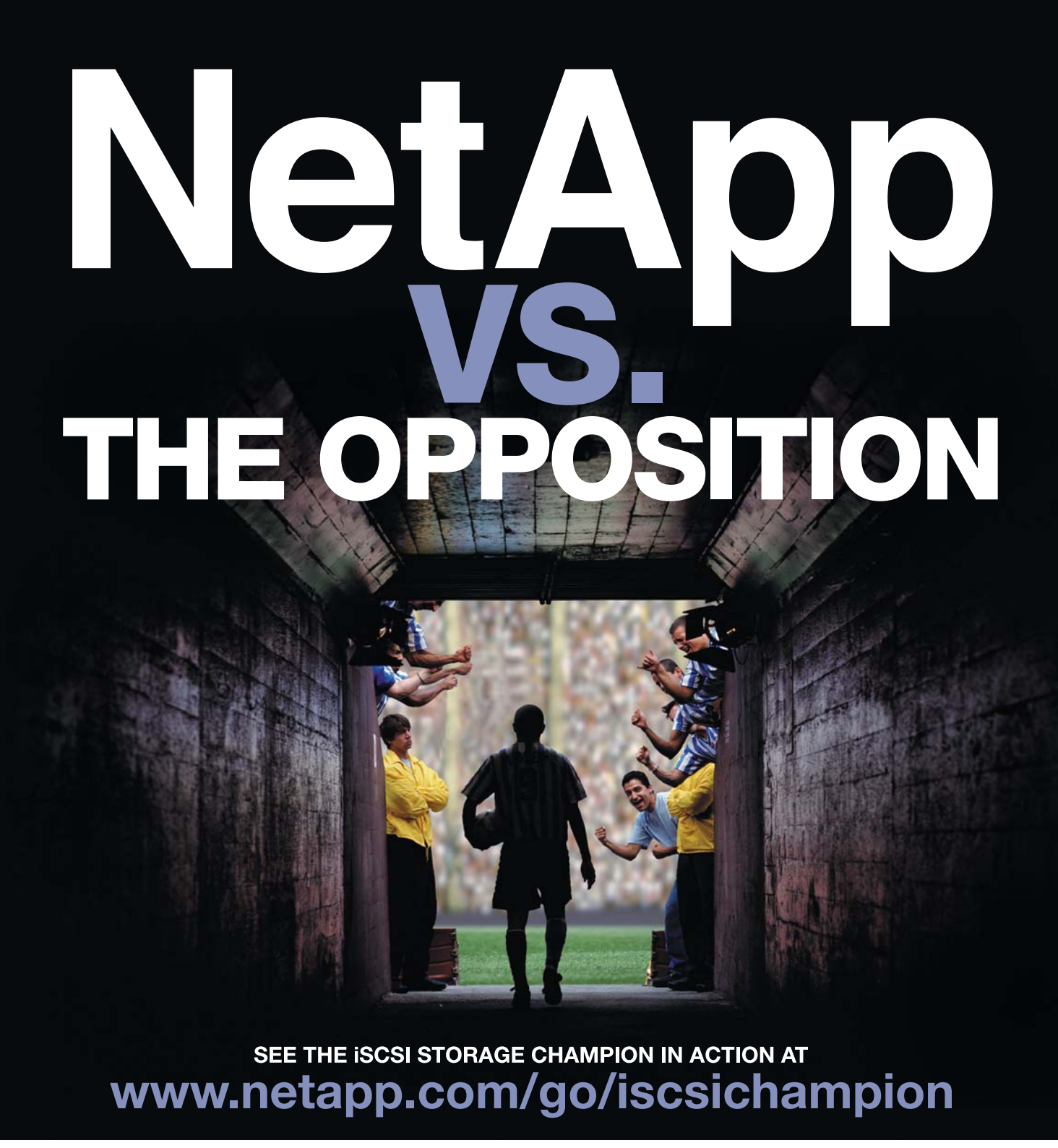
by Tony Redmond

Microsoft shipped Exchange 4.0 in March 1996 after a gestation period of some four years. The new messaging server went through many different design phases. Microsoft grappled with the challenge of enterprises and small companies, figured out what it had to do to be competitive, understood how best to migrate users from other platforms (including their own), and achieved the necessary performance and scalability levels—albeit limited by the capabilities of Windows NT 3.51 and the available hardware.

Exchange replaced Microsoft Mail and went into immediate competition with other messaging systems such as those favored by large corporations (IBM PROFS, Digital Equipment Corporation's ALL-IN-1 and MailWorks, and HP OpenMail) and the PC LAN-based systems such as Lotus cc:Mail, Banyan Vines, Novell GroupWise, and Lotus Notes. Exchange 4.0 was the first version in the initial Exchange architecture and this generation spanned Exchange 5.0 and 5.5, released in 1996 and 1997 respectively. The next generation arrived in 2000 with Exchange 2000 and Microsoft developed the architecture further with Exchange 2003. Exchange 12 advances the state of the art with the third distinct architecture for Exchange

It's hard to realize just how much progress messaging technology has made since 1996. Exchange has improved its capabilities dramatically in terms of functionality, robustness, security, and connectivity. We have also seen other important advances in the standards that dictate how systems connect together, the networks that we use, Windows and associated technology such as IIS, the devices that we connect with to our mailboxes, and other technology that has established the type of world we work in. The Web is the best and most pervasive example of such an influencing technology. The volume and depth of change over the decade has posed a challenge for administrators to keep up to date with new developments, and hopefully the articles published about Exchange and associated technologies in that time have helped to bridge the gap.

NetApp vs. THE OPPOSITION



SEE THE iSCSI STORAGE CHAMPION IN ACTION AT

www.netapp.com/go/iscsichampion

When choosing an IP SAN partner, go with the undeniable market leader. NetApp pioneered innovative iSCSI-based IP SAN solutions and have deployed more than 3,000 worldwide. With NetApp you'll leverage the cost and simplicity of IP networking. Plus, our comprehensive solution – composed of hardware, software, and services – integrates seamlessly into your Windows server and applications environment.

Check out our secret weapon in action at
www.netapp.com/go/iscsichampion

NetApp®
Simplifying Data Management

Microsoft®
GOLD CERTIFIED
Partner

THE WAY WE WERE

The messaging market was much more fragmented in 1996 than it now is. The administrator who set out to deploy Exchange 4.0 had to cope with a plethora of competing standards, connections, and clients. Companies such as SoftSwitch (later bought by Lotus), WorldTalk, and LinkAge (later bought by Microsoft as part of its push to migrate companies off Notes) built healthy businesses by producing software to connect different email systems so that companies could communicate together. The war between the proponents of the international messaging standards (X.400 and X.500) and the Internet standards hadn't reached a satisfactory conclusion in 1996, so we struggled in a world where you needed a great deal of magic incantations to send even a plain text message addressed to a single recipient to a foreign email system.

While we thought that X.500 would eventually result in a global directory standard that everyone could build to, directory synchronization was another black art and it was common to have weekly or monthly synchronization runs to merge directory data and provide a common view of users across multiple systems. Email addresses were more convoluted (mine was Tony.Redmond@dbo.mts.dec.com). Of course, X.500 has faded into the background since and LDAP has become the most widely used standard for directory access and interoperability. We can still see the influence of X.500 in some enterprise directories and in the design principles that Microsoft followed to build the original Exchange Directory Store and then the Active Directory, but few Exchange administrators bother about X.500 now.

TRIVIA QUIZ

Microsoft gives each version of Exchange a code name. What versions do these code names belong to?

1. Titanium
2. Touchdown
3. Osmium
4. Platinum
5. Merlin
6. Mercury

Answers on page 4.

The ease of connectivity established by SMTP, its extensions (ESMTP), and easy access to the Internet has revolutionized email. This is true for corporate users and personal users because it would have been difficult to predict the success and ease of access to email systems such as Hotmail, Gmail, and Yahoo mail 10 years ago.

THE PROTOCOL WARS

MAPI is the great survivor of the protocol wars (MAPI is actually an API, but many people refer to MAPI as a protocol, in the same way as they refer to IMAP4 or POP3). Microsoft used the first version of MAPI in Microsoft Mail, but this was a very simple version of the API that Outlook 2003 uses today because it supported only 12 functions. "Capone," the original Exchange client shipped with Exchange 4.0, was the first client to exploit full MAPI. Microsoft referred to Capone as a "viewer," which was an odd name to give to a client. The Capone client was elegant and simple, but the first release of Outlook in 1997 rapidly outdated the original Exchange client. Outlook still uses MAPI today and it boasts a range of features that most users find difficult to comprehend, let alone use. Despite rumblings over the years (many from within Microsoft) that Microsoft wanted to drop MAPI and use Internet protocols for its clients instead, no Internet client protocol has emerged that could deliver the same functionality as MAPI, so it has powered ahead as the engine for Outlook. MAPI remains a mystery to many, so if you're interested in finding out more, head over to http://www.insidemapi.com, the Web site dedicated to "Inside MAPI," the definitive book on the API (out of print for many years).

Of course, Outlook is not the only client that you can connect to Exchange, and ever since Microsoft realized that it had to support the Internet in mid-1996, Exchange has been able to support other client protocols. First, Exchange supported POP3, then IMAP4, and then via HTTP to the Outlook Web Access (OWA) client. OWA is a real success story for Microsoft. Like many other projects that come out of Redmond, the initial version (shipped with Exchange 5.0 and then 5.5) was slow due to some aspects of its architecture, its interaction with the Store, and implementation details, so it could not really scale. The version of OWA shipped with Exchange 2000 marked a dramatic step forward in the UI and performance and OWA became a client that you could use to do real work with. Microsoft made further improvements to OWA in Exchange 2003, not least of which was to respond to the needs of the service providers who wanted to deliver segmented functionality to their users. Further improvements are in Exchange 12. The bottom line with OWA is that many users who work in offices with reliable network connections find that they do not need to use Outlook because all the functionality that they need is in OWA.

EVER INCREASING MOBILITY

We were just getting used to having cell phones in 1996 and the pager was the most common device that people carried if they needed to keep in touch with the office. RIM (http://www.rim.net) was founded in 1984 and developed its BlackBerry device as a solution that was initially targeted at executives. Now, BlackBerry has become a term that people understand to mean constant connection to the office and many of those connections are to Exchange. Of course, BlackBerry is not the only mobile device that Exchange supports. The GoodLink server (http://www.good.com) connects BlackBerry devices to Exchange along with its own devices and those running Palm OS and Microsoft-powered PDAs. You can choose from a wide range of SmartPhones, too, and Microsoft continues to focus on mobile access as one of its key development strategies for Exchange.

Microsoft did a huge amount to improve connectivity for mobile devices in Exchange 2003 and has recently made many improvements in Exchange 2003 SP2, especially for devices that run Windows Mobile 5.0. You can expect even more functionality for mobile users in Exchange 12, including Microsoft's first venture into the uni-

TRIVIA QUIZ ANSWERS

1. Exchange 2003
2. Exchange 4.0
3. Exchange 5.5
4. Exchange 2000
5. Exchange 4.0 (trick question – due to the length of the development project, Microsoft used several code names for Exchange).
6. Another trick question. Mercury was the code name for a release that Microsoft scheduled after Exchange 2000. However, it never appeared and Microsoft proceeded to ship Titanium (Exchange 2003) instead.

fied messaging market. Microsoft's favorite demo for unified messaging is to show how you can ring Exchange while you are en route to the office and cancel all your appointments for the day, perhaps because you have another pressing engagement. Having such a wide range of connectivity options is very convenient for users, but the sheer number of connections that an Exchange server now supports has put a huge load on kernel mode resources that Windows finds hard to satisfy. Increasing the server's ability to support client connections is one of the primary reasons why Microsoft made the decision to make Exchange 12 a 64-bit release only. It's also fair to say that the increase in user connectivity options has made administration more complex because of the places where things can go wrong and disrupt the messaging flow.

INTERESTING PROJECTS

Like any technology, Exchange is useless until you deploy it in projects to solve customer business problems. The early projects were all about migration and usually contained some interesting technical problems. For instance, one European post office wanted to replace PROFS with Exchange, but they only had 2Kbps connections to each post office. The link was OK for "green screen email," but could not handle the RPCs that Exchange and MAPI clients depend on. Later on, we faced the challenge of bringing Exchange and Windows together because Exchange 2000 absolutely depended on a solid implementation of Active Directory before it could function. Active Directory posed a huge learning curve for administrators, system designers, and consultants alike, but we have past the curve now and the combination of Windows 2003 and Exchange 2003 is a much more stable platform. The new combination of 64-bit Windows and 64-bit Exchange 12 will take some time for administrators to become accustomed to, but it should be as stable as Windows 2003/Exchange 2003.

Introducing Exchange to the world of Internet service providers broke a lot of new ground around the turn of the century. Microsoft had not designed the first versions of Exchange to deal with the demands of ISPs, yet they expected Exchange to replace MCS, the company's previous email solution for ISPs. The world of ISPs is significantly different from enterprise deployments because the focus is all about short connections for huge numbers of POP3 and IMAP4 clients rather than the leisurely extended connections enjoyed by corporate users. Maybe the most interesting project was the system deployed to provide email to political parties. Even within the same party, users did not trust each other and the politicians were not happy to have their email stored on the same computer as data owned by other politicians. This was not a technology challenge—except in convincing the users that Exchange and Windows could provide the necessary security to isolate everyone's data and keep it secure—but there were many interesting debates along the way.

THE NOT SO GOOD POINTS

Not everything has gone well for Exchange since 1996. Public folders are probably the biggest piece of functionality that has underperformed and disappointed across a large number of deployments. When Microsoft was stoking the market before it shipped Exchange 4.0, the company made enormous play about the capabilities of public folders, especially when you linked them to the power of the 16-bit Visual Basic-like Electronic Forms Designer (EFD). With EFD, you could quickly put together a form such as a travel request or expense claim, link it to a public folder, and allow users to create and use the form much more easily than paper equivalents. With replication, you could move that information around your organization and collate it centrally. It all looked promising, but in practice EFD was a disaster because it generated forms that performed well with a couple of users or with a small number of items in a folder, but rapidly ran out of steam after that.

EFD sank quickly while public folders have lingered on. Microsoft has made a couple of runs at improving public folders, most notably when they introduced multiple folder hierarchies in Exchange 2000, but no one seemed to be interested because public folders are difficult to manage and maintain and it did not seem like a good idea to introduce more complexity with the extra folder hierarchies. The net result is that many companies have large numbers of public folders, but no good way to audit, clean up, report on, or effectively manage their contents. We will not mourn the passing of public folders when Microsoft eventually puts a bullet through them. Exchange 12 marks the start of the phase-out process for public folders. Other technologies, such as Share Point Portal Server, did not exist in 1996 and do a much better job of categorizing, searching, and managing data, so it will be a relief to move.

APIs are the other disaster area for Exchange. Microsoft needed Exchange to have great programming capabilities to help wean companies off Lotus Notes. Notes is not a great messaging engine, but it has extremely strong collaborative and programming capabilities that companies exploit to put together mail-enabled applications that take advantage of the Notes replication engine (also better than the replication implemented in Exchange public folders). We have endured multiple attempts by Microsoft to deliver an equivalent development platform for Exchange. To give Microsoft credit, it is persistent, and the company has been very persistent, but very awful, with the APIs that have shipped with Exchange. We have seen CDO, CDOEXM, Exchange

AN AMD PERSPECTIVE

"Microsoft's Exchange has had a huge impact on the way companies conduct business," said Joe Menard, Corporate Vice President, Software Strategy for AMD. "The evolution of Exchange, from the simple mail application based on a single database store introduced 10 years ago to the comprehensive enterprise-class email and calendaring system of today, closely follows the evolution of the Internet as a key business communication medium. Exchange 12 is expected to deliver the next level of innovation for this application—integrating email and voicemail. This advanced functionality will more fully utilize the 64-bit, multi-core and architectural innovations AMD is driving into the x86 market. AMD is teaming with Microsoft to help ensure that Exchange 12 running on AMD Opteron™ processor-based servers provides the powerful platform required for the next generation of digital messaging."

Routing Objects, the infamous EFD, WMI, client-side MAPI and server-side MAPI (yes, there are differences between the two), WebDAV, and so on.

Microsoft knows that it has a mess on its hands and we all hope that the advent of Monad support in Exchange 12 will at least mean that we have a solid and robust interface to build management tools upon. At this stage, things look good because Microsoft is building the new version of Exchange's management console on Monad and it ships a selection of cmdlets (packaged commands that you can reuse in your own code) with Exchange 12. Exchange 12 also promises great things in terms of support for standard Web services. Outside of that, there is still no good way to develop mission-critical client side applications that exploit the storage and messaging power of Exchange and we await developments in this area.

REVIEWING PREDICTIONS MADE IN 1996

Scary as it seems, I have been writing about Exchange since 1996. The vast bulk of my scribbling has appeared in *Windows IT Pro* magazine and the "Exchange and Outlook Administrator" newsletter, and I hope that the articles have helped you understand and exploit Exchange to the maximum over the years. However, my first article appeared in a publication called "Exchange Manager" that did not last very long. I wrote an article called "Scaling Exchange" where I looked at the practical issues involved in scaling Exchange 4.0 to deal with hundreds of users (my advice was not to support more than 300 users on a server). I wrote: "Lots of people get hung up about the 16GB limit for the Information Store… I don't, because it's a limit that most of us will never encounter." Gee… I was right in one respect because Microsoft has only just upped the 16GB limit for the standard edition of Exchange in Exchange 2003 SP2, but the sheer number of messages that we send and the average size of those messages has exploded since 1996. Then, most messages in corporate email systems were between 5KB and 10KB. Now, they are bloated through a mixture of user indiscipline (horrible autosignature files, too many replies to replies, and so on) and huge attachments.

I went on to ask: "When was the last time you saw a Windows NT system that had more than 100GB of disk attached? Or more than 4 CPUs? Or even more than 256MB of memory or 512MB on a RISC system?" How times have changed. In my defense, our first Exchange servers boasted 66MHz 486 CPUs, 64MB of memory, and 4GB of disk. Today, the advice is to buy 64-bit capable systems to be ready for Exchange 12, preferably with dual CPUs, gigabytes of memory, and lots of disk. In its justification (http://blogs.technet.com/exchange/archive/2005/12/29/416613.aspx) for the move to an exclusive 64-bit platform for Exchange 12, Microsoft cites the fact that it believes that 500GB disks will be standard when Exchange 12 ships and that 1TB disks will be available.

I looked into the future by predicting that "in the long term, the evolution of Windows NT to support 64-bit computing will raise the performance bar even further and allow people to consider even larger systems … systems that can support thousands of users on a daily basis." I went on to ask Microsoft to consider raising the limit on the Information Store from 16GB to 16TB (Microsoft did this for the Enterprise version in Exchange 2000); support clustering (Microsoft shipped Wolfpack clustering in 1997 with Exchange 5.5, but the Exchange clustering story has been an uneven success since); support a single mailbox restore (possible with third-party products since 1997, PSS generated the ExMerge utility in 1998, the Mailbox Recovery Center arrived in Exchange 2003); provide better support for multiple processors (done in Exchange 5.5 and much improved since); and optimize for non-Intel processors (alas, the multi-platform play for NT and Exchange terminated after Windows NT 4.0/Exchange 5.5, but the AMD x64 platform is now a great success). Looking back, it was not a bad list to ask for.

WORKING WITH MICROSOFT

The past 10 years has marked enormous growth for Exchange (to some 140 million seats) and it is now the world's most commercially successful email system. As is the norm in software development, the Exchange developers have changed a lot as versions came, but they have always been a good team to work with. I disliked Microsoft's decision to halt the annual Microsoft Exchange Conference (MEC) because I think that it was an event that the Exchange community valued, even if Microsoft marketing did not. However, we have the Exchange Connections event now, and the event is gathering momentum in attendance, content, and all-round support (the first European Exchange Connections event takes place in Nice on April 25-27).

INTO THE FUTURE

Overall, working with Exchange has been a blast. I've met great people across the Exchange community and enjoyed the projects that I have worked on. The only question is how the next 10 years can be any better. ■

Tony Redmond is a contributing editor for *Windows IT Pro*, a senior technical editor for *Exchange & Outlook Administrator*, vice president and chief technology officer for HP Services, and author of *Microsoft Exchange Server 2003 with SP1* (Digital Press).

The Evolution of Exchange Hardware

by Pierre Bijaoui

The way Exchange 4.0 was architected and designed is quite different from what is done today. There are several reasons for this: the hardware has changed, the practices have changed, and the software components that make up an Exchange network have changed. With this in mind, let's study what we had to deal with 10 years ago, and how things have changed.

Exchange 4.0 servers didn't need to host many users because users tended to be distributed, and the network communications were such that you had to deploy servers near users. This resulted in servers with 400 to 500 mailboxes, with relatively small disk capacity. At the time a 20MB mailbox quota was considered quite reasonable for most corporate users.

When Exchange 5.0 appeared with the unlimited store, we saw consolidation: reduction in server count in the infrastructure to make it simpler and cheaper to operate. At the same time came the increased need for uptime. Email was not mission-critical for most companies in the late 1990s, but as businesses gradually integrated email in their process, mail could not be unavailable. This was a challenge for IT administrators: The need to achieve consolidation, shorten problem resolution times, and implement a simpler infrastructure that can scale and is easier and cheaper to maintain.

Servers and storage are the key hardware components that Exchange requires: servers can have many roles, such as mailbox, public folder, message transfer, and front-end. Storage is required for two areas: mailbox and shared data (public folders). For mailbox data, it's a never-ending battle of size versus recoverability: how can I have large mailboxes, many of them, and still be able to back them up and recover them within acceptable time limits.

SERVERS: THE NEED FOR SPEED

To increase a server's speed, give it fast processors and RAM with data to process: the less RAM you have, the more paging and network activity, which will keep your processors waiting.

`Symmetric Multi Processing and Processors`—Many Exchange servers today run in dual or quad processor modes. One major architectural advance was AMD's alternative to Intel's central chipset implementation: by attaching a memory bank directly on processors, you avoid having a central component becoming a possible point of contention.

Processors communicate by means of a bus with other components of the server, such as memory and external devices (network, disks, video boards). While you can have fast processors, speed serves little purpose if you can't feed the processor fast enough. Also, you should not judge the power of a processor by the speed it operates at. In fact, modern processors tend to run slower to reduce heat, which in turn requires less air cooling and reduces power consumption. In other words, fast processors will consume more; slower processors will save on electricity and data center bills.

For a long time, the sweet spot for Exchange was on four-way servers, because more than four processors increases the inter-process bus communication overhead and results in reduced performance gain. Exchange is a multi-threaded application but it has only one core process, the Information Store (STORE.EXE), which doesn't scale well across processors. In light of this, the industry improved the execution capabilities of processors. Intel introduced Hyper-Threading technology, which enables a single physical processor to present two machine states back to the operating system. Next, AMD offered the dual-core, followed by Intel, which implemented two physical processors on the same socket.

Exchange benefits from this additional processing power without major architectural change. In the future, quad-core processors may have an even greater impact on Exchange application performance and be delivered at an affordable cost.

`Memory`—In 1966, it was common to ship an Intel server with 32MB of RAM and scale it up to 512MB (and sometimes 1GB) for most models. Today, most Microsoft Exchange servers have 2GB of RAM for small models and up to 4GB at the high end. If more memory

improved Exchange performance, it would be used. However, Exchange does not benefit from more than 4GB of RAM. Exchange's 32-bit architecture does not permit this, even though the NT kernel allows addressing more than 4GB through the Physical Address Extensions (PAE) and Addressing Windowing Extensions (AWE), which can use 36-bit for addressing RAM. Microsoft never took advantage of going beyond the 32-bit address space for Exchange in the way that SQL Server did. With Exchange 12, this will change, because it will be a true 64-bit application, with ample Virtual Address space (8TB).

The speed of access to memory is also a critical performance factor and we have seen a continuous increase of data transfer rates between processors and memory banks from the 320MB/s of 10 years ago to 6.4GB/s today. To accomplish this speed increase, Intel used the North Bridge chipset while AMD uses HyperTransport.

`Going to 64-bit`—With Windows servers we typically see several architectures:

- The 32-bit architecture.
- The 64-bit extended architecture (EM64T, AMD64), which enables the processor to operate in either 32-bit legacy or 64-bit mode.
- The 64-bit Itanium Family (IA-64), which provides a complete 64-bit processing environment.

The move to 64-bit Exchange requires updated management software and readiness of all the tools and applications that make up an Exchange infrastructure, such as deployment tools, patch management and monitoring software. In return, using a larger address space will enable Exchange to use more RAM, have more data readily available, and depend less on the storage subsystem.

`Clustering`—Clustering with Exchange has been a long lasting love-hate relationship. The Active/Passive mode imposed with Exchange 5.0 and, later, with Exchange 2000 was never impressive. Let's admit that Exchange does what it can in a clustered environment that doesn't allow sharing of files between nodes.

Clusters are useful for dealing with system maintenance and failure: they allow monitoring of the resources that the EVS requires, and if one or more is unavailable, the cluster resource manager will attempt to take corrective actions, such as moving the resources to another available node. With Exchange, clusters help in partitioning the overall workload across a set number of nodes.

`Virtualization`—For demonstration purposes, integration testing, functional testing, and development, virtual servers bring a unique advantage: you make better use of otherwise underused hardware by physically consolidating several operating system instances on a single powerful server.

Not only do you get a better CPU utilization for your server, but you also get more flexibility. I/O restrictions are a traditional problem with Exchange 2003 in virtualized server environments. However, with the advent of E12 and its reduced I/O footprint, this is likely to become less of a problem.

STORAGE: MORE FOR LESS

Quiz: What has grown 500 times in the past 10 years? The capacity of a disk drive! I could also have answered: The size of my mailbox. To satisfy the needs of business users, you need to offer fast and large storage subsystems and assume that you'll need 100 percent growth over the next 18 months for your deployment.

Not all environments grow that fast, but many do. In fact, as messaging becomes more relevant to all kinds of business and communication, the need for Exchange to operate multi-gigabyte mailboxes is here—and we look for E12 to address that need.

The challenge with storage is that adding more, larger drives can solve an overall capacity problem, but it increases a management problem. How can you ensure that the large quantities of data attached (physically or logically) to a particular server can be backed up and recovered quickly and can be reassigned to another server if the owner server fails?

In fact, this problem has no single answer. Nor it is unique to Microsoft Exchange. So the storage technology has evolved to address the storage management needs. As a result, we see major progress made in two areas:

- Networking: moving away from server-tethered storage
- Backup and recovery: the ability to back up and recover information rapidly, regardless of size

`Networking`—Beyond capacity, storage has been hit by the network paradigm shift. Long gone are the days where the only way to provide capacity for a server was to attach disks on a back-end bus. Disks are now grouped and controlled, managed by storage arrays, connecting into specialized networks (SAN), or more generalized infrastructures (NAS, file sharing over IP). For Exchange, this means your databases can be rapidly reassigned to another server, without physical intervention on the server.

What has grown 500 times in the past 10 years? The capacity of a disk drive!

At the same time, the way of providing storage-enabled clustering technology is developing. The first clusters were built on a shared SCSI interconnect, which often had restrictions due to the physical characteristics of the SCSI media. With the adoption of networking technology, we can now build flexible infrastructures that use reliable media and interconnects. The deployment can span multiple sites facilitating disaster tolerant Exchange configurations.

The flip-side of networking enhancements is that complexity was added to the infrastructure. An Exchange server now depends on a variety of components, not only at the application level (dependency on the network and other services such as Windows Active Directory), but also on the hardware level. Make friends

64-bit is here. You need to prepare for it sooner rather than later.

with the storage administrators that manage complex SANs, because you need to obtain the storage capacity and throughput that Exchange demands.

We deal with this complexity using best practices and improved technology. Getting best practice in place is a continuing challenge. To solve this problem, Microsoft is taking a different approach for E12, by relying less on the intrinsic features of storage networks and relying more on application functionality. The goal is to make Exchange more flexible and less demanding on the storage components, such that you can reach a satisfactory deployment with less experience and knowledge in storage networks.

This trend, which consists of simplifying the operation of complex components, applies to many other components of a server: automatic monitoring, component discovery, fault analysis and correction. To be effective, they need to be applied in a concerted way and many Exchange features are provided through the operating system. A good example of this is the introduction of the Volume ShadowCopy Services (VSS) in the Whistler code base (Windows XP, Windows Server 2003), that we'll discuss later.

Today, IP SANs are gaining ground by their versatility in deployment. For low to moderate workloads, you can establish a common data and storage network infrastructure using commodity components such as Ethernet infrastructures and gigabit Network Interface Cards (NICs). This might not be appropriate for Exchange 2003, as we know it today. But as E12 reduces its I/O footprint, IP SAN technology enables clustering and low-cost storage networking: this should not leave you indifferent.

Backup and Recovery—Backup and recovery is the number one concern for any Exchange deployment that necessitates large quantities of data. Ten years ago we dealt primarily with Digital Audio Tape (DAT) or Digital Linear Tape (DLT) technologies. Many vendors offered many variations: the challenge was to choose the right technology to match capacity requirements. Email was not business-critical at the time and email downtime of one or two days was relatively insignificant for most companies. Today, we can't cope with more than one or two hours of downtime: the business processes is adversely affected. To match the data growth and fast recovery capabilities, we have classic approaches:

- High-speed interconnects to fast tape libraries: by far the most common approach, which can deliver in excess of 100GB/h of backup and recovery speed.
- Fibre Channel attached Virtual Library Systems (VLS): they are disk-based libraries that behave exactly like tape libraries by emulating the most common models. They can therefore integrate more rapidly into an existing backup and recovery environment. A VLS is typically a commodity server with a specialized operating system and dedicated interfaces for the throughput required in high-end backup and recovery environments.

Faster than fast: Instantaneous—The idea of being able to back up and recover large quantities of data within seconds is not new and has in fact been in use long before Exchange hit the market. However, for Exchange to truly benefit from instantaneous backup and recovery we need proper interfaces to the back-end storage management functions, such as cloning and snapshot functionality, through the operating system.

The principle relies on the ability of the back-end storage array to implement features that can duplicate data and replicate changes from one volume to another. Information is written twice, in independent volumes: the primary volume and secondary volume. If the primary volume fails, you need to present the secondary volume, which will recover the application to the state it was at the time of failure. There is no dependency on size: presenting a 1TB volume is as fast as presenting a 20GB volume.

For Exchange, the important point is to make sure that the application is not issuing transactions while the mirror is broken or while you interrupt the primary to secondary volume synchronization. Windows Server 2003 VSS notifies Exchange of the intent to perform such operations and Exchange, in return, suspends any write activity while the back-end array disconnects the two volumes.

SUMMARY

From the 1996 Pentium-II to the 2006 quad dual-core processor servers; from the 1GB to the 500GB disks, there has been a constant increase in data to process, processor capabilities to process data, and storage to store this data.

The predominance of the x64 processors in modern servers should enable rapid adoption of 64-bit computing by the ISV, not necessarily from the customers that typically depend on precise hardware refresh cycles (six or 12 months) for adoption of new technology. However, just as the industry moved from 16-bit to 32-bit computing, it will move from 32-bit to 64-bit computing and the best you can do about it is to seriously prepare for it, sooner rather than later.

On the storage front, indexing and searching through masses of information becomes an increasing priority. With legal constraints sometimes applied to information and rapid growth of data volumes, you must be able store and retrieve information quickly and efficiently. Indexing content is one way of achieving this.

Overall, it is no exaggeration to say that an Exchange deployment is critical to the business of modern companies. At the same time, the cost of IT has to come down, for the companies to focus on their businesses. The challenge is here to build dependable servers for Exchange that are affordable. Any takers? ■

Pierre Bijaoui is a senior solution architect for Compaq's Applied Technologies Group in Sophia-Antipolis, France.

AMD

The AMD Opteron™ processor with Direct Connect Architecture is engineered to perform with your company's e-mail computing needs in mind. It enables you to exploit the features of Exchange—giving you the performance and power efficiency you require to meet the most urgent business needs. And while deploying other 64-bit processing solutions might force you to retool, recode, and redeploy, AMD Opteron™ processors are on your side, reducing complexities in an all-too-complex IT environment. Power your business with AMD Opteron™ processor-based solutions today.

AMD 64 Opteron

http://enterprise.amd.com

The Evolution of Exchange Clients

by Eric Legault

Microsoft Exchange Server has had a long and varied history as a messaging and collaborative solutions platform. From the early groundwork laid by Exchange Server 4.0 in 1996 to the upcoming Exchange 12, in-house developers and Independent Software Vendors (ISVs) have leveraged a range of different technologies to extend Microsoft's flagship groupware offering. Having lived through many back-end server APIs, mail client customization solutions, and Web application designers, developers and customers alike have seen the best and the worst.

Going all the way back to Exchange 4.0, the focus of the initial toolset was around OLE Messaging 1.0 (also known as CDO 1.0), Extended MAPI, and customizations of the Exchange Client (code name "Capone"). The primary aim of the OLE Messaging object model was to allow developers to add mail and messaging functionality to custom applications designed with an OLE Automation controller, of which there were many. Example programs include Visual Basic/C++/FoxPro, and even the first Microsoft Office products that had support for Visual Basic for Applications (VBA)—Access, Excel, and Project. However, OLE Messaging was limited primarily to working with information stores, messaging objects, and attachments only. MAPI, on the other hand, provided full access to the underlying Windows Messaging Subsystem (WMS) but was an extremely daunting API to work with for all but the most advanced programmers. Another shortcoming at this early stage in the evolution of the product was a way to access Exchange via Internet Information Services (IIS), so running messaging applications in a Web browser would have to wait for such support in the future.

EFD—NOT READY FOR PRIME TIME

The sole mechanism for extending the Exchange Client was through use of the Exchange Forms Designer (EFD) to build custom forms. An early implementation of what eventually became custom Outlook forms, this technology was a good first try but not quite ready for prime time. The EFD provided no support for the VBScript that later Outlook developers were able to use and was restricted to designing only custom email or Post forms. However, at least opportunities were available for developers to begin creating groupware applications such as discussion forums, bulletin boards, tracking systems, and electronic help desks. Programmers using competing applications from rivals such as IBM's Lotus Notes were already creating similar solutions with abundance (some would say very painfully and with dubious results), but at least the collaborative playing field was healthily competitive and becoming more and more innovative.

A key disjoint with the Exchange Client was the lack of support for calendar functionality. To fill this void, Schedule+ took over where the Exchange Client left off. Originally intended as a companion to the Exchange Client, Schedule+ was later released with Office 95 and Windows 95 and helped users to not only schedule appointments, but also to manage tasks, create to-do lists, and organize contacts. Like the Exchange Client, Schedule+ also functioned as an OLE Automation server, but with an even deeper programming model. Regardless, it's clear that a union of the Exchange Client and Schedule+ applications was necessary and this ultimately led to the creation of Microsoft Outlook 97.

Up to now it can be said that the development platform for Exchange 4.0 appeared immature and the toolset incomplete at best. In retrospect, however, the framework was still useful and a robust one to build on in the future. Exchange's infrastructure provided a semi-structured database engine for data access, easy lookups with a built-in directory service, and it shielded developers from having to implement their own security model by exposing access control lists (ACLs). What is also noteworthy is the fact that the Exchange 4.0 Directory gave the product the distinction of being the key building block for Microsoft's Active Directory service introduced in 1999 with Windows 2000.

EXCHANGE 5.0 BROUGHT IMPROVEMENTS

Fast-forward one year to 1997 and the release of Exchange 5.0. Continuing on

the foundation laid by its predecessor, an introduction of new development, server, and client enhancements paved the way for drastic improvements in Microsoft's collaborative framework. OLE Messaging adopted the new moniker of Active Messaging (aka CDO 1.1) and finally introduced support in its object model for addresses and address lists, as well as new collections of Field objects that allowed for easier access to named object properties. The biggest change, however, was the ability to work with IIS 3.0 as a server-side component via Active Server Pages (ASP). Now developers could create custom Web applications to work with data in the Exchange information store. Script-based solutions using Active scripting languages such as VBScript or Jscript could now benefit from automatic compilation as well as the performance enhancements of server-based execution. Furthermore, Active Messaging introduced an implementation in its object model for rendering applications using Active Messaging HTML (AMHTML), which essentially sowed the seeds for Outlook Web Access later introduced with Exchange 5.5.

Arguably, the most notable development with the release of 5.0 was the concurrent introduction of Outlook 97. Microsoft finally had a unified messaging tool and full Personal Information Manager (PIM) in one package, and positioned Outlook as the client fully optimized for Microsoft Exchange Server 5.0 and the defacto upgrade to the Exchange Client and Schedule+ applications. Outlook 97 successfully combined email, calendar and scheduling features, contact and task management, journalizing, and custom forms-based applications into a single interface. Yet what really stood out with Outlook 97 were improvements in designing custom message forms. This once awkward solution had now matured to the point of being very intuitive to learn and had grown in power with support for VBScript. This new Forms3 (Forms Cubed) Engine has certainly proven to have impressive staying power and is still being used relatively unchanged up to Outlook 2003. Solutions built on these forms have also been adopted quite widely, if not completely enthusiastically. Rare is the Exchange environment indeed that has not implemented or at least piloted a variety of custom Contacts forms or Post items in mailboxes or Public Folders in some manner. Finally, rounding out the Outlook extensibility toolset was Exchange Client Extensions. Using COM-based interfaces and developed with Visual C++, this technology became a key driver for many custom Outlook enhancements delivered by many third-party software vendors.

EXCHANGE 5.5: A BOOST FOR DEVELOPERS

While the release of Exchange 5.0 was significant, it was overshadowed that same year by the "incremental" release of Exchange 5.5. With drastic improvements to the plumbing in terms of increased database capacity, support for clustering, and new connectors this release was more than a cobbled together package of patches and it contained significant changes for developers. The primary Active Messaging API was renamed Collaboration Data Objects (CDO) 1.2 (with new calendaring and scheduling features), and it became the stalwart CDO 1.21 version still in use today with the 5.5 Service Pack 1 upgrade. A "lite" flavor of CDO, called CDO for Microsoft Windows NT Server (CDONTS), was also introduced and it gave developers the opportunity to create SMTP-based messaging applications primarily for bulk mailing or mail-enabled Web servers that didn't require the full Exchange feature set. New support for LDAP 3.0 allowed developers to manipulate any object in the Exchange 5.5 directory via Active Directory Services Interfaces (ADSI).

Perhaps most significant, however, was the introduction of Exchange Server Scripting and Routing. Using the Exchange Event Service, the Exchange Scripting Agent and a COM-based routing engine, a toolset was born for working with events in messaging folders. Working closely with CDO, programmers could now use any Active scripting or COM-based development language to create routing/approval applications better known as workflow. Another notable programming tool included with 5.5 was a Rule COM Component to extend and manage folder rules. Finally, the CDO Rendering Library (CDOHTML) replaced AMHTML, and in conjunction with the Exchange Form Design Wizard it was now possible to extend Outlook Web Access interfaces.

With all the action in Building 43 on the Microsoft campus in the late 1990s, it's no wonder there was a gap of nearly three years until the release of Exchange 2000. The Exchange team needed all that time to design a plethora of dramatic new enhancements to the development platform illustrated by many expansions to CDO. CDO for Exchange 2000 Server Objects (CDOEX) facilitated the creation of collaborative applications; CDO for Exchange Management Objects (CDOEXM) provided tools to manage mailboxes and Public Folders; and CDO for Workflow (CDOWF) grew out of the initial work done with the Routing COM components released with Exchange 5.5. In addition, the release of Windows 2000 caused support for CDONTS to be deprecated, requiring a companion API coined CDOSYS that supported both server and client based applications as well as exposing new server events. Lastly, for Windows 2000 Servers, SMTP now exposed transport events that messaging applications could hook into.

A NEW STORAGE TECHNOLOGY

The most prominent change, however, was the transformation of the Exchange database store into the Web Storage System. This new storage technology combined the features and functionality of the file system, the Web, and a collaboration server through a single location. Incorporating support for OLE DB providers and ActiveX Data Objects (ADO), as well as the ability to use the HTTP protocol with enhancements supplied by the WebDAV specification, helped to streamline data access. New services exposing synchronous, asynchronous, and storewide Web Storage System Events (also known as Exchange Event Sinks) created additional mechanisms for designing more robust workflow and real-time

collaboration solutions. Microsoft was confident enough in the changes made to the underlying structure of the platform that it used them as the foundation for its newest collaborative solution, SharePoint Portal Server 2001. SharePoint used the Web Storage System as its data repository and introduced the Publishing and Knowledge Management Collaboration Data Objects (PKMCDO) programming interface. This API borrowed some workflow concepts from CDOWF but contained many new objects for searching and document management operations. Although the messaging and collaboration workspace now appeared to be getting rather crowded, Microsoft was obviously committed enough to Exchange as a serious development platform that many ISVs and other solution providers were quick to leverage this rich foundation through their own integrated offerings.

While all the furious changes in the back-end may have overshadowed the client side of things, Microsoft did manage to release Outlook 2000 along with the rest of the Office 2000 suite with much fanfare. Nothing really changed in regards to custom forms, but macro warriors were now diving into the new support for VBA and quickly creating ad-hoc messaging solutions in many Exchange environments. Similarly, an additional extensibility model became available via COM Add-Ins, giving developers who may have shied away from the complexities of Exchange Client Extensions a more palatable approach to rapidly developing professional and marketable custom solutions.

BUILDING ON A SOLID FOUNDATION

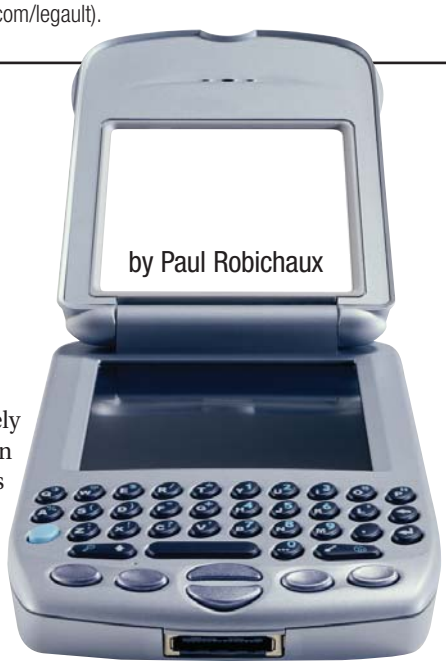
Indeed, the intense efforts of the previous years paid off for Microsoft in terms of a very solid product. Customer adoption was abundant and many vendors offered a slew of integrated products and tools. When Microsoft released Exchange 2003, many people may have been disappointed (or pleased) that there were essentially no significant changes to the development platform. Only minor differences in APIs differentiated the two versions, and with surprisingly nary a new CDO flavor in sight some might have said that Microsoft was resting on its laurels (or had exhausted the possibilities of new acronyms to suffix CDO). Behind the scenes was a different story. The Exchange team had been hard at work on Exchange Server Objects (XSO), a new managed API that was to take advantage of the blossoming .NET Framework. Using a remoteable, server-safe object library via an XML-based interface, XSO was supposed to permit outside applications the ability to populate Exchange data without requiring any use of CDO. Unfortunately, Microsoft dropped this promising technology to the Campus cutting room floor and shelved it for the time being.

Moving forward, Exchange's development platform appears to be undergoing rejuvenation. No doubt stung by criticisms of excessive APIs, Microsoft is unifying them completely for Exchange 12. All object models are being rebuilt from the ground up with managed .NET code, and Exchange itself will be using the same interfaces internally. No older APIs will be getting new features, and architectural changes will force some out completely but with continued support for Exchange 2000/2003. Microsoft's strategy appears to be focused on highlighting platform convergence around Exchange's key competencies of management, transport/protocol layers and data storage. The new Monad system management engine will aim to ease administration with flexible and powerful Cmdlets. A rebuilt SMTP stack based on managed code is now designed specifically for extensibility while an Agent API will provide enhanced access to manipulate and manage mail flow. Finally, fusing store access via Exchange Web Services and the long-awaited integration with Visual Studio .NET will do much to gain the respect of modern developers already comfortable with this toolset. Trimming the fat and flexing its key strong points appears to be a sound strategy by Microsoft for the future of Exchange as it matures, a future that is increasingly looking like an innovative new era for Exchange development. ■

Eric Legault (B.A., MCDBA, MVP - Outlook) is a software consultant for Imaginet Resources Corp. (http://www.imaginets.com), a Microsoft Certified Gold Partner in Winnipeg. He maintains a blog at Office Zealot (http://blogs.officezealot.com/legault).

The Evolution of Mobile Devices

by Paul Robichaux



The messaging world has changed a great deal since 1996. Back then mail systems were primarily focused on LAN mail. Internet mail was still a rarity at most companies and the spam, virus, and phishing problems that we face today were barely on the horizon. The changes we've seen during the past 10 years are perhaps most obvious when we look at the field of mobile and wireless email—

See your e-mail. Send your e-mail. Get your e-mail.

Quest Availability Manager for Exchange eliminates the evils of Exchange outages.

Quest Software has addressed the evils of outages with a solution for fast, reliable, always available e-mail. Switch users rapidly and automatically to a defined Exchange server. Provide users ongoing access to cached messages. Move users back to their original server without data loss after the failed server/store has been restored.

No more fooling around with e-mail when outages occur. Keep your critical communications flowing with continuous access to e-mail with Quest—Microsoft's 2004 Global ISV Partner of the Year.

Learn how to ensure critical send/receive e-mail availability. Get your free white paper titled: *Exchange High Availability: Patterns and Practices.*

Visit www.quest.com/seeyouremail for your free white paper!

Application Management | Database Management | **Infrastructure Management**

QUEST SOFTWARE®

Ten years ago, there was effectively no such thing as wireless access to email.

something that didn't exist in 1966. To see how far technology has advanced since then, we have to look at three trends in parallel: the development of mobility features for Exchange, the growth of both capacity and ubiquity of wireless communications, and the growth of a broad ecosystem of mobile and wireless devices.

Before we can dive into these three interconnected areas, though, we need to distinguish between "mobile" access and "wireless" access. Ten years ago, there was effectively no such thing as wireless access to email, but we all know what that means today: access to messaging and calendar data without a fixed physical connection. "Mobile" access is a little harder to define, but it's fair to say that we should count any access by users who can change their location at will.

1996: EXCHANGE 4.0

Microsoft shipped Exchange 4.0 in June of 1996, at a time when the wireless and communications infrastructure was vastly different than it is today. Relatively few companies offered any kind of remote access to their networks, and those that did overwhelmingly did so via dial-up modem. Cellular phones were still relatively expensive, and the dominant standard in North America was the Advanced Mobile Phone Standard (AMPS)—the analog system that has now largely disappeared. The Global System for Mobile Communications (GSM) standard was beginning to be widely deployed in Europe; it originally offered a blazing 9.6kbps data connection speed (better than AMPS, which had no digital data capability). For the majority of the world, that meant that there was effectively no mobile email, although savvy GSM users began using the Short Message Service (SMS) for quick communication.

The hardware needed for mobile and wireless access was still in its infancy, too. Apple had shipped the Newton in 1993 to mixed reviews (well, OK, they were mostly bad) and USRobotics had shipped the first of the Pilot line earlier in the year. The 1996 crop of PDAs occasionally had provision for an external modem, which was most often used for connections to dedicated private systems like Compuserve (remember them?) or to work networks.

What about Exchange? Well, Exchange 4.0 didn't incorporate anything remotely resembling wireless or remote access capability. It had an SMTP server (in fact, it was an open relay and relaying couldn't be turned off), but it had no Web access and its POP support was limited. At the time, most Internet email users preferred programs like Qualcomm's Eudora or the free pine mailer to Outlook, which was brand new and only available as part of the Office 97 suite.

1997: EXCHANGE 5.0

1997 brought us Exchange 5.0, a significant improvement over Exchange 4.0 in terms of stability and features—but it still didn't include any mobility support. With suitable dial-up connections, Outlook could be used from a laptop to reach an Exchange server. Exchange 5.0's Internet Mail Connector (IMC) was a major improvement over the Exchange 4.0 version, but we still had no effective Web access. The email landscape was definitely shifting as more companies added the ability to exchange Internet email with outside organizations; in addition, the ongoing browser war between Microsoft, Netscape, and a host of smaller companies helped fuel the widespread deployment of Web browsers—which in turn led to the deployment of first-generation Web applications for electronic banking, travel management, and so on.

In the communications world, AMPS was still dominant and the major worry that most consumers had was that their phone would be cloned; however, the emerging Code-Division Multiple Access (CDMA) standard was starting to offer competition to GSM for digital data service, with up to 64Kbps data speeds. Unfortunately, no integrated phone/PDA devices were on the market and laptops were still relatively heavy and bulky, although they were adequate for running the contemporary crop of mail and PIM programs (Outlook 97 chief among them).

1998-2000: EXCHANGE 5.5

In the summer of 1998, Microsoft shipped Exchange 5.5, which brought some key improvements for mobile and wireless users. The biggest feature from that standpoint was the debut of Outlook Web Access (OWA), a purpose-built Web client that allowed users to access their email and calendar data from Web browsers across the Internet. OWA 5.5 could run on a separate server, and it used MAPI to request user data from the information store. This made it somewhat slow and finicky, but because it could run on a separate server, most users didn't complain. Exchange 5.5 also included much better POP and IMAP support, as did Outlook 2000; because Windows NT included software for offering virtual private network (VPN) connections, many organizations running Exchange began to replace their expensive, finicky dial-up modem banks with VPN capability.

CDMA and TDMA mobile phones were making larger inroads in the communications market and in 1999 Palm shipped the Palm VIIx, one of the first PDAs that included a built-in wireless modem. While the VIIx didn't catch on because of its bulk and cost, another company shipped a breakthrough product: the Research in Motion Inter@ctive Pager, which combined two-way paging capability with a built-in thumb-sized keyboard. RIM was a primary provider of wireless data services, and the company started selling devices to spread its service. In 1999, RIM began offering email via its paging service, and that service—and the associated hardware—took on the BlackBerry brand that's so familiar today. During this time, Microsoft started moving aggressively to establish a version of Windows as a viable platform for data-capable PDAs.

2000-2003: EXCHANGE 2000 SERVER

On November 29, 2000, Microsoft shipped the long-awaited successor to Exchange 5.5: Exchange 2000 Server. The new version contained too many functional improvements to list here. From a mobility standpoint, the big news was that it completely replaced the Exchange 5.5 version of OWA with a new version that used a new Exchange system component to pull data directly from the information store. This made OWA much faster and gave it better scalability. In addition, the Web client gained a great deal of capability compared to the 5.5 version, and there were a number of stability and performance improvements in the POP and IMAP engines.

By the end of this period, most companies offered some kind of mobile access to their email systems, most often via VPNs. When Microsoft shipped the Internet Security and Acceleration (ISA) Server product, it included the ability to securely publish Exchange servers for RPC access directly to the Internet, enabling secure access from Outlook clients without the need for a VPN.

What about mobile access? This period also marked the emergence of a real market for wireless data-capable PDAs. These devices, which ranged from the large and heavy Kyocera 6035 to better BlackBerry devices to a wide range of Pocket PC Phone Edition devices, could take advantage of improvements in the worldwide wireless data network—cellular carriers were rapidly deploying updates to their systems during this period, which resulted in better coverage and faster data access. Microsoft helped make these devices more valuable by providing them access to email; in 2001, Microsoft shipped the first release of the Microsoft Mobile Information Server product—MMIS—(followed by a second release in May of 2002). MMIS provided a way for wireless devices to synchronize data directly with the Exchange server; of course, at roughly the same time, RIM started shipping its BlackBerry Enterprise Server product, which quickly took a dominant market position.

2003-2006: EXCHANGE SERVER 2003

This period marked the emergence of a truly useful set of mobile and wireless data capabilities for Exchange. This happy state of affairs came about through several interlocking factors:

- The improved feature set of Exchange Server 2003, which included mobile access for cell phones (through Outlook Mobile Access, or OMA) and Windows Mobile devices (through Exchange ActiveSync, or EAS). Together, OMA and EAS represented a superset of the functionality available in MMIS, and because they were included as part of the base product Microsoft could claim some credible savings over competing solutions.
- The widespread deployment of fast wireless services, along with devices that could take advantage of them. UMTS and EV-DO offered burst speeds of up to 4Mbps, making them useful for sustained communications, and cellular carriers around the world raced to see who could deploy these technologies first and most widely.
- Competitive pressure in the mobile device market. A market that was formerly dominated by a few large players (Motorola, Ericsson, and Nokia come to mind) was stirred up by the entry of a large number of fast-moving mobile device manufacturers, including HTC, Palm, Kyocera, and Samsung. These emerging vendors helped do to the mobile device market what happened to the laptop market in the same time period: foster an explosion of device types and capabilities at different price points.
- The emergence of mobile information workers as a key part of many companies' business operations. Traveling sales representatives, field service technicians, and other job roles that require access to mail from anywhere helped fuel the adoption of these technologies.

Perhaps unsurprisingly, users started to clamor for mobile and wireless access, and this helped drive its market penetration as well. In 2005, Microsoft shipped Windows Mobile 5.0 and Exchange Server 2003 SP2, which in conjunction with the Mobility and Security Feature Pack (MSFP) add a number of desirable mobility features to Exchange (including true "push" email, which Microsoft calls Direct Push). As MSFP-capable devices continue to enter the market, expect to see more pressure to deploy these devices in your organization. In addition, the Exchange team has scored some coups by licensing the Exchange ActiveSync protocol to Motorola, Nokia, Sony-Ericsson, and PalmOne—the leading manufacturers of PDA/phone devices.

THE FUTURE

I remember reading science fiction stories in the 1970s in which characters had small, portable computers with ubiquitous communications; a little more than 25 years later, we have many of those capabilities available at a relatively low per-device cost. Microsoft has been very clear that it views mobile and wireless access as a key technology for Exchange; the company's investment in Windows Mobile and related technologies (including the Exchange 2003 SP2 DirectPush features) is a strong indicator of this. Microsoft has been fairly closemouthed about the mobility features in Exchange 12, but we'll learn more about those in coming months; and I expect them to generate a lot of discussion. ■

Paul Robichaux is a principal engineer for 3sharp, an MCSE, and an Exchange MVP. He is the author of several books, including *The Exchange Server Cookbook* (O'Reilly and Associates), and creator of the http://www.exchangefaq.org Web site.

2000-2003 marked the emergence of a real market for wireless data-capable PDAs.

Ten Years of Outlook Web Access

by Kevin Laahs

Some readers may immediately suspect the validity of the title of this article because Outlook Web Access (OWA) did not exist in Exchange 4.0. However, it was first released with Exchange 5.0, which actually shipped in November 1996; therefore, OWA is, legitimately, (almost) 10 years old.

I find it hard to imagine that we've had a Web-based client to Exchange for 10 years. I recall that when I first set up OWA 5.0 most of my customers were still trying to come to grips with Outlook as they migrated from other mail systems to PC-based systems. The thought of doing mail "on the Web" was completely alien to them then.

Of course, that is clearly not the case these days because everyone from grandchildren to grandparents freely uses Web-based mail systems such as Hotmail, GMail, and Yahoo. Their success was difficult to predict 10 years ago but the ubiquitous Internet and movement toward industry-standard protocols such as SMTP and HTTP has made it all possible. And some ISPs actually use OWA as their mail client of choice, so we see its usage in the consumer and the business space. So let's take a look back at how OWA has evolved into being not only a first-class client in its own right but also the perfect companion for those times when you can't, for whatever reason, run Outlook.

FROM WHENCE WE CAME

Exchange has always supported a broad range of clients—from the original Exchange client (known as "Capone") to a multitude of lightweight clients using protocols such as POP3, IMAP4, and HTTP to connect to the server (e.g., Outlook Express, Eudora, and Pegasus). The Exchange client shipped on the Exchange CD and was the only client developed by the Exchange team until OWA came along. Indeed, the Exchange client was quickly superseded by Outlook so OWA is, to this day, the only fully featured mail client developed by the Exchange team.

To date, there have been two generations of OWA with completely differing architectures. The first generation shipped as an optional product to Exchange Server 5.0 and 5.5. The name given to the ability to view your mailbox through a browser in 5.0 was Outlook WebView, but this was changed to Outlook Web Access with the release of 5.5.

The second generation delivered with Exchange Server 2000 and 2003 embeds OWA functionality into the product itself. Thus, there is no separate or optional installation but you can switch off support for HTTP, effectively disabling OWA. This generation of OWA actually delivers two different "flavors," currently known as the premium and basic versions. We'll look at the different architectures shortly but let's first look at why a browser-based client came into existence in the first place.

Exchange is an enterprise messaging server and, as such, everyone in an enterprise should be able to make use of its services. However, the clients that initially shipped with Exchange Server were all Windows based (although we did eventually see Macintosh support). This meant that non-Windows users (of which there were plenty in those days!) were left out and could not be a part of the enterprise messaging system. Furthermore, back then, a lot of desktop PCs were not powerful enough to run the latest client applications.

Browsers were the most common factor among differing operating systems. Having the ability to process email regardless of operating system platform meant the whole enterprise could join in the fun. Browsers freed the client desktop of any processing other than rendering the HTML that was generated by the server. Therefore, the ability to process email became desktop agnostic and contributed to the ease of deploying messaging for all without the need to worry about configuring client-side settings such as Messaging profiles.

The use of a browser also freed the user from being tied to a single desktop where their client applications were installed and allowed them to roam different devices and still be able to process their mail. In essence, if you could lay your hands on a browser, you could get to your mailbox. Thus, Exchange Server could quickly satisfy the needs of an enterprise without having to do expensive desktop replacements or infrastructure upgrades. Of course, these days it would be unthinkable to not have browser-based access to your information!

OWA 5.X ARCHITECTURE AND FUNCTIONALITY

The basic job of a browser is to receive HTML pages from a server and render them for viewing. As such, there has to be some process on the server that generates the relevant HTML for the task in hand—be it listing the contents of your Inbox folder, viewing a calendar item, or reading a message.

The server component that does this for all versions of OWA is Internet Information Server (IIS). The main difference in the architecture between the first and second generation of OWA is in the way that IIS talks to the Exchange Store to retrieve the requested data and how it wraps that data in suitable HTML before handing it back to the requesting browser.

Active Messaging was the term used to describe the overall server components required to enable IIS to send and receive information from Exchange. Active Messaging was essentially an ISAPI extension as we know it today, allowing IIS to be extended with application-specific functionality. When browsers direct their URL at the virtual Exchange application running on IIS the extension takes over and handles the communication with Exchange.

With OWA 5.x that communication was essentially a MAPI connection from the server that was running IIS to the Exchange Server. To all intents and purposes, the Exchange Server would see this connection as it would an Outlook client application and process the request accordingly. This communication was therefore handled using RPC calls. A series of Active Server Pages (ASP) with supporting files (such as .gif files for graphics and stylesheets for formatting) wrapped the data in the correct HTML and passed it back to the browser. The ASP pages were essentially the source code for the OWA client and some brave souls modified these to add functionality and to do corporate branding.

In the early days of OWA, the requirements on the browser were minimal because it only had to handle the display of static HTML pages. The ability to handle Javascript and frames was basically all that was required to run OWA.

While you didn't have the same functionality as an Outlook client you certainly had enough to do basic processing of your mailbox, as well as to access public folders. OWA 5.5 introduced the ability to handle calendars and 5.5 SP1 added extra features such as check names, managing contacts, and the ability to change your NT password.

This first attempt at a browser-based messaging client was not a bad result, except for one thing. The architecture couldn't scale very well and, with the success of Exchange seeing ever growing populations, something had to be done about this. The reason for the lack of scalability was the use of MAPI client sessions between the IIS server and the Exchange server. In effect, this basically limited each IIS server to support, at most, about 800 concurrent sessions. Not only were the number of sessions limited, but the performance of ASP pages in IIS made the experience feel a little slow and clunky. I guess it's fair to say that when OWA was first designed the types of user loads that it would ultimately have to support were not entirely expected—so it's no wonder it started to creak under the strain!

With the existing architecture reaching its limits a change had to be made if OWA was to be able to respond to the growing user base and their demand for more and more functionality. Browsers themselves had moved on and could contribute to the end result. For example, the introduction of dynamic HTML (DHTML) meant that Web pages could effectively be changed "on-the-fly" by the browser itself without having to request that the server regenerate the HTML. This functionality offered scope for a far richer browser experience.

OWA 200X ARCHITECTURE AND FUNCTIONALITY

Exchange 2000 heralded the arrival of the next generation of Exchange with a completely new architecture, which is also used in Exchange 2003. Support for Internet protocols (SMTP, POP3, IMAP, and HTTP) were embedded into IIS and OWA was rewritten from scratch. Gone were the clunky ASP pages; all the HTML was now being generated by an ISAPI extension (compiled code) that had high-speed access to the store via ExOLEDB. And this wasn't a remote link between IIS and the Store—this was on each Exchange server, which meant that any Exchange server hosting mailboxes was OWA-ready out of the box.

In addition to a messaging platform, Exchange 2000 was also being positioned as a collaboration platform upon which collaborative applications could be built. It was envisaged that all application data would be held in the Store and Web-based clients would be used to access this data. As it has transpired, Exchange was not successful as an application platform for reasons that are outside the scope of this article. However, many of the core changes introduced for this purpose benefited OWA and also meant that your Exchange-based data was easily accessible from other Web-based applications. For example, every single item in the Exchange Store is accessible via a user-friendly URL and thus views of folders or items can be embedded anywhere you can reference a URL. From sending links to Store-based items to including views of your Inbox or group calendars inside a portal—all this was now possible.

To address the previous scalability issues, Exchange 2000 also supports front-end/back-end deployments. In such a deployment, the stateless load-balanced front-end servers act as a proxy server masquerading to the back-end server as the requesting browser client. This approach has many advantages, including off-loading certain processing from the back-end server (such as authentication) to having much more flexibility when considering the type of deployment commonly seen in ISP environments.

With the advancements made in browser technology, Microsoft had a dilemma in deciding how it could support the plethora of different browsers with their differing capabilities. For example, not all browsers could support DHTML but, given that the browser is meant to be the ubiquitous client, they couldn't stipulate any particular browser.

Therefore, Microsoft decided to support two different flavors of OWA—the reach and rich versions. Browsers announce their capabilities to servers through something called the user-agent header. By interrogating this header, server-side applications can then decide what the browser is capable of and render HTML appropriately. The "reach" client is so-called because it makes no assumptions that the browser can do anything other than render HTML 3.2. Thus, any browser can be "reached." The rich client is so-called because it does assume that the browser can handle techniques such as DHTML and therefore exploits this to offer a richer client experience.

OWA 2000 offers some excellent functionality to end users, including preview panes, resolution of ambiguous names, sorting of items, and a rich text editor. OWA 2003 extends this functionality, particularly in the area of security. One of the goals of OWA 2003 was to make it more secure and more viable to be deployed over the Internet. It introduced several features that aid in this goal, such as forms- (or cookie-) based authentication. This effectively puts a timeout on a browser session such that if it is left unattended it can't be high jacked—a bit like a password-protected screen saver, if you will. Other security features include the ability to disable attachments, support for S/MIME, and the ability to filter out Web Beacons.

SUMMARY

The ability to access Exchange data using a Web browser has come a long way in 10 years. OWA 2003 is an excellent mail client that offers Outlook-comparable features in a robust way. It will be very interesting to see how it evolves as Exchange enters its next generation with the expected release of Exchange 12 in 2007. ■

Kevin Laahs is a principal consultant in the HP Services Advanced Technology Group. He is coauthor of *Microsoft SharePoint Technologies: Planning, Design, and Implementation* (Digital Press).

Changing Times, Changing Protocols

by Kieran McCorry

Long before Microsoft released the first version of Exchange (the successor to Microsoft Mail), Exchange 4.0 on June 11, 1996, the engineers in Redmond had been working for many years on what they hoped would become "the" definitive messaging system. About that time the X.400/SMTP "protocol wars" were in full swing. The rival factions, the International Telecommunications Union (ITU) for X.400 and the Internet Engineering Task Force (IETF) for SMTP, maintained that their chosen protocol was superior to the other. And while each had their technical merits (and demerits) it certainly wasn't clear which would prevail.

Exchange 4.0 was launched, firmly based on the X.400 messaging protocol and a complementary X.500-like directory service. But Microsoft hedged its bets—to some extent—with an SMTP communications gateway called the Internet Mail Connector (IMC). With the benefit of some 10 years' worth of hindsight, it's interesting to look back at the development of Exchange and observe how Exchange has evolved from its solid X.400 beginnings to the SMTP-based system it is today.

X.400: THE CORNERSTONE OF EXCHANGE

The X.400 Recommendations (they're actually recommendations, not standards) were first published in 1984 by the Comité Consultatif International Téléphonique et Télégraphique (CCITT), now called the ITU. (The ISO also published similar recommendations at the same time.) The recommendations were informally referred to as the Red Book because of the color of the cover. The recommendations were reviewed in 1988 (the Blue Book) and then again in 1992 (the White Book).

During the late 1980s and the early 1990s the messaging world was very much reminded that X.400 represented the most comprehensive and reliable messaging protocol because X.400 had a rich set of features that advocates of SMTP could only dream of (at the time). These features included delivery and receipt notifications, better support for non-text messages, rich attachment support, and the concept of the Reliable Transfer Service, which provided for the retransmission of an email from the point of failure after the original transmission had failed. At the time, this latter capability was much lauded because data networks were nowhere near as reliable as they are today.

So Exchange 4.0 was launched firmly rooted on X.400 technology. We saw this at the most fundamental level with Exchange 4.0 because every Exchange user was equipped with an X.400 address. The X.400 address format, although well structured and hierarchical, was cumbersome. Mine was of the form

```
C=US;A=ATTMAIL;P=DIGITAL;
  O=DIGITAL;OU1=BVO;S=MCCORRY;
  G=KIERAN
```

Such addresses were never a favorite for users; fortunately, the

NO VIRUSES.
NO SPAM.
NO DOWNTIME.
EMAIL DONE RIGHT.

No one can promise complete email security and availability. We don't live in that kind of world. Yet one company has earned a worldwide reputation for making email as secure and available as it is important. A company that not only screens out viruses, spam and spyware, but also provides solutions for speedy recovery in case of system failure. A company that reduces storage costs by archiving to secondary storage and blocking unwanted emails. A company that provides management tools for efficient email retention and fast email discovery. A company that does email right. Symantec. Because we know it's not just email, it's your business. For more information visit www.symantec.com/esa or call 800-745-6054

BE FEARLESS.

www.symantec.com/vision

Symantec**Vision**2006

May 8–11, 2006 | Moscone West | San Francisco, California, U.S.A.

symantec™

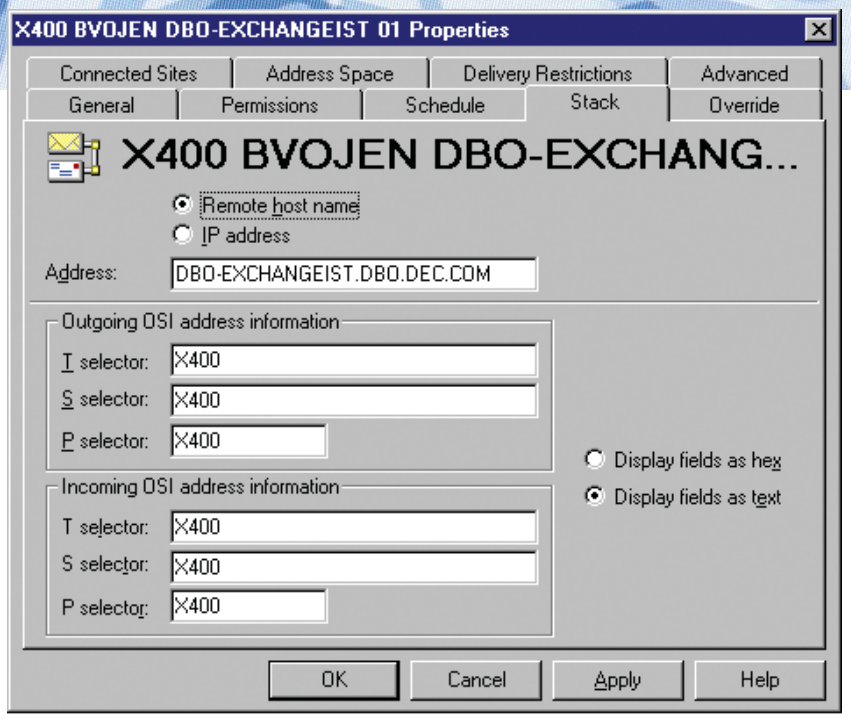


FIGURE 1: X.400 CONNECTOR STACK CONFIGURATION

Exchange Global Address List (GAL) protected users from the inconvenience of having to type, let alone remember, such addresses. X.400 was used as the underlying protocol for the exchange of emails and all emails, whether transferred internally or between servers, passed through the heart of Exchange 4.0: the X.400 Message Transfer Agent (MTA).

The strength of Exchange's X.400 communications stemmed from its ability to work well over poor networks; that is, networks that were unreliable and that suffered from low bandwidth or from high latency. And as a message format, X.400 was quite efficient, especially for complex body-part types such as Word documents, PowerPoint, and Excel. X.400 encoded these document types using ASN.1 format. SMTP relied on Base64 encoding, which is less efficient because a Base64 encoded message often swelled by up to 30 percent or so of its original size. Thus, X.400 Connectors were the obvious choice for wide-area inter-site links in the mid 1990s.

But X.400 Connectors were plagued by complexity. We see clear evidence of this when we look at the configuration requirements for Exchange inter-site communications. How many of you remember having to battle through the configuration of an X.400 Connector between two Exchange servers, defining transport stacks with much-loved OSI terms such as Presentation Services Access Point, Session Services Access Point, and Transport Services Access Point, or their equally cumbersome shorthand forms of P-Selector, S-Selector, and T-Selector. Figure 1 shows a screenshot of an X.400 Connector configuration of mine from a server that my group used to run in Dublin.

Apart from the technical benefits of X.400 over early 1990's networks, Microsoft had another reason for building Exchange 4.0 around X.400. At the time, X.400 was the messaging protocol of choice for large corporate and, especially, governmental organizations. Furthermore, it was the preferred protocol for inter-company communication and various national defense departments had strict guidelines and requirements for X.400 to be used within secure messaging environments. (In fact, a special version of Exchange—Exchange Defense Messaging System (DMS)—was made available). To be taken seriously in this space, Microsoft had to demonstrate its commitment to X.400 integration and interoperability—and Exchange 4.0 was that demonstration.

THE MOVE TOWARD SMTP

While Microsoft developed Exchange 4.0 with X.400 at its heart, SMTP connectivity was obviously a requirement for Internet connectivity: the IMC served this purpose. The IMC was an Exchange Store-based connector, which means that any email messages that either came into Exchange via the IMC or left Exchange via the IMC at some stage passed though the Exchange Store. While this was certainly inefficient, it had some advantages, not least the fact that the passage through the Store meant that addresses on processed email could be modified using the AddressRewrite and ResolveP2 registry keys. It's interesting to note that some of this rewrite functionality associated with Exchange 4.0 was removed with the introduction of Exchange 2000 and didn't surface again until Exchange 2003 SP1 and the availability of the Exarcfg.exe.

The IMC remained very much an add-on component to Exchange's X.400 basis even with Exchange 5.0, despite being strategically renamed as the Internet Mail Service (IMS) and integrated into the Store process. However, it wasn't until Exchange 2000 that Microsoft reflected the sea change in the industry's view of messaging protocols. Exchange 2000 abandoned the primarily X.400-based message transfer protocol in favor of an SMTP-based mechanism. The MTA, responsible for processing all email messages within an Exchange system, which previously used X.400, was replaced with an SMTP MTA at Exchange 2000's core. There was a still a place for the old X.400 MTA, but by now it was relegated to add-on and legacy connectivity.

The move to SMTP with Exchange 2000 was natural and predictable. At the time, many questions were raised about the ability and performance of SMTP versus that of X.400. Microsoft silenced any criticism by demonstrating blistering performance from a message throughput perspective—in excess of that from X.400—and through support for a series of IETF-driven SMTP enhancements such as 8BITMIME, CHUNKING, and PIPELINING and improving security functions within SMTP such as S/MIME. The industry had matured SMTP and it was ready for prime-time deployment; at the same time, the sun was beginning to set on X.400 as the messaging protocol of choice.

The inter-server routing mechanism in Exchange 5.5 was troublesome and prone to failure, often resulting in what became known as message "ping-ponging." With Exchange 2000, Microsoft touted a much-improved Link State Routing model based on Djikstra's Least Cost Routing Algorithm, a model that had already enjoyed considerable success in the physical networking world under the soubriquet of Open Shortest Path First (OSPF).

Unsurprisingly, SMTP has remained at the core of Exchange as Microsoft moved to Exchange 2003. And, indeed, Exchange 2003 brought subtle improvements to the operation of the transport by introducing leaf-node routing. There is little doubt that SMTP will remain fundamental to Exchange as Exchange 12 comes to life. It's perhaps interesting to chart Microsoft's movements with respect to routing models through the various versions of Exchange. Exchange 4.0 through Exchange 5.5 had a routing model that was firmly based on the concept of islands of high bandwidth connectivity, known as Exchange "sites," combining both routing and administrative boundaries. Microsoft expanded this concept to give email system designers more freedom to design flexible routing models of their choice with the concept of Routing Groups first debuting in Exchange 2000 and continuing in Exchange 2003. With the advent of Exchange 12 it's likely that this flexibility will be reduced, because Microsoft intends to use the underlying Windows site topology as the framework for message routing.

THE DIRECTORY IS DEAD: LONG LIVE THE DIRECTORY

From the outset, Microsoft was coy about labeling the Exchange Directory Service (DS). The DS was billed as an X.500-like directory service, but not a true X.500 directory. The positioning of the DS is unsurprising. In the early to mid 1990s, X.500 directories were being positioned as the way forward for directory services and directory integration. However, the X.500 recommendations were complex and had a heavy footprint in terms of underlying OSI transport requirements. The recommendations defined no specific requirements for the internal structure of an X.500 database, so vendors were free to choose whatever database or internal structure they wished, so long as the naming and hierarchical structures of X.500 were externally visible. A quick look at any object in the Exchange 4.0 DS reveals objects with classic X.500 naming structures similar to

```
/O=HP/OU=IRELAND/CN=Recipients/
  CN=KieranM
```

So the Exchange 4.0 DS certainly did provide X.500-like structures, but it relied on its own proprietary protocols for directory access and replication. Quite right, too! The X.500 recommendations described a wealth of sophisticated, difficult-to-implement protocols for directory access, including the Directory Access Protocol (DAP), Directory Information Shadowing Protocol (DISP), Directory Operations Protocol (DOP), and Directory Systems Protocol (DSP). Microsoft, to some extent, ignored these encumbrances in favor of a more efficient mechanism. And that worked well for Exchange. In fact, the ease of management and efficiency of the Exchange DS was arguably one of the big success stories of the product.

Just as the industry was moving away from X.400 protocols for messaging, it seemed that Microsoft, with the Exchange DS, had in some way helped precipitate the move away from X.500 directories, too. The DS remained firmly established as a part of Exchange up to Exchange 5.5, but the introduction of Exchange 2000 brought not only big changes for the messaging protocol, but also for the directory model.

The Exchange DS had evolved as a messaging-specific directory to become the bedrock of Windows 2000 as the Active Directory (AD). To Microsoft, the idea of jettisoning Exchange's own DS in favor of relying on the AD made perfect sense. Why have two directory services, with all of the attendant synchronization and management headaches, when just one, consistent, shared directory between the operating system (for the management of users) and Exchange (for the management of mailboxes) would suffice? On the face of it, this makes sense, but the resulting complexity of migration as Exchange 2000 moved to the AD was—to put it mildly—a headache. Thank goodness (and I say this with only some degree of seriousness) Microsoft provided easy-to-use tools such as the Active Directory Connector (ADC) to ease the transition!

Relying on the AD for Exchange 2000 and then Exchange 2003 was not just a change in technology. The disparate communities of mail administrators and Windows administrators now had to collaborate and work together. Even in 2006, it's questionable whether this massive paradigm shift was been completely successful.

However, one thing is certain. The AD reflects the movement of the industry away from X.500 toward LDAP. The AD is very much an LDAP V3 compliant directory and it has been from Windows 2000 up to Windows 2003's support for advanced LDAP initiatives.

SUMMARY

Looking back at the past 10 years of Exchange's history, a few lessons with respect to messaging and directory protocols are obvious. First, Exchange always demonstrated a clear adherence to standards. We see this with the strong alignment to X.400, and to some extent X.500, with the earliest versions of Exchange up to the present, where Exchange is intimately aligned with SMTP and LDAP. Second, we observe that Microsoft was always eager to push Exchange forward, not just as technology advanced, but as the industry advanced in tune with that technology. Third, Exchange, and Microsoft on the whole with other technologies such AD, was instrumental in extending, enhancing, and progressing the technologies themselves. It's perhaps fitting to close this glance back at Exchange over the past 10 years with this quote from the author and clergyman Henry van Dyke: "Time is too slow for those who wait, too swift for those who fear." ■

Kieran McCorry, based in Ireland, is a Principal Consultant in HP's Advanced Technology Group and a Microsoft MVP. He is a regular contributor to *Windows IT Pro* and the *Exchange and Outlook Administrator* Newsletter.

Exchange Resource

We hope you're enjoying the historical look at Microsoft Exchange that this supplement is providing. We felt that celebrating 10 years of Exchange, noting the advances that have been made, and looking forward to the future would be a worthwhile endeavor and that it would be of value to *Windows IT Pro* and *Exchange & Outlook Administrator* readers. But this supplement would not have been possible without our sponsors: AMD, HP, NetApp, Quest Software, Symantec, Promodag, Neverfail, and Waterford Technologies. We encourage you to take time to review the Exchange solutions that they offer—everything ranging from 64-bit processing solutions to storage solutions to Exchange management and email archiving and reporting solutions.

AMD

I want a server that runs Exchange without compromise.

The AMD Opteron™ processor with Direct Connect Architecture is engineered to perform with your company's email computing needs in mind. It enables you to exploit the features of Exchange—giving you the performance and power efficiency you require to meet the most urgent business needs. And while deploying other 64-bit processing solutions might force you to retool, recode, and redeploy, AMD Opteron™ processors are on your side, reducing complexities in an all-too-complex IT environment. Power your business with AMD Opteron processor-based solutions today.

http://enterprise.amd.com

HP

HP is a technology solutions provider to consumers, businesses and institutions globally. The company's offerings span IT infrastructure, global services, business and home computing, and imaging and printing. For the four fiscal quarters ended April 30, 2005, HP revenue totaled $83.3 billion. More information about HP (NYSE, Nasdaq: HPQ) is available at www.hp.com.

http://www.hp.com

NetApp

Network Appliance, Inc. (NetApp) is a world leader in unified storage solutions for today's data-intensive enterprise. Since its inception in 1992, NetApp has pioneered the development of seamless storage solutions that simplify data management and include specialized hardware, software, and services. Major corporations and service providers, including Citicorp Securities, Lockheed, Merrill Lynch, Oracle, Texas Instruments, and Yahoo!, utilize NetApp global data management solutions. With over 3,000 deployments worldwide, our iSCSI storage solutions for Exchange provide companies with high-performing, cost-effective storage and management capabilities. Sophisticated solutions allow quick database recovery, ensure maximum uptime, reduce costs and expand easily for compliant storage requirements.

http://www.netapp.com/solutions/iscsi/

Guide

Quest Software

Quest Software offers Microsoft Exchange management solutions to help ensure that your email service works efficiently and at maximum performance every day, all day, even during migrations.
Our solutions reduce administrative burden, satisfy end-user expectations and enable you to meet the critical business objectives of your organization's messaging systems. Some of our key products for Exchange management include Quest® Archive Manager, Quest® Availability Manager for Exchange, Quest® Recovery Manager for Exchange, Quest® MessageStats™ and Quest® Spotlight on Exchange. Streamline administration, ensure continuous e-mail availability, reduce costs and build powerful, custom reports with these and other Exchange products from Quest Software.

http://wm.quest.com/WITPResourceGuideForExchange0406

Symantec

With innovative technology solutions and services, Symantec helps individuals and enterprises protect and manage their digital assets. Symantec provides a wide range of solutions including enterprise and consumer security, data management, application and infrastructure management, security management, storage and service management, and response and managed security services. Symantec is the world leader in providing solutions to help individuals and enterprises assure the security, availability, and integrity of their information.

http://www.symantec.com

Promodag

PROMODAG Reports for Microsoft Exchange Server is an advanced reporting tool that comes with more than 130 canned reports on traffic patterns and mailboxes' content. Traffic reports allow administrators to get thorough analysis of traffic patterns, both for internal and Internet traffic, and of the behaviour of individual email users or groups of users. With mailboxes' content reports, administrators can analyze the mailboxes' content by type of items (messages, delivery reports), search and list specific attachments, appraise the number, age and size of the items and access rights to mailboxes. All those features make tracking email traffic possible and ensure better security, business continuity and performance of your organization. Contact information: info@promodag.com

http://www.promodag.com

Neverfail

Neverfail is a leading global software company providing affordable data protection, "cluster-class" high application availability and disaster recovery solutions for the Microsoft technology platform including Exchange, SQL Server, File Server, IIS, SharePoint and RIM BlackBerry. Neverfail's software solutions enable users to remain continuously connected to the live software application irrespective of hardware, software, operating system, or network failures. Neverfail's mission of eliminating application downtime for the end user delivers the assurance of business continuity, removes the commercial and IT management costs associated with system downtime and enables the more productive use of IT resources.

http://www.neverfailgroup.com

Waterford Technologies

MailMeter Archive is the only second-generation Intelligent Email Archiving and Management solution that affordably solves multiple business problems associated with Microsoft Exchange environments, such as email storage management, policy compliance and email monitoring. MailMeter Archive stores, compresses and encrypts email in a separate file system, which reduces the storage burden on your production Exchange server by over 80%. This translates into the elimination of User Quotas, PST file management, and long back up times. Waterford Technologies also provides robust email reporting through MailMeter Insight, and advanced e-Discovery for legal investigations through MailMeter Investigate. Waterford Technologies has over 400 customers globally.

http://www.waterfordtechnologies.com

HP BladeSystem for Microsoft® Exchange Server. 10 Years together, and still transforming the world.

Congratulations to Microsoft for a decade of Exchange Server on HP platforms. The HP BladeSystem for Exchange delivers a high-performance messaging infrastructure for the ultimate in consolidation, security and high availability. All with a lower total cost of ownership. And the smallest data center footprint in the business. Join the celebration at:

www.hp.com/solutions/microsoft/exchange.

hp

invent