Exchange 2007 Availability & Storage

- Exchange 2007: Always On
- Storage Strategies for a New Exchange
- Bonus: Exchange 2007 Reader Review
Exchange 2007: Always On

Exchange 2007 comes equipped to let you use several high-availability techniques.

Most organizations need their data available every second of every day. Unfortunately, computers, networks and storage devices will all fail eventually—no matter how much we pay for them or how closely we monitor them.

Disaster recovery solutions are our most common defense against such technology failures. However, these only let you restore your data back to the point of disaster. Data and time are bound to be lost, and even if the time lost is minimal, time lost is still money lost.

The catch phrase these days for keeping systems running is “High Availability.” The promise of high availability solutions is 24x7 uptime—or, more accurately, no unscheduled downtime.

There are three primary options for high availability with Exchange Server 2007: Local Continuous Replication, Cluster Continuous Replication and Single Copy Clusters.

Understanding these options will give you a better vision of what you can provide your organization when you’re deploying Exchange Server 2007 out of the box. These solutions all provide varying degrees of high availability, so not all solutions are equal—and not all solutions involve clustering. Clustering is often synonymous with the concept of high availability, but it’s no longer an essential ingredient.

First Line of Defense: Transaction Logs

Exchange makes a valiant attempt to provide its own redundancy right out of the box. Part of Exchange’s overall architecture includes storage groups. Each of these storage groups contains several databases. Exchange 2007 Standard Edition lets you create up to five storage groups and mount up to five databases. Exchange 2007 Enterprise Edition lets you create up to 50 storage groups and mount up to 50 databases.

When you install Exchange in the mailbox server role, you’ll find one default storage group, which contains one default mailbox database (typically labeled Mailbox Database.edb, as database files use .EDB extensions). For each new database you add, you increase the number of .EDB files within a storage group. You could also create additional storage groups with additional databases.

Transaction logs help keep each database up-to-date. When data comes into the Exchange server, typically as an e-mail message, it enters the system memory.
From memory, it’s written to a transaction log. Each log reaches a maximum size of 1MB (a reduction from 5MB in Exchange 2003). These transaction logs are eventually added to the database that stores the mailbox for the intended recipient.

There’s a check file that keeps track of which transaction logs have been updated into the database. The benefit here is that you have redundancy, although it acts as protection for you only if you go through the effort of separating the disk location of your logs and database.

This allows for better performance and proper disaster recovery. In the event that the database is corrupted or the disk carrying the database crashes, those transaction logs are invaluable. You can combine them with the latest backup to restore your system. Understanding how the transaction logs and the database work together is essential to understanding these high-availability solutions for Exchange 2007.

**High Stakes, High Availability**

As mentioned earlier, there are three primary high-availability options beyond your ability to structure your database and transaction logs for improved performance and availability. Placing your database on a Raid 5 disk structure and mirroring your transaction logs is a recommended practice, but even that won’t prevent database corruption.

Understanding how the transaction logs and the database work together is essential to understanding these high-availability solutions for Exchange 2007.
the GUI with the Exchange Management Console or even through PowerShell commands from within your Exchange Management Shell. In the event that one disk crashes or the database is corrupted, you can switch over to the secondary copy of the data by typing in a manual switch. Keep in mind that this is an inexpensive solution that you can do from Standard Windows Server 2003.

**The benefit with CCR is that you eliminate single points of failure because there are two unique systems with two sets of storage. This offers a higher level of availability than an LCR set.**

To fully understand the way CCR works, visualize the two servers. The active server has a network connection to the public network. The passive server does, as well. Between the two nodes, however, is a private network connection on a separate network-addressing scheme that carries the “heartbeat” signal between them.

The passive server waits patiently, as long as it receives a heartbeat from the active node saying “I’m alive.” It then responds back that it, too, is alive. You can configure the cycle of these heartbeats, but by default they’re sent every 1.2 seconds from each cluster node.

If the passive server doesn’t receive a heartbeat (which could happen for any number of reasons), it starts getting edgy and eager to become active. If, however, it did become active while the other server was also active, it could cause a problem known as split-brain syndrome. To prevent this problem, there’s a quorum (called a Majority Node Set, or MNS quorum) that maintains a share file witness between these two servers.

This is held on a third server (typically the Hub Transport server of the same Active Directory site as the passive and active nodes), and makes the final determination for the passive node if indeed the active is alive and

Figure 3. Single Copy Clusters also use an active and passive node, but share the same storage location.
In the event that the active server is actually down, the passive server will automatically come to life and assume the workload.

While asynchronous log shipping (also used in CCR) may involve some data loss, there’s another process that can prevent this loss when used in a CCR set. On the Hub Transport server, there’s a feature you can configure called the Transport Dumpster. This retains a predetermined amount of mail-message data before delivering it to the cluster.

If the active node goes dead and the passive node jumps in, one of the first orders of business is for the “new” active server to check in with all Hub Transport servers and request any mail data it may not have received. This new active server will double-check all incoming data. It will retain any new messages and discard duplicates. This ensures a greater degree of high availability than LCR.

**One-Stop Shopping**

Single Copy Clusters (SCCs) are similar in design to the high availability solution available in Exchange 2003. You have a two-node cluster that relies on a single-storage location (see Figure 3, previous page). This type of solution provides system redundancy, but requires that you provide your own storage redundancy (which could be a NAS or SAN with RAID-level redundancy).

In Exchange 2003, you could configure an active/active mode where both servers were active simultaneously. This solution was so problematic that instead of being updated and enhanced for Exchange 2007 it was discontinued. SCC works with the active/passive configuration. To evaluate this solution on cost, keep in mind that SCC requires two systems, a RAID-enabled storage solution and the Enterprise Edition of Windows Server 2003.

While LCR, CCR and SCC are the three primary options, the Exchange development team has announced it will release another solution with Exchange 2007 Service Pack 1 later this year.

“With Standby Continuous Replication [SCR], data can be replicated on a per-storage group basis to standby servers or clusters,” according to the Exchange development team. “The SCR target, whether a single mailbox server or a cluster, can be placed inside the primary data center or in a remote location, ready to be manually activated if the primary server or data center fails.” Stay tuned for more on this development.

**Which Way To Go?**

Making the right decision of which approach is best for your environment is a tough one. You need to weigh the cost of high availability against your needs. You may decide a third-party solution is worth the added cost for even higher availability. Whichever method you choose, rest assured that Exchange 2007 has been designed to make any high availability solution easy to execute.

Storage Strategies for a New Exchange
How storage needs have changed in Exchange Server 2007.

Planning storage for Exchange Server databases has always been something of an art form. Databases have to be arranged in a way that reduces the chances of a catastrophic failure, while maintaining an acceptable level of performance.

In smaller organizations this might simply mean keeping databases and transaction logs on separate disks. In larger organizations though, ensuring performance and resilience often means investing in a SAN.

Changes to Exchange Server Databases
In spite of early rumors that Exchange Server 2007 was going to do away with the JET database format and use SQL Server databases instead, Microsoft chose to continue using JET in Exchange Server 2007. Although Exchange 2007 retains the same basic database format as its predecessors, there have been some changes made to JET that will ultimately affect the way that you have to plan for storage.

Required Disk Space
There is one storage consideration that holds true for all Exchange Server deployments, simple or complex: The databases must reside on a separate disk from the transaction logs. Not only does keeping the databases and transaction logs isolated from each other improve performance, it also ensures that data that has not yet been backed up will not be lost if the drive containing the databases was to fail.

Once you have accepted the fact that transaction logs and databases need to be stored separately, then the next logical question is how much disk space you are going to need for each. Fortunately, Microsoft gives us some simple formulas for determining how much disk space will be needed.

Calculating Space for the Database Disk
Determining how much space you are going to need on the disk that stores your Exchange Database is more complicated than you might initially assume. It
seems logical enough that if you multiply the number of anticipated mailboxes by the disk quota that you plan on imposing on each mailbox, you will have derived the maximum database size and will therefore know how much disk space will be required. This method won't even give you close to a correct answer though.

Multiplying the number of anticipated mailboxes by the mailbox space quota does give us a good starting point though. However, you must also take into account overhead caused by things like receiving or deleting messages and the nightly maintenance process.

In order to be able to calculate how much disk space will be consumed by overhead, you need to have an idea of how much mail users send and receive each day on average. To see how this process works, let's pretend that you have 100 users and have set a 250MB mailbox quota for each user. Let's also assume that each user sends and receives a total of about 15MB of messages each day on average. Eventually we have to assume that each user will ultimately reach their storage quota. At that point, each user must delete enough old messages each day to make room for the new messages that are coming in.

To see how much space could potentially be consumed by the dumpster, let's go back to my earlier example in which an Exchange 2007 server contains 100 mailboxes and each mailbox has a 250MB storage quota. Since we said that users send and receive a combined total of 15MB worth of messages a day, let's pretend that the received messages account for about 10MB per user and that sent messages account for the other 5MB per user. With this in mind, you can determine the maximum amount of space that should theoretically be consumed by the dumpster by taking the daily total of received mail, multiplying it by the total number of mailboxes, and then multiplying your result by the 14-day retention period (10MB * 100 mailboxes * 14 days).

The amount of space consumed by the dumpster will vary for a while because users may not delete the same amount of messages each day. Eventually we have to assume that each user will ultimately reach their storage quota. At that point, the maximum amount of space that will be consumed by the dumpster though. In case you are unfamiliar with the dumpster it is used as a temporary repository for messages that have been deleted. By default, deleted messages will be retained in the dumpster for 14 days after deletion.

The amount of space consumed by the dumpster will vary for a while because users may not delete the same amount of messages each day. Eventually we have to assume that each user will ultimately reach their storage quota. At that point, the maximum amount of space that will be consumed by the dumpster though. In case you are unfamiliar with the dumpster it is used as a temporary repository for messages that have been deleted. By default, deleted messages will be retained in the dumpster for 14 days after deletion.

Database Dumpster
With this in mind, the maximum amount of space that can be consumed by messages is about 25GB (250MB multiplied by 100 mailboxes). We also need to account for the amount of white space that will be consumed by the dumpster though. In case you are unfamiliar with the dumpster it is used as a temporary repository for messages that have been deleted. By default, deleted messages will be retained in the dumpster for 14 days after deletion.

The result is that about 14GB of space could be consumed by the dumpster. Being that the total amount of space consumed by messages is only 25GB, the dumpster’s size could be up to 56 percent of the total size of the database. In the real world the overhead percentage will vary depending on the dumpster’s retention period, your inbound mail volume, and your disk quotas. It is also possible for the dumpster to temporarily exceed its estimated size if a user unexpectedly deletes a large number of messages.

White Space
Another factor that you must consider when planning the amount of disk space to dedicate to your databases is the amount of whitespace that exists within the database. White space is created when messages are deleted from the database. When a message is deleted, an empty database page is left behind. This empty page consumes disk space even though it doesn’t actually contain data. When the maintenance process runs at night, the database is defragmented and whitespace is grouped together, but not removed. The only way to get rid of white space is to perform an offline defragmentation (which is not usually recommended).

The amount of whitespace in a database is constantly changing, but you can estimate the maximum amount of space consumed by whitespace by multiplying the number of mailboxes on the server by the average amount of mail sent and received by each user. Therefore, if you had 100 mailboxes and each user sent and received a combined total of 15MB worth of messages each day on average, then about 1.5GB of disk space could be consumed by white space.
The fluff factor is just Microsoft speak for the unknown. The basic idea is that your database will sometimes swell beyond its normal size for reasons beyond your control.

**Indexing**
In an Exchange 2003 environment, I usually advised people not to use content indexing because it consumed a tremendous amount of system resources. In fact, the index itself was 35 to 45 percent of the total size of the database. In Exchange Server 2007 though, the index has been completely redesigned and only adds about 5 percent to the size of the database that is being indexed.

**The Fluff Factor**
One last thing that you need to account for when planning your database's size is something known as the fluff factor. The fluff factor is just Microsoft speak for the unknown. The basic idea is that your database will sometimes swell beyond its normal size for reasons beyond your control. For example, earlier I mentioned that the dumpster size could increase if a user unexpectedly deleted an excessive number of messages. In order to account for the fluff factor, Microsoft recommends adding 20 percent to your projected database size.

**Maintenance and Disaster Recovery**
So far I have discussed the primary factors that contribute to a database's physical size. The amount of disk space that you are going to need is going to be different from the database’s size though. There are some maintenance and disaster recovery processes that require a temporary copy of the database to be created. An example of such a process is the offline defragmentation that I mentioned earlier. To account for any possible maintenance or disaster recovery processes that need to be performed, you must ensure that the disk housing your database has room for two copies of the database, plus 10 percent for overhead.

**Remember Those Log Files**
Now that I’ve shown you how to calculate the space required to support Exchange Server databases, let’s take a look at the log files. Determining the space requirements for log files isn’t nearly as complicated as figuring out how much space will be needed by databases. All you really need to know is that every transaction is initially written to a log file, and that log files are purged when the database is backed up.

To see how this works, let’s go back to our earlier example in which there were a hundred mailboxes and each user sent and received a combined total of 15MB worth of messages on average each day. All we would have to do to figure out how much space would be consumed by log files is to multiply 15MB by 100 users.

**Calculating Database Size Requirements**
As you can see, there are a lot of factors that contribute to the amount of disk space that your databases are going to consume. We already calculated that only about 25GB would be consumed by user mailbox data, but let’s see how much disk space this 25GB of data will actually require:
As such, the log files would consume about 1.5GB of space. Before you settle on such a small amount of disk space though, there are two things that you need to keep in mind. First, log files are not purged if the backup fails. Therefore, I recommend dedicating enough disk space to accommodate a week’s worth of logs just in case you ever have backup problems. That would mean that your log file volume would need about 10.5GB of space.

The other thing to keep in mind is that any databases in a storage group share a common set of log files. Therefore, you will need to account for the combined total log file needs of each database in the storage group when planning log file capacity.

Is the Database Too Large?
The last thing that I want to talk about is database capacity. Even though the database’s capacity might technically be unlimited, there is a practical limit to databases sizes, after which performance begins to suffer. Microsoft recommends that databases not exceed 100GB in size unless some form of continuous replication is being used. If Local Continuous Replication or Cluster Continuous Replication is being used, then Microsoft recommends keeping Exchange Server databases under 200GB in size.

One thing that I want to clarify is that this recommendation applies to the database itself, not to the amount of disk space set aside for the database. When we were estimating disk space requirements with our earlier example, we estimated that just over 112GB of disk space would be required. This does not exceed the 100GB threshold though because the database itself is only 51.02GB in size. The remaining 61.2GB is just empty disk space that has been set aside for use in maintenance or disaster recovery situations.

If you did determine that your database was too large, then you would want to create another database and move some mailboxes to it. Although a single storage group can hold up to 50 databases (5 in the Standard Edition), you are usually better off using a separate storage group for each database. Of course you will also want to dedicate separate disks for each database and for each set of transaction logs.

Sharing a disk between multiple databases completely defeats the purpose of keeping the databases small. If your database is too large, though, and your server can’t accommodate any more disks, one large database will typically perform better than multiple small databases that share a common disk.

Room To Grow
As you can see, there are a number of factors to consider when planning your Exchange Server’s storage capacity. Even so, it is worth working through the calculations because running low on disk space can cause performance problems and could potentially cause difficulties in disaster recovery situations.

Brien Posey, MCSE, is a Microsoft MVP and co-founder and vice president of research of Relevant Technologies, which specializes in IT best practices and information security. Brien is also a prolific technical author, having contributed articles to Tech Target and other online publications, as well as written numerous white papers. www.brienposey.com
You can contact Brien about “Storage Strategies for a New Exchange” at editor@mcpmag.com.
The new Exchange offers welcome changes on the admin side, but not as many on the client side.

Among the major improvements to Microsoft Exchange Server 2007 are better security, more flexible deployment options and easier administration and management. Early users say they’re pleased with the changes but are facing hurdles with widespread deployment.

Besides the fact that Exchange 2007 is a native 64-bit application that often requires new hardware, it also demands a complete overhaul of any current Exchange 2003 infrastructure. Plus, the true benefits of Exchange become clear only when using it with Microsoft’s other new 2007 releases like Office 2007 and SharePoint 2007.

Brandon Haag, executive director of IT at Gainesville State College in Gainesville, Ga., says he plans to move the school over to Exchange 2007 sometime this summer as he moves to Microsoft’s other new suites. “The main incentive for moving to Exchange 2007 is the additional capabilities we’ll get from integrating with SharePoint Server and the Office 2007 suite,” he says. “Obviously, when you get everything updated and at the same level, they all play a very sweet tune.” Getting to that point will take time, though, as moving to Exchange 2007 is a major upgrade.

Role with the Punches

Exchange Server 2007 introduces an expanded concept of server roles. In Exchange 2003 server roles existed, but they were fairly primitive, allowing only for front-end and back-end roles.

In Exchange 2007, server roles are far more robust, which gives admins unprecedented flexibility and deployment options. For example, server roles can split the functions of an Exchange server and place each role, or a combination of roles, on different servers. This eases management and performance constraints.

Exchange 2007 introduces five new server roles: Edge Transport, Hub Transport, Client Access, Mailbox and Unified Messaging. This breakdown lets smaller organizations deploy several roles on one hardware server, while larger organizations can divvy up the roles among several servers, clustering and load balancing the application to ensure optimal performance.

For the most part, early users are intrigued with the new server roles. In fact, many are already using one or two Exchange 2007 server roles in tandem with their current Exchange 2003 infrastructure. This lets them take advantage of some new features while putting off the pain of a wholesale upgrade.

For example, Haag is using the Exchange 2007 Edge Transport server in front of his Exchange 2003 infrastructure to take advantage of 2007’s more robust anti-spam capabilities. According to Microsoft, the Edge Transport uses connection filtering, content filtering, recipient filtering, SenderID and sender and IP reputation to reduce the amount of spam de-
All of my incoming e-mail is going through the Edge server and being filtered before it reaches my Exchange 2003 servers,” Haag says. So far, it seems to be working.

The downside of server roles is that they can make it more costly to implement Exchange 2007. “It’s great that they split it up like that, but now it increases the cost to run it,” says Thommi Montoya, an independent consultant based in Tempe, Ariz. “You’re buying another four Windows licenses and another four Windows servers, all of which are 64-bit servers. It’s definitely more robust, but also a little more costly.” That cost may be offset by savings in ROI, he says, but that won’t become apparent until Exchange 2007 is deployed more widely.

Early users are also excited about the revamped Outlook Web Access (OWA) within Exchange 2007. OWA has been overhauled and is now far more feature-rich and robust. The only major complaint users have is that the new features are available only on Microsoft’s own Internet Explorer (IE) browser, and not competitive offerings like Firefox.

“That was pretty disappointing,” says Scott Nixon, network manager at Windsor School, a private secondary school in Boston. Nixon says the school’s Exchange 2003 system currently supports 600 mailboxes, 500 of which are OWA-only. “If you’re using Firefox and you go to create a new message, it’s just going to be plain text only. And there’s no spell check, which is a big deal to our users.” Although Firefox has added a spell check feature to the browser itself, Nixon would’ve liked to see more capabilities from Microsoft.

Others are more than pleased with 2007’s OWA. “I’ve had nary a complaint,” Haag says, noting that his school’s 16,000 student accounts all access Exchange via OWA, mostly with IE. “OWA 2007 has a long list of configurable options for the Web interface. It must be three or four times longer.”

Aaron Foint, Windows systems administrator at Worcester Polytechnic Institute (WPI) in Worcester, Mass., offers a similar assessment. “If you use Firefox with Exchange, you have to use the Lite version,” he says. “You can do your basics: read e-mail, process messages [and] view calendar items, but you don’t get any of the advanced features.” Foint says he just makes sure his OWA users know they need IE to access advanced features.

Users love the idea of adding the shell, but question some of its capabilities. “It seems like in certain aspects, they went backwards,” says WPI’s Foint. “The Exchange Management Shell is great, and there’s definitely a lot of stuff you couldn’t do before in bulk mode very easily without some type of scripting or a command shell. At the same time, now there are certain things you can only do from the command shell, like configure POP or IMAP, and that’s weird.”

Montoya agrees: “I like that you can do most commands from either the command-line shell or the console,” he says. “What I don’t like is that they took a bunch of stuff you used to be able to do in the console and moved it to the shell.”

Still, most like the new management console, noting that it’s easier and more intuitive to find things now. Additionally, console operations are mirrored in the shell, making it obvious what commands are being used to do what. The GUI no longer keeps everything under the covers.

“I thought it was easy to set up and I liked that as you created accounts and so on, you were actually seeing the PowerShell script that was doing the action,” Windsor’s Nixon says.

One major shortcoming is that users can no longer manage Active Directory accounts and Exchange accounts from the same management interface.
One major shortcoming is that users can no longer manage Active Directory accounts and Exchange accounts from the same management interface. Foint says that’s a major problem for him, and attributes it to the fact that the Exchange Management Console is a level above the current Microsoft Management Console (MMC).

Microsoft will probably come out with a new version of the MMC with Longhorn, so at that point, there will probably be one integrated console where you can manage both,” Foint says. Another complaint Foint has is that there’s no longer an easy way to export mailboxes to a .PST file. “We used to be able to dump mailboxes to a .PST file using Xmerge in 2003,” he says. Foint says he expects the support in Service Pack 1.

Safe and Secure
Exchange 2007 has much tighter security than 2003. “If you do the recommended deployment in Exchange 2007—with your perimeter servers—it’s a more secure and robust model than they’ve ever had,” Montoya says. “They put a lot more thought into the deployment model.” Still, he’s reserving judgment until more organizations roll out 2007.

A downside to the increased security is that it’s more complex and difficult to do certificate management for enabling SSL in Exchange 2007. “In Exchange 2003, whatever common name you wanted to use for people to access OWA or ActiveSync, you had to create a certificate for that name,” explains WPI’s Foint. “For Exchange 2007, you need to create a certificate with multiple common names—it has to be for the common name you’re going to use publicly, it has to be for the actual server and then also for every possible combination.”

All of which makes it harder and more costly to manage. “It’s doable but it’s more involved, and if you’re getting a certificate from an external certificate authority, it needs to be registered for all the names, which is more expensive,” Foint says.

Client Constraints?
Beyond the changes to OWA, the new features aren’t as compelling on the client side. Foint says his users are looking forward to the new scheduling capabilities and out-of-office message flexibility, but both features are available only to users running Outlook 2007, part of the new Office 2007 suite.

“Advanced meeting and scheduling assistant is good,” Foint says. “If you pick a couple of users and a room you want to have a meeting in, it will auto-suggest times when everyone’s available, but you have to be on Exchange 2007 and Outlook 2007. Not many people are in that situation yet.”

The new out-of-office message feature is another plus, but it also requires Outlook 2007. “You can choose not only if you’re in or out of the office, but during what timeframe, [which] message you want to go out for people inside your organization and [which] message you want to go out for people who are outside your organization,” he says.

According to Foint, even with the less-than-stellar client-side features, Exchange 2007 is a great tool and worth the upgrade. “There’s no must-have feature,” he says. “It’s a lot of little things and I definitely think it’s an improvement. It’s easier to manage, and it gives us new options.”

Others say they’re waiting to see what happens once a majority of organizations make the move to Exchange 2007. “I really want to see it in a big environment to see how it goes,” Montoya says, noting that he’s deployed it only in a test environment at home. “Right now, so far, so good.”

Joanne Cummings is a freelance technology journalist. You can contact Joanne about “Exchange 2007: So Far, So Good” at jocummings@comcast.net.