

realtimepublishers.comtm

The Definitive Guidetm To

Exchange Disaster Recovery & Availability

Paul Robichaux



Jim McBee, technical editor

Chapter 7: Building a High-Availability–Ready Infrastructure.....	132
What Exchange Doesn't Do.....	132
Technology Support.....	132
Choosing Appropriate Solutions.....	134
Infrastructure Design Issues.....	134
AD Design Issues.....	135
DNS Design Issues	137
Exchange Component Design Issues	139
Outlook Access	139
OWA and Exchange ActiveSync.....	140
Message Hygiene	140
Mobile Device Access	141
Solutions Blueprints.....	141
Solutions for Disaster Recovery	141
Scenario 1: Small Company, Small Budget.....	142
Scenario 2: Medium Company, Medium Budget	143
Scenario 3: Large Budget.....	145
Solutions for High Availability and Business Continuance	147
Scenario 1: Small Company, Small Budget.....	147
Scenario 2: Medium Company, Medium Budget	148
Scenario 3: Large Budget.....	150
Solutions for Validation, Testing, and Deployment	152
Summary	152

Copyright Statement

© 2005 Realtimedpublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimedpublishers.com, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimedpublishers.com, Inc or its web site sponsors. In no event shall Realtimedpublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimedpublishers.com and the Realtimedpublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimedpublishers.com, please contact us via e-mail at info@realtimedpublishers.com.

[**Editor’s Note:** This eBook was downloaded from Content Central. To download other eBooks on this topic, please visit <http://www.realtimepublishers.com/contentcentral/>.]

Chapter 7: Building a High-Availability–Ready Infrastructure

The preceding chapters of this book have described how high-availability and disaster recovery technologies work, and Chapter 6 described how to use disaster recovery tools and procedures. This chapter is the counterpart to Chapter 6; it explains what you need to know to build an infrastructure that’s ready to provide high-availability messaging services to your users. Some aspects of this infrastructure are easy and quick to implement; others may require you to carefully examine what is currently in your environment and make some tough decisions about whether the expected gain in capability is worth the additional cost and risk of changing the infrastructure.

Earlier, this guide talked about how hard it can be to pin down the exact definition of “high availability.” The definition that works for your organization may be quite different from what another company in the same business expects—much less what may be appropriate in another market sector. Rather than attempt to specify a given level of availability as high (or medium or low), it’s more appropriate to talk about *enhancing* your systems’ availability so that the infrastructure is better than what you’ve got right now. The degree of enhancement will depend on how much uptime you need, how much money you’re willing to spend, and specific features of your environment.


Any discussion of high-availability will involve counting 9s at some point. This chapter’s discussion will assume that your organization requires a maximum of 99.9 percent uptime; although it’s certainly possible to go beyond that level of uptime, doing so can be very expensive, and it’s rarely justifiable from a business standpoint. Many organizations that want $24 \times 7 \times 365$ uptime find that they actually need something closer to $6 \times 18 \times 365$, or even less, and the cost savings gained by finding the correct level of required uptime can be very significant.

What Exchange Doesn’t Do


This chapter is intended to describe how to build a high-availability–ready infrastructure, so it’s necessary to begin with a discussion of which high-availability features and technologies Exchange supports. This discussion includes all the key high-availability technologies talked about in earlier chapters: RAID, clustering, SANs, replication, and failover.

Technology Support

Exchange doesn’t care whether you put its databases, queues, or logs on a RAID volume. In fact, when you install Exchange on a new server, the mailbox databases and transaction logs are located on the system boot volume by default; you have to move them yourself if that’s not where you want to put them.


 For more information about RAID, see Chapter 3.

Clustering is a different story. As described in earlier chapters, the Enterprise Edition of Exchange Server 2003 (and Exchange 2000) supports the Microsoft cluster service (MSCS) clustering functionality built-in to Windows. Because Exchange is a cluster-aware application, it takes full advantage of the MSCS interfaces: you can define additional instances of some types of Exchange resources (such as SMTP virtual servers) and then move them between nodes in a cluster at any time. Exchange supports as many as 8 nodes in a cluster, although you must have at least one passive node in any clustered Exchange configuration (in other words, you can't build a two-node active/active cluster). MSCS offers a good failover model; you can fail over resources automatically upon failure or manually. Most of the Exchange components are cluster-enabled, although some (like the MTA) can only have a single instance per cluster. Many third-party products built on Exchange are cluster-aware, or at least offer cluster-aware versions.

 VERITAS Software (now part of Symantec) sells a clustering product that works with Exchange. However, it's not supported by Microsoft, and I don't recommend that you use it as part of your clustering design unless you have enough in-house expertise to properly design and implement a non-Microsoft clustering solution.

Exchange's support for SANs (whether they're FC or iSCSI) is somewhat akin to its support for RAID—Exchange doesn't know or care whether your databases are stored on a SAN volume or a directly attached storage device. However, Microsoft explicitly supports the use of SANs with Exchange, meaning that the company provides product support for SAN-based configurations—provided that the SAN appears on the Windows Hardware Qualification List (WHQL). With appropriate attention to the amount of write latency imposed by the SAN, you can also use geographically dispersed clusters with Exchange (although, as described earlier in this book, these are expensive and finicky clusters that require a great deal of specialized knowledge to set up and run). However, Exchange doesn't do any kind of SAN management or integration; no matter what kind of SAN you use, regardless of where it's located or how it's replicated, Exchange treats the SAN volumes just like any other disks. Brocade, Symantec/VERITAS, and other companies offer SAN management software to help you get better utilization and performance out of your SAN resources.

Any discussion of replication support in Exchange will necessarily be short: Exchange supports replication of public folders, and that's all. It's worth mentioning that Active Directory (AD—and AD-integrated DNS) supports multi-master replication, thus providing an efficient means of replicating the directory data upon which Exchange depends. Third-party solutions offer replication of individual mailboxes, Exchange databases, or entire Exchange volumes, but none of these services are provided by Exchange.

 As of this writing, Microsoft has announced that it will provide continuous data protection support in the next version of Exchange, code-named Exchange 12. However, the company hasn't publicly disclosed any details, so it's difficult to judge what value these features will add.

Exchange provides excellent failover support through MSCS: clustered resources can automatically or manually be failed over to any node in the cluster. Bear in mind that each node in a cluster can support a single Exchange virtual server (EVS) at once. This support is important for clusters with more than two nodes. Say that you have a three-node cluster, made up of active nodes A and B and passive node C. If A fails, its EVS can move to C, leaving B and C as active nodes. If, however, either B or C fails, their workload can't be moved to the surviving node because a single node can't host two EVSs.

However, Exchange's failover support is tied only to MSCS. If you're not using MSCS, Exchange doesn't provide intrinsic support for moving mailboxes between servers. If you restore mailboxes to an alternative server, you (or your failover solution) will have to update the user account objects that own the relocated mailboxes so that they point to the correct mailbox location. You can get additional availability for services such as OWA and SMTP by using Windows' native network load balancing (NLB) features; although NLB isn't the same as failover, it does provide redundancy of service access for services it supports.

Choosing Appropriate Solutions

Fred Brooks, a famous software architect best known for his book *The Mythical Man Month*, wrote a seminal essay in 1986 titled "No Silver Bullet: Essence and Accidents of Software Engineering." This book is well worth reading if you're interested in how software is developed; Brooks' major point is that there is no single solution that fixes the problems and difficulties inherent in doing large-scale software development. It turns out that the same thing is true for Exchange. There is no single product or technology that you can apply to your infrastructure to suddenly improve its availability. However, there are many individual technologies and techniques that you can apply, as described earlier in the book. This section will examine the range of infrastructure services for which you need to provide high availability and will discuss methods to address their individual requirements.

Infrastructure Design Issues

Although the majority of this guide has been dedicated to discussions of Exchange high availability and disaster recovery, it's important to realize that Exchange can't do much of anything without appropriate DNS and AD resources. Accordingly, now is a good time to talk about solution design for those resources, as an effective Exchange design will necessarily include redundancy and resiliency for these prerequisite services.

AD Design Issues

AD design is deserving of an entire book of its own; thus, this section will cover only the most important points. First, remember Microsoft's recommendations: The company specifies that there should be at least one Global Catalog (GC) server in every AD domain or site that contains Exchange servers. This recommendation holds for several reasons:

- Without access to a GC, Exchange's functionality will be severely limited. Mail delivery to internal users, mail routing to external users, and directory and GAL lookups will all be impacted. If Exchange can reach a domain controller, some of the queries it needs to make can still be satisfied (particularly if you have only a single domain), but in multiple-domain environments, Exchange will require access to a GC so that it can retrieve attributes for objects that are in other domains. In addition, Exchange requires access to a GC to expand the membership list of mail-enabled distribution groups. You can finesse this requirement by defining a default expansion server for domain controllers, but that shifts load over to the expansion server, which may not be suitable for your environment.
- If the Outlook client can't find a GC, it won't be able to display a global address list (GAL) to the user. This problem may be mitigated somewhat if your Outlook clients have access to a current offline address book (OAB).
- Exchange will attempt to locate a GC that's "close" in network terms by giving preference to GCs that are in the same AD site. That provides good performance for queries; if a same-site GC isn't available, the DSAccess component of Exchange uses a variety of tricks to locate a suitable GC. The DSAccess detection algorithm is smart enough to take AD site links and site link bridges into account when calculating which GC it should use, but in order for that to be effective, you need to have previously designed (and tested!) a topology that provides low-latency access to GCs for all your Exchange servers. Figure 7.1 shows MExchangeDSAccess event ID 2081, which you'll see logged when the topology changes; it indicates which domain controllers and GCs your servers are using.

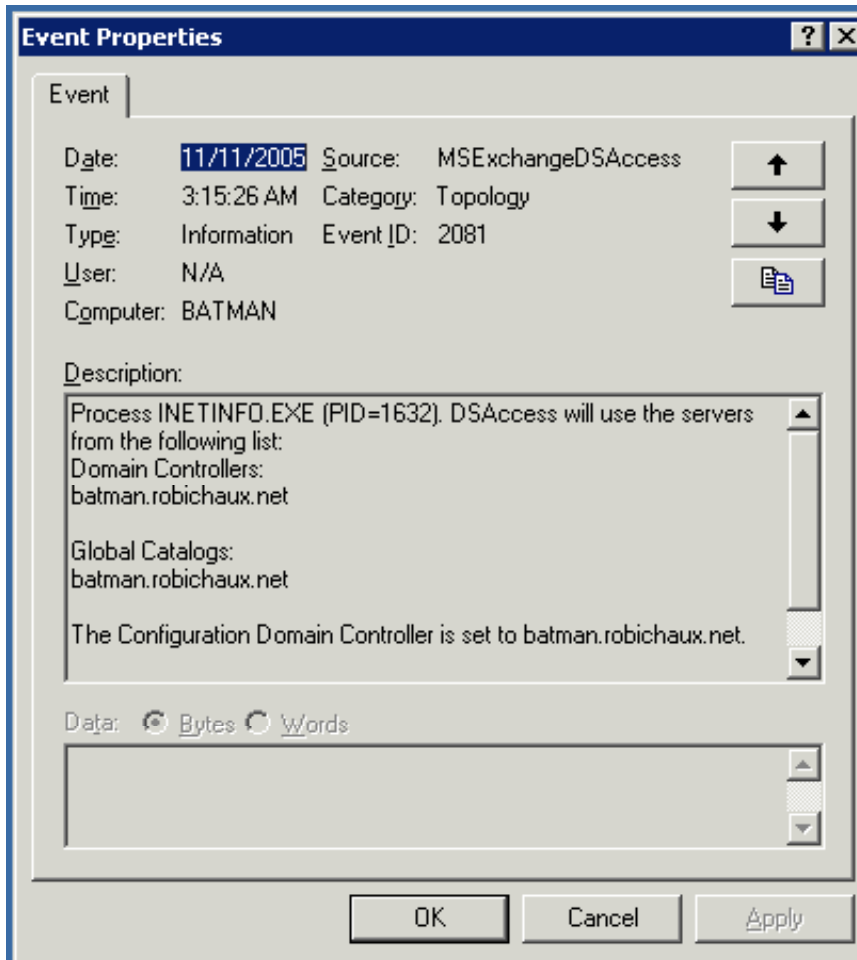


Figure 7.1: *MExchangeDSAccess* event ID 2081 tells you exactly to which domain controllers your Exchange server is talking.

If one GC per domain is good, you might think that having more GCs would be better—and you would be right! If possible, putting a GC in the same site or domain that your Outlook users are in will improve lookup performance. In addition, putting more than one GC in each site or domain that has Exchange servers will provide redundancy for *all* directory services, not just for Exchange.

It turns out that Microsoft changed the referral behavior for GC location in Exchange Server 2003 Service Pack 2 (SP2). There is a full explanation of the changes on the Exchange team blog at <http://blogs.technet.com/exchange/archive/2005/11/04/413669.aspx>. Briefly, though, the new selection algorithm calculates a score for each GC that it can detect, awarding points to each GC based on whether it's available, which domain it's in, which AD site it's in, and whether it supports the same protocol set as the client. This method makes the detection process much more transparent, as you can always calculate the score using the same mechanism that Exchange itself uses. Bear in mind that this behavior change applies only to the process used to select the GC by Exchange; it has nothing to do with the process by which Outlook finds a GC for its own use. However, Exchange can provide a GC referral to Outlook; Microsoft article 319206 “How to Configure Outlook to a Specific Global Catalog Server or to the Closest Global Catalog Server” describes how to override this referral behavior if you want to force Outlook to use a specified set of GCs.

Of course, the number of GCs and their network location isn't the only important factor to consider in your AD infrastructure design. You also need to consider their processing power. Shortly after Exchange 2000 was released, Microsoft released a recommendation that specified that you should have one GC for every four Exchange servers. This recommendation confused a lot of people, so Microsoft clarified it by saying that you should have one GC *CPU* for every four Exchange CPUs. That explanation didn't clarify the company's meaning, but knowing that Microsoft was referring to CPUs of equivalent power is a key part of the explanation. For example, if you have two Exchange servers, each with a pair of 2GHz Opteron processors, your GC should have at least one 2GHz Opteron. The number of physical servers isn't important, but the total CPU capacity is. (No word yet from Microsoft about how or whether these recommendations change if you're using Virtual Server 2005 R2 to virtualize either your Exchange servers or your AD servers.) Of course, the number of physical servers *is* important if you have only one of them, as a failure will leave you stuck.

In November 2005, Microsoft announced that the next version of Exchange (code-named Exchange 12) will run only on 64-bit hardware running the 64-bit edition of WS2K3. This development has some interesting implications for Exchange scalability. It has similar implications for AD GC scalability. For example, while testing AD scalability, Hewlett-Packard found that a single dual-Opteron x64 server with 14GB of RAM could handle the same number of directory queries as 11 32-bit servers—a total of more than 20,000 Exchange mailboxes worth of load. This statistic is a compelling argument in favor of moving to 64-bit hardware; as part of your infrastructure design you should consider doing so. However, don't forget that having only a single server leaves you vulnerable to single-point failures in this critical portion of your infrastructure.

DNS Design Issues

The guidelines for DNS server provisioning aren't quite as firm as the ones for AD provisioning. Obviously, if you have only one DNS server, it's a single point of failure, so you would want to make sure to have at least two accessible to each Exchange server. DNS is required for several circumstances. First, Win2K and later use DNS to locate domain controllers and GC servers. They look for SRV records registered in the AD domain for the machine's IP address; if the DNS server isn't reachable, the client won't be able to log on. However, because Windows supports caching of logon credentials, this inaccessibility isn't the end of the world (unless, of course, you've disabled logon caching).

Exchange uses DNS in a variety of ways:

- To locate domain controllers for logon requests. Because an Exchange server can be in any domain in a forest, users who try to log on to the Exchange server may have accounts homed in a different domain.
- To locate Sender ID/SPF records when Sender ID resolution (a new feature added in Exchange Server 2003 SP2) for inbound mail is enabled. The SPF record indicates which IP addresses are authorized to send mail on behalf of a given domain; depending on how you have Sender ID configured on your server, if you can't get the Sender ID record for a domain, Exchange may reject mail from outside domains. Therefore, Microsoft sets the default Sender ID configuration to mark inbound messages with Sender ID information but not to reject messages based on that information. The Sender ID rating is passed to the IMF for use in its determination of the spam confidence level (SCL) for a given message.
- Your Exchange server will perform MX record lookups while sending outbound mail; these lookups enable Exchange to find the target servers for individual domains to which your users have addressed mail. If the server can't perform those lookups, mail will remain queued on the server until such time as the lookups succeed. Note that this won't be true if you've configured your server to route outbound mail through a smart host.
- For inbound mail, mail servers in the outside world perform DNS lookups to find the MX record for your domains, which tells them which of your servers accept inbound mail. If the MX record isn't available, mail to your domain will be undeliverable until the problem is fixed.
- If you've turned on reverse DNS resolution on your inbound SMTP virtual server, Exchange will make DNS queries to attempt to resolve sender addresses.

What you may have noticed from this list is that there are two separate sets of dependencies: your Exchange server depends on DNS to send mail out to the Internet, and the outside world depends on your DNS to send mail to you. (In fact, with Sender ID or reverse DNS resolution turned on, your inbound mail flow will depend on DNS services as well!) Thus, you need to have two separate sets of concerns in mind. First, you'll need to maintain a sufficient number of your own DNS servers to provide redundancy for internal queries. There is no rule for deciding how many servers you should have; if you only have one, it represents a single point of failure. For that reason, you should plan on having more than one server available to your Exchange servers. Most sites use DNS servers collocated with domain controllers; AD-integrated DNS offers the simplest and most straightforward approach. You can also provision dedicated DNS servers for one or more subnets if you need to additional capability.



I recently worked with a client that had an unexpected outage when one segment of its network stopped passing DNS traffic to the DNS server—servers on other network segments were unaffected, and there was nothing wrong with the DNS server itself. In all your consideration of infrastructure design, don't forget to consider the lowly network, because without it you won't be getting much done! Many of the Exchange outages I see in production networks come about because the network management folks change firewall rules, network routings, IP addresses, or accessible ports without warning. For example, imagine that some well-meaning but ignorant firewall administrator blocked TCP 135 between your Outlook clients and Exchange servers—instant pandemonium without any warning signs from the Exchange server!

For the other half, your options may be more limited. If you've configured Exchange for direct delivery, it will need access to DNS records to deliver messages. You probably don't have a copy of the entire DNS for the world, so you will need to set up forwarding queries to an outside server—and deal with interruptions or failures of that service. The reverse is also true; for people to be able to send you messages, they have to be able to look up your servers' addresses. Most small and midsized companies have Internet service providers (ISPs) maintain DNS service because by doing so they gain insulation from local DNS failures: a failure of their server, or even of their Internet connection, won't prevent outside servers from finding the correct place to send mail.

Of course, because Windows supports the use of multiple DNS servers, there is no reason not to configure each of your Exchange servers with the addresses of *all* the DNS servers in your organization. By doing so, you gain a higher degree of resiliency because the Windows DNS client will work its way down the list of available DNS servers, repeating a given query until it gets an answer or runs out of servers. The one case where this can cause problems is when you move, or remove, one of those servers and then don't update the list on your servers.

For that reason (among others), it might be a good idea to look into using DHCP to assign your Exchange servers addresses—just create a DHCP reservation so that the server always gets the same IP address, then allow routing and DNS information to be provided by the DHCP server.

Exchange Component Design Issues

Besides the components already described, you probably have other components that are important to your messaging environment. These components require a high degree of attention to ensure that they're still available when you need them. Later, this chapter discusses specific measures that you should take to ensure their availability, but it's important to mention their unique requirements up front so that you understand what they are.


Outlook Access

The Exchange and Outlook product teams expend a mind-boggling amount of effort to make their products work well together—this effort despite the fact that each product works with other technologies (such as the Outlook Connector for Lotus Notes and Microsoft's Entourage for Mac OS X). Because of the high effort put into this partnership, there are relatively few individual issues that you have to worry about to ensure that Outlook can talk to the Exchange server; those issues that do exist are mostly focused on the client side. In particular, you must ensure that the client can correctly resolve the DNS and NetBIOS names for the Exchange servers and GCs that it needs to talk to (bearing in mind that you can force Outlook to use a particular GC or set of GCs by adjusting the registry on the individual client). In addition, you should be monitoring the health of the DSAccess and DSProxy services on the Exchange servers themselves, because Outlook requires them.

One situation I often run into is basic network connectivity. If you're using RPC over HTTPS with Outlook 2003 and Exchange 2003, network connectivity is less of an issue; if, however, you've enabled RPC access to your Exchange servers through Microsoft's Internet Security and Acceleration (ISA) server or another similar solution, you may find that individual ISPs block the RPC ports (notably TCP port 135). This blockage is a constant problem for people who travel because hotels, convention centers, and other public facilities often have these ports blocked without notifying anyone. This leads to the unfortunate situation in which Outlook works fine in one place but not at all in another, depending on which network provider is in use.

OWA and Exchange ActiveSync

One important change in Exchange 2000 was the introduction of front-end servers; perhaps a better name for these servers would be "client access servers" (which is what they're called in Exchange 12). If you use front-end servers in your topology, you've probably found that one of the best features of this arrangement is that the front-end servers are all interchangeable. If one fails, any other front end can take over its load because all they do is proxy connections from Internet protocol clients to the appropriate back-end server. Although front-end servers can be used to proxy IMAP and POP traffic, most sites only use it with OWA. Exchange ActiveSync (EAS) essentially uses the same system, which is why I lump it in with OWA. In either case, the primary issue with these components is making sure that they're reachable by the outside world. EAS and OWA both use TCP port 443, so access is fairly easy to accomplish.

 You can run OWA over TCP port 80, using plain, unprotected HTTP—but you shouldn't! Although you might get away with this setup on your internal network, it's not a good idea because it essentially leaves all your traffic open to eavesdropping—and you should never deploy OWA without SSL on the Internet.

Message Hygiene

Message hygiene is increasingly important. Five or six years ago, relatively few companies had server-based antivirus systems and very few had anti-spam filtering. Now, of course, these two technologies are found in the overwhelming majority of Exchange installations—a sad commentary on how the messaging landscape has changed.

Some organizations get so much spam and virus mail that their operations would essentially be crippled if their message hygiene systems were unavailable; others would merely be inconvenienced. However, all inbound mail typically flows through these systems—so if they fail, inbound mail flow may be completely stopped. This is usually more problematic than a simple filtering failure would be. Depending on the filtering and hygiene solution you use, you may have more or fewer availability options. For example, if you use a hosted filtering service such as MessageLabs or Symantec BrightMail, it's a safe bet that the service provider has a more robust infrastructure than you do, so you don't have to worry about it. In contrast, if you use an appliance or software solution, you should be prepared to handle equipment or configuration failures of your filtering solution. In this situation, the Intelligent Message Filter (IMF) can come in handy for two simple reasons. First, it's installed by default in Exchange Server 2003 SP2. Second, because it does a very good job of first-line spam filtering, it's becoming the primary solution for many organizations; even if you're not using it, it makes a good fallback solution for times when your primary filtering system isn't available.

Mobile Device Access

Mobile device access is becoming increasingly critical at some companies, both because it's convenient for individual end users and because providing pervasive mobile access can greatly improve user productivity. However, this leads to a problem: as mobile access becomes more important, interruptions in that access have a correspondingly greater impact. For workers who don't have a conventional office, not having access to Exchange through EAS, a BlackBerry, or other device can be a serious problem. Fortunately, the devices themselves usually have device-side caching, so service interruptions don't necessarily prevent you from reading your existing mail. However, if the network connection between your servers and the service provider's network operations center are disrupted, you'll probably find that you can't send or receive *new* mail. As with hosted message hygiene services, availability of the service provider's network isn't always a concern.

Solutions Blueprints

It's time to take the theoretical discussions we've been having throughout the preceding six chapters and turn them into some practical blueprints for disaster recovery, availability, and continuance. In each category of blueprint, there are three scenarios included:

- A small company with a small budget. In these scenarios, "small" means fewer than 250 mailboxes, no matter how they're divided between offices and servers. In the majority of cases, these organizations have a single Exchange server in a single location; where the proposed solution doesn't work for those cases, I've noted it appropriately.
- A medium company with a medium budget. Here, "medium" means between 250 and 1000 mailboxes, spread across one or more servers.
- Companies with lots of money, regardless of the organization size. I've worked with some companies whose availability or disaster recovery requirements are so stringent that they're willing to spend money out of proportion to their size or the number of mailboxes they host. Examples include law firms (where billable time is king), financial services companies, and emergency services providers.

In each of these cases, I've provided a blueprint that diagrams what the solution looks like, plus some explanation of what particular issues the design is intended to resolve.

Solutions for Disaster Recovery

In planning and designing disaster recovery solutions, it's fair to assume that every organization needs a certain basic level of disaster recovery capability, often implemented with simple backup solutions. This will be the starting point; from here, it's possible to add more capability where it makes sense to do so.

Scenario 1: Small Company, Small Budget

Figure 7.2 shows the basic design for a small company with a limited IT budget; for this example, let's assume that they're a 100-employee manufacturing company. This configuration isn't the smallest or simplest possible configuration; for example, a single server running Microsoft's Small Business Server (SBS) product could consolidate the domain controller, Exchange, and firewall roles into a single computer (with a concomitantly higher risk of failure). However, for this blueprint, these roles are split into separate computers.

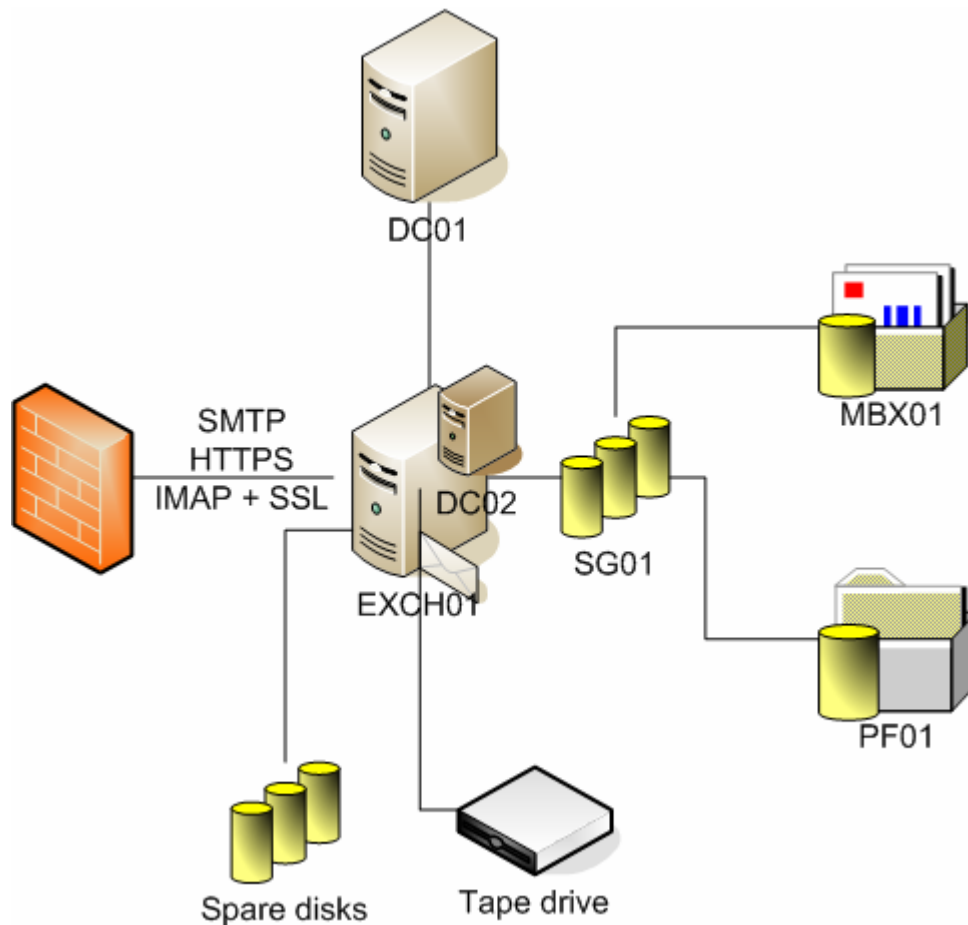


Figure 7.2: A basic design for a small company

This design features a single Exchange server with a single storage group; the storage group contains one mailbox database and one public folder database. The Exchange server has two backup mechanisms: an attached tape drive and enough disk space to allow disk-to-disk backups using `ntbackup`. The backup schedule is simple: daily full backups to disk at 6am, followed by dumping the day's on-disk backup to tape at 6pm. This setup provides good coverage through the day without undue performance impact and with no additional cost for backup software.

Although not shown in the diagram, the storage group's log files and the database files are on the same logical volume, a RAID-5 set. Although it would be preferable to have separate RAID sets for the transaction logs and the databases, keeping them on the same RAID-5 set provides adequate protection at a reduced cost—though on a busy server it would be better to separate the logs and databases to provide more consistent throughput.

Of course, this option is a budget solution; as such, it doesn't provide some of the capabilities discussed in earlier chapters. The biggest omission is in the area of rapid recovery; for most small companies, an 8- to 10-hour RTO is a reasonable balance between recovery time and expense. Without the means to provide for an alternative site or additional servers, it's difficult to improve on the capability of disk-to-disk backups. Of course, a hardware failure of either of the servers may break the recovery time budget; thus, DC02, which is actually a virtual machine, is shown "piggybacked" on EXCH01. Firing up that domain controller each night, then shutting it down again (which is easy to implement via script), allows adequate time for AD changes to replicate and ensures that a failure of the domain controller won't knock out email users.

Scenario 2: Medium Company, Medium Budget

For the medium scenario, let's consider a 600-employee construction management company with primary offices just outside Toledo; they have an office in Detroit that handles customers in and around southern Michigan. About half of the employees work primarily in the field, while the other half work in one of the two major offices. Field users typically don't use mail much, but there is a core of sales, marketing, engineering, and planning staff that are heavily dependent on email. For this reason, the users are segregated into two groups, each of which has its own storage group (and thus its own set of log files). Figure 7.3 shows one design that would suit this organization.

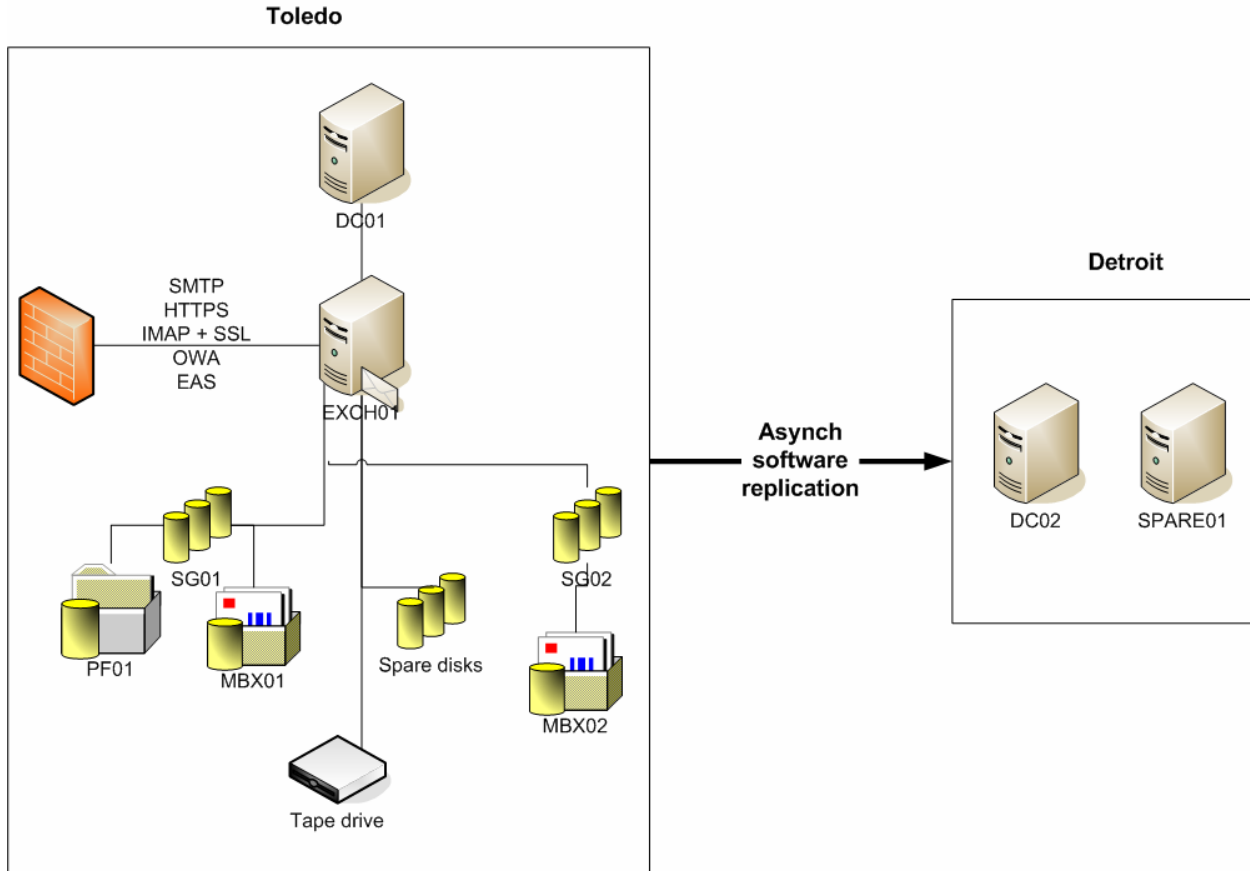


Figure 7.3: The medium company design includes more redundancy.

As with the first scenario, backups are performed first to disk, then to tape. However, putting the two groups of users in separate storage groups allows for more flexibility in backing up and restoring user data when needed and provides a way to do “surge” backups of both storage groups in parallel (one to disk, one to tape) in case of an unexpected situation. Because cost is an issue, as with the first scenario, a RAID-5 array is used to store all three databases. However, the transaction log sets are each stored on a separate mirrored pair; although this setup adds to the cost, the improved resiliency and performance is worth it.

Notice that Figure 7.3 doesn't show anything that looks like a SAN. Many midsized companies lack both the budget and the in-house skill set required to design, implement, and maintain a SAN. The disadvantage of not using a SAN, of course, is the lower scalability of direct-attached storage. For this imaginary company, this shortcoming is not a problem because the company expects its growth rate to be relatively low. For organizations that expect more growth (in either mail data or mailbox count), a good idea is to evaluate iSCSI SAN solutions because of their lower acquisition and deployment costs.

The Detroit site has its own domain controller so that workers in the Detroit office can continue to log on and work even if the Toledo site, or the link between sites, fails. Because only 25 percent of the total user population is in Detroit, it isn't necessary to put a separate production Exchange server there, although you certainly could. For most applications like this, where one group of users is less dependant on email than another, it's acceptable to use OWA or Outlook 2003 in cached Exchange mode to provide access at a remote site.

The biggest change between this design and the one shown in Figure 7.2 is the addition of asynchronous software replication. This addition provides a low-cost, low-impact way to copy Exchange data from the primary site in Toledo to the backup site. In this configuration, it's possible to set separate replication schedules for the two storage groups because they're on separate volumes—an SLA-friendly way to ensure that needed data is available without saturating the link between the two sites.

It's certainly possible to improve on this model for organizations that need higher availability. There is already a spare server (SPARE01) in Detroit that can be pressed into service as a domain controller or Exchange server; with appropriate replication and failover software, it could be used as a warm-standby Exchange server. Although this diagram doesn't show it, it would also be possible to allow inbound SMTP mail to flow to an Exchange server in Detroit by specifying that server as a mail exchanger using a secondary MX record; this setup would provide additional redundancy for mail delivery.

Scenario 3: Large Budget

For a reasonable example of a high-budget disaster recovery customer, consider the example of a law firm that has offices in Houston, Dallas, Austin, and San Francisco, with a total of about 2400 mailboxes (1200 in Houston, 500 in Dallas, and the rest roughly evenly distributed between the other two cities). Figure 7.4 shows the configuration for the Houston office.

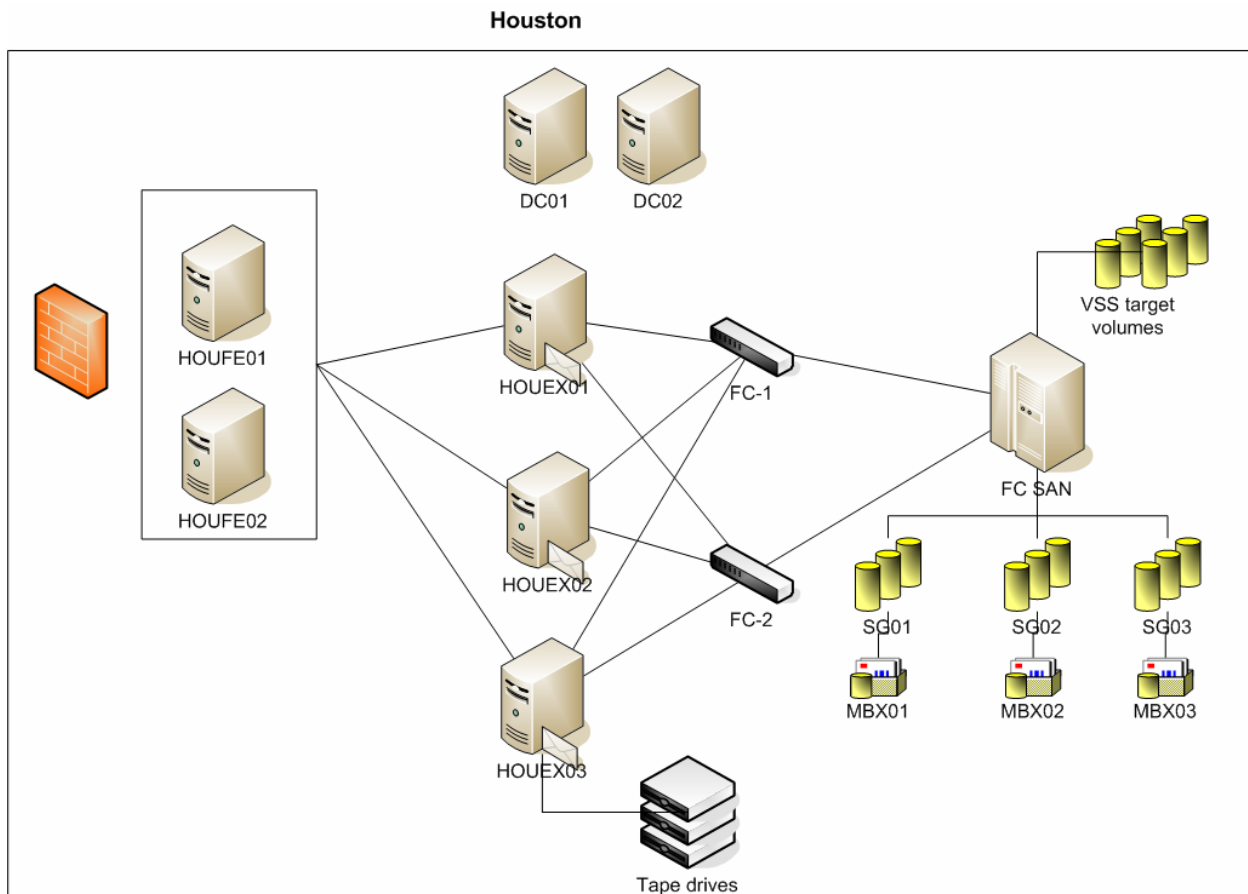



Figure 7.4: When money is no object, you can build a robust high-availability design.

This client is using a three-node (2 active, 1 passive) cluster in Houston and a two-node (1+1) cluster in Dallas. This setup would probably be overkill for most organizations of similar size; because the attorneys and support staff at this firm are heavy email users, the firm decided to spend the extra money to gain a higher degree of potential availability—but then they backed up that choice by making sure their people were adequately trained and qualified to manage a clustered Exchange environment. They also invested in a broad-scale management tool deployment to give them good visibility into what the SAN, cluster nodes, and infrastructure servers are doing.

Law firms generate prodigious numbers of documents, and virtually every law firm has some kind of document management system (often paired with a litigation management and tracking system). Large FC SANs are perfect for storing the large volumes of data that these firms generate; as a side bonus, it's possible to carve out a generous amount of storage space for Exchange. By their nature, SANs allow quick reallocation of physical drive space to various logical volumes, making it easy to dynamically re-provision space as needed. Of course, from a disaster recovery perspective, the best argument in favor of using SANs is that they provide a high degree of resiliency to begin with; in this case, that resiliency is supplemented with the use of the Volume Shadow Copy Service (VSS) as a primary backup mechanism. This requires that some disks on the SAN be reserved for VSS shadow copies, but those copies can be mounted on HOUEX03 and backed up to tape, or, more precisely, to *tapes*. By deploying three tape drives, the firm's administrators can back up or restore all three storage groups in parallel. They can also stripe a single storage group backup across multiple tapes, which greatly speeds recovery times. By carefully juggling their backup schedule to include both striped and normal backups as well as judiciously keeping more than one generation of VSS backups, they can recover to almost any point in time—as long as nothing happens to the Houston data center or its backups! (That contingency is addressed in their high-availability deployment, which is discussed in a moment.)

There are a few subtle details here. Each cluster node has a pair of FC HBAs, each of which is connected to a separate FC switch (for example, HOUEX01 is connected to both FC-1 and FC-2, which in turn have redundant paths to the SAN enclosure). In addition, instead of allowing direct access to the mailbox server, all inbound traffic is routed through a pair of front-end servers that are load-balanced with the Windows Network Load Balancing Service (NLBS). The front-end servers run the Intelligent Message Filter (IMF) and a gateway virus scanner, providing good protection without adding a performance or stability burden to the all-important mailbox servers.

 There are several details that aren't shown in this diagram for reasons of space and clarity. For example, this firm makes heavy use of BlackBerry devices, so they have several BlackBerry Enterprise Server (BES) servers.

Solutions for High Availability and Business Continuance

The solution blueprints shown earlier for disaster recovery share some features with the blueprints for high availability and business continuity, largely because some of the same technologies provide both types of protection. This section focuses on the specific differences inherent in high availability and business continuity design.

Scenario 1: Small Company, Small Budget

The sad, simple fact is that most small companies don't do much to provide high availability or business continuity for their messaging systems. There are several possible reasons for why this is so; it may have to do with the perceived cost of those technologies, the perceived lack of need for them, a lack of technical understanding, or some mixture of the three. Having said that, even if you have very little money, there are some simple steps you can take to help increase your high availability and business continuance capability.

First, I assume that you have more than one computer. Note that I didn't say "more than one server." For \$99, you can buy the standard edition of Microsoft's Virtual Server product and use it to build a replacement Exchange server. This server isn't intended to be used for production; instead, it's designed to let you bring up a database using a recovery storage group, then move mail to a replacement database. This method is a somewhat unconventional approach, and one that's not necessarily supported by Microsoft. It is, however, inexpensive and flexible, and it's surely better than nothing at all (provided, of course, that the server on which you host the VM has enough disk space and RAM to run both Exchange and Virtual Server).

Second, you should buy the best high availability hardware that you can afford. This acquisition may be as simple as adding an uninterruptible power supply to your existing server or as elaborate as buying new servers with more advanced high-availability capabilities. Regardless, good hardware will provide a significant increase in robustness and resiliency, and this increase is difficult to match with software alone, especially if you're on a budget.

What about business continuance? True continuance is often outside the budgetary or technical reach of small companies, but you may be able to simulate it in a variety of ways. For example, if you have a server at home, could it be pressed into emergency service as an Exchange server? Probably, assuming that you have a surviving domain controller so that you can join your home server to the domain and install Exchange on it. An even less expensive alternative is to keep records (on paper, and not in your office) of alternative email addresses that you can use to communicate with your staff. If your server fails, you can switch over to using Hotmail as a temporary measure.

Scenario 2: Medium Company, Medium Budget

With increased size and budget limits come increased high availability and business continuance capability. The same rule that applies to small organizations applies mid-sized companies: buy the best hardware you can afford to beef up the resiliency of the underlying systems. Beyond that, though, you can add redundancy to your infrastructure by adding servers: more GCs, more Exchange servers, and so on. These additions have the unfortunate effect of increasing complexity, too, which means you must carefully consider the degree of complexity you're willing to tolerate with the degree of redundancy that is appropriate for your business needs.

Beyond simply adding more of what you already have, you can supplement the disaster recovery solutions you have in place with more advanced high availability and business continuity tools. For example, look back at Figure 7.3, which shows a disaster recovery design for a mid-sized construction company. That is a solid basis for a design that provides effective disaster recovery *and* high availability, with a few changes (see Figure 7.5).

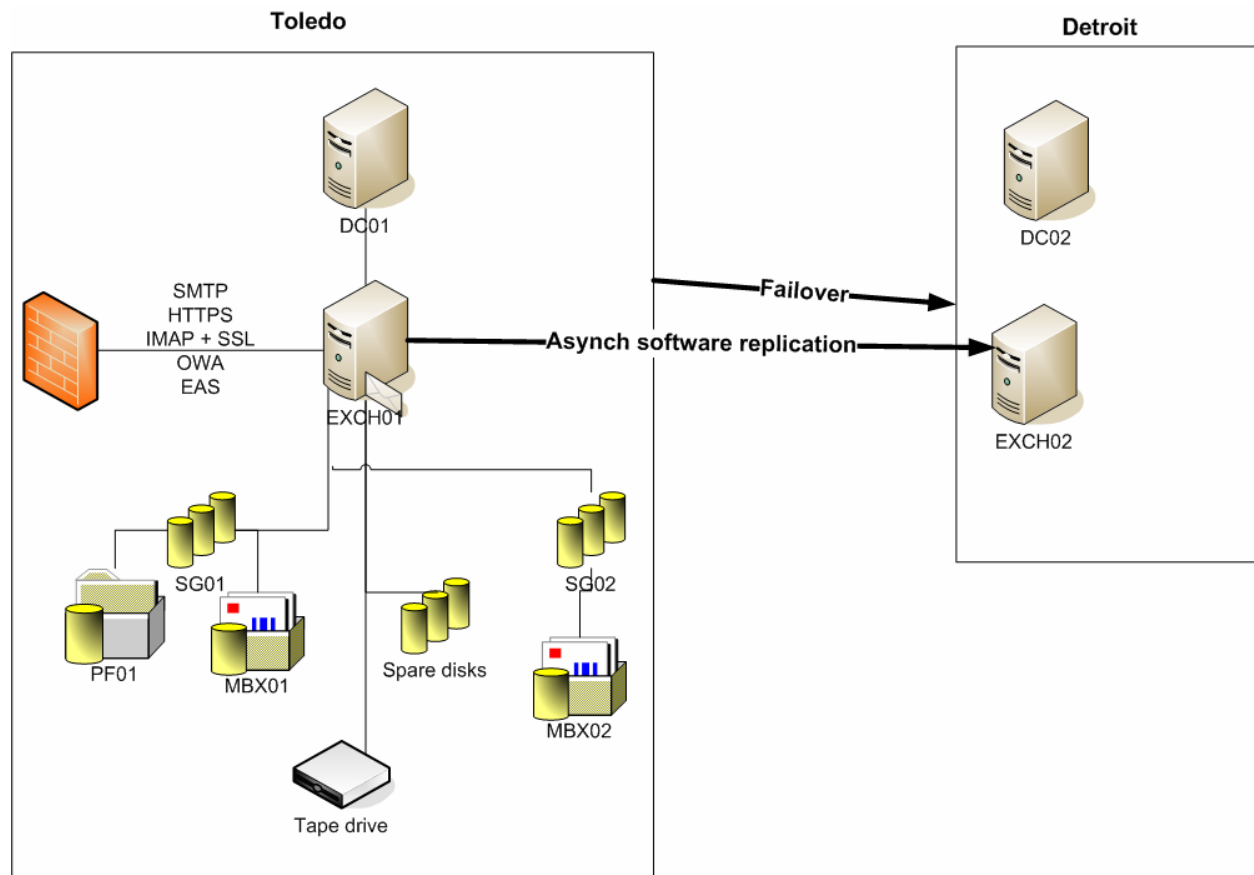


Figure 7.5: Improving the basic midsized company design.

The “async software replication” line has been drawn between the two Exchange servers to more clearly indicate what is being replicated between them: Exchange data. The spare server from Figure 7.3 has turned into a full-blown Exchange server, giving a quick way to provide Exchange service to Detroit or Toledo users if necessary. In addition, a failover solution is in place to provide the ability to quickly fail over operations to the remote site. Many replication products provide failover, which wasn’t shown in the preceding diagram because you would only use it if you wanted to also provide high availability.

Outsourcing Business Continuity

Most companies implement their own continuity services by buying replication and failover products from companies such as XOssoft. However, that requires you to invest in your own operations to a degree that some companies find daunting. Depending on your line of business, you might find it desirable to add business continuance capability by using third-party providers. These providers offer a wide range of services; large providers such as IBM Global Services and Sungard can set up a complete alternative data center for you, while smaller providers such as MessageOne focus on providing hosted business continuance services that give you messaging and mobile device access during an outage and the subsequent recovery.

How do you choose a business continuance service provider? Very carefully and with due deliberation. Having a poor business continuance service provider is much worse than not having one at all because the fact that you have a provider may lull you into thinking that everything is OK—until you have a failure, at which point you’ll be in big trouble.

There are some specific questions you should ask any business continuance service provider that you’re thinking about hiring. First, you want to do standard business due diligence: how long have they been in business, who are their major customers, and so on? These are important questions, however, the more important questions revolve around their capabilities: can they do what they claim? The best way to find out is to talk to other customers. If the service provider won’t provide references, that in itself should be a warning sign.

Another issue to consider is how often you can test the provider’s services. For providers such as MessageOne, where there is no need to activate a separate physical data center, the provider is likely to let you have several tests per year; for a full-up alternative-site recovery program, though, you may be limited in how often you can test and how long those tests can take.

Perhaps the ultimate measure of a business continuance service is the SLA they’re willing to commit to (and the penalties they’re willing to submit to). Ultimately it doesn’t matter how good a game the provider can talk unless they can deliver what they promise, and a strong SLA backed by penalties is a good way to validate their claims *before* you have to rely on their services.

Scenario 3: Large Budget

For large-budget organizations, there is a wide variety of potential solutions for raising the high-availability and business continuity capability of the messaging system. For example, you probably want to start with an arrangement like the one shown in Figure 7.4 for the primary office; clustering, a SAN, and multipathing combine to provide solid high-availability capability. For improved high availability and continuance, you could move to an architecture like the one shown in Figure 7.6; this diagram omits some of the details you already saw in Figure 7.4, but it adds some new features too. The chief change is the addition of hardware-based synchronous SAN replication between Houston and Dallas, using a dedicated link. Because synchronous replication increases write latency on the target, a fourth node was added to the cluster so that there are three active nodes (each with a smaller number of users than in Figure 7.4). For additional resiliency, asynchronous software replication is used to mirror a subset of the Exchange data from the Dallas SAN enclosure to a hot-standby server in San Francisco. This setup provides operational capability in case some large-scale outage occurs in Texas (or the surrounding area, as in the 2005 hurricane season).

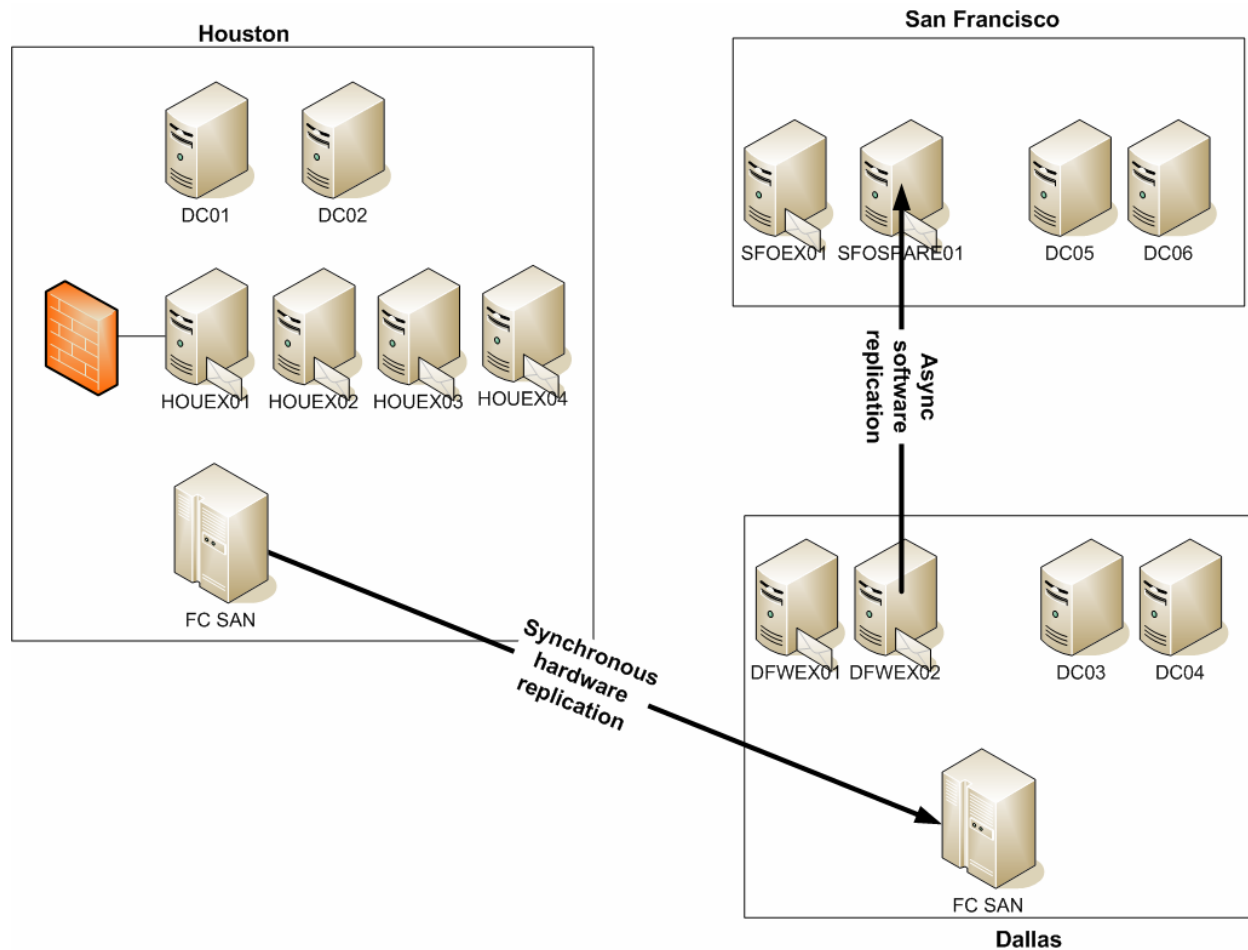


Figure 7.6: Adding business continuance capability with hardware SAN replication.

Figure 7.7 shows a significantly different blueprint: notice that there is no longer a synchronous replication connection from SAN to SAN. In fact, there is no SAN at all in Dallas or San Francisco; the primary SAN remains in Houston, but the four-node cluster formerly in Houston has spread across two *parts* of Houston: the primary data center downtown and a backup facility in Clear Lake. The San Francisco and Dallas offices are pretty much the same as they were before.

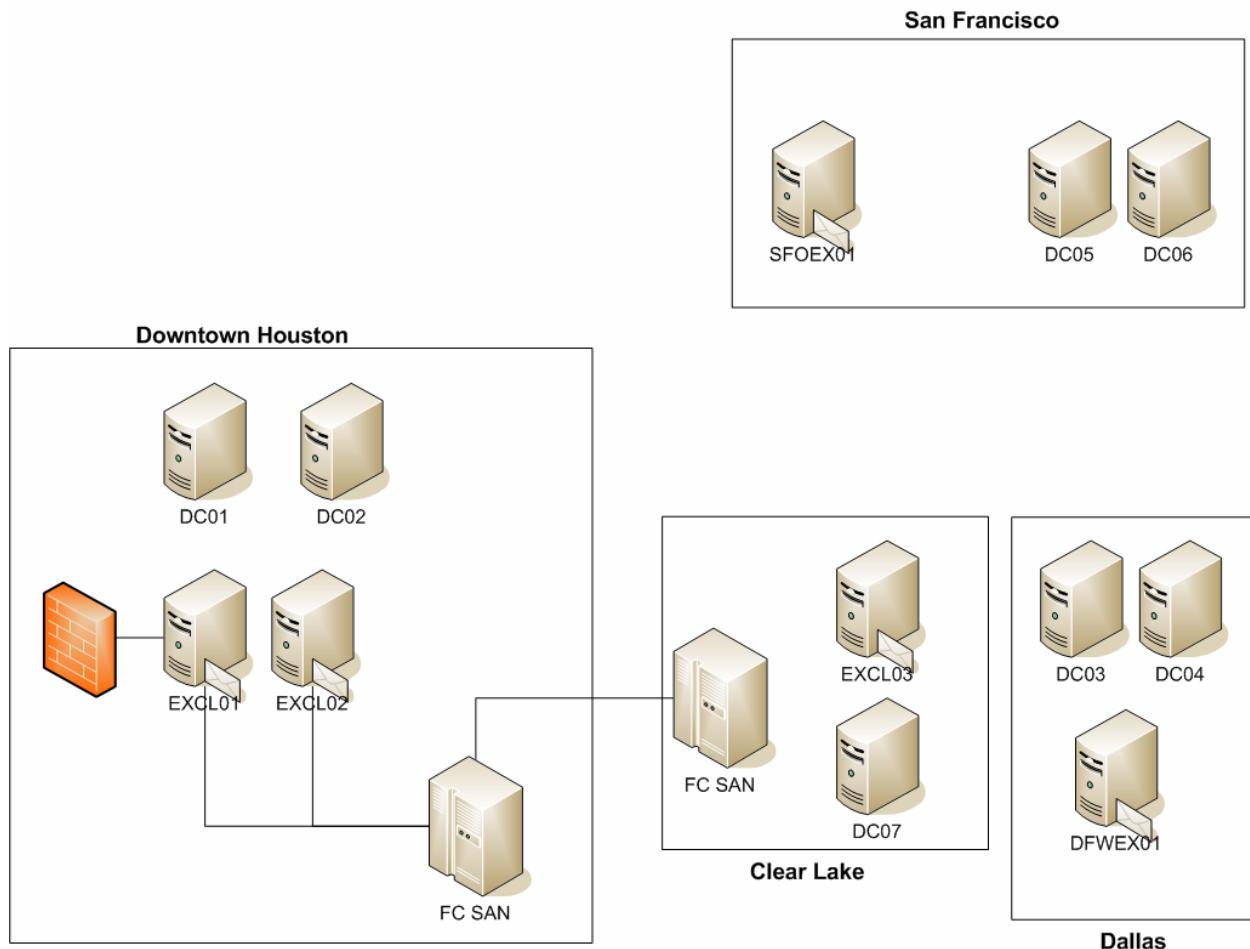


Figure 7.7: Adding business continuance capability with a geographically dispersed or “stretched” cluster.

The big difference is the addition of a geographically dispersed cluster. This is an extremely expensive technology, and it has some fairly stringent limitations. The chief limitation is one of distance; because Microsoft recommends keeping write latency for Exchange as low as possible (certainly below 100ms). Even though light is fast, the practical limit on synchronous fiber-based replication is about 100Km. Cisco and other communications hardware vendors offer Fibre Channel over IP (FCIP) solutions that allow you to evade this distance limitation (or at least stretch the limits considerably) by using extremely high-bandwidth WAN links. Such an approach provides excellent resilience and continuity potential, at a high implementation and maintenance cost. This solution isn’t recommended for any organization unless they can demonstrate both the technical competency needed to successfully manage it *and* a genuine business need for uptime that justifies that magnitude of capital outlay.

Solutions for Validation, Testing, and Deployment

Developing blueprints for validation, testing, and deployment is relatively easy compared with the technical challenges of designing and implementing a complete high-availability, business continuance, and disaster recovery solution. However, the validation, testing, and deployment cycle is no less important, and you should treat it accordingly. There is no effective way to provide you with a canned validation, testing, and deployment blueprint because the size and characteristics of each organization will vary, and those determine what the validation, testing, and deployment blueprint looks like.

Having said that, there is no real need to talk about *separate* validation, testing, and deployment scenarios for different sized companies. Why? First, every design needs validation, testing, and deployment, whether it's for a 10-person architect's office or an industrial behemoth with 15,000 mailboxes. For validation, unless your organization has unusual competency in high-availability design, it's usually a good idea to take your proposed design and have it validated by an outside specialist. Vendors such as XOsoft who sell replication and failover solutions often have consulting or professional services teams who can help do this, or you can find a consultant with specific expertise in your business domain. Either way, an outsider will be better prepared to catch problems with your design early in the design cycle, before you've started ordering expensive equipment and software. In addition, by having your proposed design validated by someone with more high-availability, business continuity, and disaster recovery experience than you have, you'll get the benefit of their broader experience.

Testing is a slightly different story. Although outsiders can be very valuable when you're creating a test plan, ultimately, you have to live with the results of the test. For that reason, although it may be useful to have an outside team help you develop your test plan, you need to be comfortable with it to ensure that it adequately tests your proposed design for suitability.

By contrast, deployment is often where you can save the most money by hiring outsiders. This may seem like a paradox, but think about it: when you're going to do something more than once, it makes sense to learn how to do it using only your internal resources. However, for something that you'll only do once—like migrating from Exchange 5.5 or performing the initial deployment of your clustered SAN-based Exchange system—saving money by *not* hiring someone experienced in that particular task can end up as a false economy.

Summary

Putting together the technology pieces covered in the preceding chapters of this book is challenging, but you'll certainly benefit from taking the time to survey the available technologies to determine which ones make sense for your organization's requirements.

Content Central

[Content Central](#) is your complete source for IT learning. Whether you need the most current information for managing your Windows enterprise, implementing security measures on your network, learning about new development tools for Windows and Linux, or deploying new enterprise software solutions, [Content Central](#) offers the latest instruction on the topics that are most important to the IT professional. Browse our extensive collection of eBooks and video guides and start building your own personal IT library today!

Download Additional eBooks!

If you found this eBook to be informative, then please visit Content Central and download other eBooks on this topic. If you are not already a registered user of Content Central, please take a moment to register in order to gain free access to other great IT eBooks and video guides. Please visit: <http://www.realtimepublishers.com/contentcentral/>.