In This Chapter

- The Single Server Farm Environment
- The Two-Tier Farm Environment
- The Small Three-Tier Farm Environment
- The Multiple Farm Environment

It can be easy to think that SharePoint disaster recovery is a niche subject, a straightforward process, or, even worse, not worth the time to consider. Hopefully, the printed chapters of the *SharePoint 2010 Disaster Recovery Guide* have done their job and dissuaded you of notions like those. In fact, by this point if your head’s spinning a bit from everything we’ve thrown at you, we completely understand. We’ve covered a lot of ground on SharePoint disaster recovery, and there’s a lot of information and solutions for you to consider and start applying against your own SharePoint 2010 environment. That process of applying the contents of this book to your own farm is the most important thing you can do with it.

As we said, we understand this is easier said than done. Trust us, after writing those print chapters, we know all too well how daunting, diverse, and difficult SharePoint disaster recovery can be. Making it even more difficult is the fact that it is not just a technical problem; it isn’t necessarily something you can solve by throwing a tool or a script at it. Effective SharePoint disaster recovery also requires careful planning; total buy-in and participation from your environment’s users, sponsors, and administrators; thorough testing; and much more. As information technology (IT) professionals, we know that it’s a common trap to focus on responding to challenges like these with the use of technology. That’s why we spent so much time at the start of this book talking about the nontechnical aspects of disaster recovery, in the hopes of grabbing your attention by conveying the importance of that information.
We’ve defined the situations discussed in the coming pages with the intent of pulling together both the technical and nontechnical concepts from Chapters 1 through 12 and utilizing them to show you how certain solutions can be used to properly protect an environment based on those concepts. Now, it’s important to remember that these are called “sample” scenarios for a reason: there is no one right way to solve any of these situations. Far too many variables can play into the design process for a disaster recovery scenario for us to be speaking in absolutes in this chapter. (Remember, the best answer to give for an IT-related question is usually, “It depends.”) Instead, our goal is to outline one potential solution for each situation and give you some insight into how that decision-making process works. That way you can apply those lessons to your own environment.

Much as a good rug can really tie a room together, our goal for this chapter is to help you coalesce the diverse and complex knowledge we’ve been throwing at you for the past 400 pages or so. The process we’ve gone through to plan these situations is similar to the process you need to be preparing to put your own organization through after finishing this book. The scale, scope, constraints, and resources may be different for you, but the design process should be comparable regardless of the other factors. Any other parallels you encounter in these situations are going to be gravy, so don’t get hung up too much on major technical differences you may find. Instead, think about how we came to the conclusions and choices we made and what you need to do to drive your environment and organization to a successful disaster recovery solution that’s right for you.

The scenarios that follow are listed in order of the size and complexity of the SharePoint 2010 environment in question, from least to greatest. Again, please keep in mind that these are samples, not absolutes. Your mileage for the specifics of a given example may vary, but the true intended value for them is in seeing the process, considerations, and decisions that we used to put each of them together.

**The Single Server Farm Environment**

This truly is the simplest of the many ways an organization can deploy SharePoint 2010: a single server running almost everything that SharePoint needs to function—including SharePoint itself, Internet Information Services (IIS), and some form of SQL Server—and storing all its content and data locally on the same server. To be specific, let’s highlight some details of the environment:

- **Database.** Microsoft SQL Server 2008 R2 Express Edition.
- **Web server.** Microsoft IIS 7.5.
- **SharePoint.** SharePoint Foundation 2010.
- **Authentication.** Active Directory is used to authenticate users, but this server is not a domain controller.

- **User base.** 25–50 users accessing the server via local area network (LAN) connections.

### The Situation

In each of the scenarios discussed in this chapter, we are introducing you to a fictional company to better illustrate the example and add a certain reality to what we’re talking about. In this case, our guinea pig is Foo Corp, a small business that has implemented SharePoint for document centralization and collaboration. Various departments within the company have their own site collections, but there is no centralized information architecture in place for the farm. Foo Corp has not customized the environment or installed additional templates or Web Parts. The farm was installed by Foo Corp’s IT administrator, who has limited experience with most server platforms and is really more of a desktop support engineer.

Foo Corp’s farm consists of a single server with SharePoint Foundation 2010 and SQL Server 2008 Express, both installed locally. (But don’t worry; it didn’t use the dreaded Basic installation option.) This server is a virtual machine running in a Microsoft Hyper-V Server physical host. It has one content SharePoint 2010 Web application hosting the three departmental site collections that its users collaborate in and upload content into. Because Foo Corp uses SharePoint as a centralized document storage utility, it is important to protect the content placed into it, but the company does not believe that is necessary on a frequent basis.

Although Foo Corp does store documents about its business in the environment, it is also careful to set scaled-back expectations for the SharePoint farm. Users understand that this is not a redundant system and do not expect high availability (HA) or resiliency. Also, because the farm supports Foo Corp’s business operations instead of directly implementing them, there is no direct urgency in how quickly an outage can be resolved or when a restoration needs to be completed in the event of a disaster. Foo Corp’s requirements for the farm’s recovery point objective (RPO) require that content be backed up every week, but its recovery time objective (RTO) is much more generous, allowing for up to two weeks to restore the SharePoint 2010 environment after a disaster.

### The Solution

Admittedly, we’re starting with a straightforward scenario without many big moving pieces. Everything is nicely located on one server for us, and because SharePoint hasn’t had customizations deployed to it, we can focus on the content that has been saved into it and protect it in the best manner possible. The interesting thing is that there are a lot of ways to solve this disaster recovery problem: backing up SharePoint, backing up SQL, and perhaps copying the virtual machine’s files on the host server. The good news is that you can consider and use any of these options, depending on the situation and its constraints.
Initially, Foo Corp seriously considered using the SharePoint Central Administration site to manually back up the farm every week, using the approach described early in Chapter 9, “SharePoint 2010 Central Administration Backup and Restore.” The driving force behind this proposed solution is that Foo Corp’s IT administrator was not comfortable with the complexities of SQL Server backup or PowerShell scripting but felt good about using Central Administration site’s graphical backup tool. Her only other experience with backups in Foo Corp’s environment was scheduling regular backups on every server with the Windows Server Backup tool.

Luckily for Foo Corp, cooler heads prevailed and pointed out the dangers of using an unscripted, unscheduled process as the key component of the disaster recovery plan to protect its SharePoint farm, a risk that is in no way worth the convenience of the Central Administration site’s backup tool. Instead, Foo Corp opted to go with the Windows Server Backup tool to back up both its server and its SharePoint environment. As you may have noticed in Chapter 5, “Windows Server 2008 Backup and Restore,” we did not discuss how to use Windows Server Backup directly to back up an entire SharePoint farm; without additional configuration, it can’t. Sure, it could back up the data files of SharePoint’s databases, but not the farm itself.

But Foo Corp did some research and discovered that by taking advantage of an STSADM.exe operation called RegisterWssWriter, it could configure its server’s Volume Shadow Copy Service (VSS) writer to use it with SharePoint. (For more information on VSS and how it works, make sure to review Chapter 11, “SharePoint 2010 Disaster Recovery Development.”) The use of this operation allows the Windows Server Backup to back up not just the operating system and its files, but the SharePoint environment installed on it. Once Foo Corp considered this finding, the company realized it was a viable solution providing the company with all the proper functionality and was still within the means of its technical capabilities and resources to deliver.

What’s Good About the Solution
Let’s look at why this approach works well for Foo Corp.

- **Simplicity.** The most important determining factor in this equation for Foo Corp is its need for a solution that protects its SharePoint environment in a simple, repeatable manner and can be successfully executed by its administrator. Because she had already exhibited experience and success with the Windows Server Backup tool, Foo Corp’s IT administrator was able to easily replicate her configuration on the company’s SharePoint server, taking care to also include SharePoint Foundation 2010 in the backup that she was scheduling on the server.

- **Effective coverage.** Foo Corp’s top priorities are the documents and data that it’s storing in SharePoint. The Windows Server Backup tool allows the company to easily protect those assets through backups of both the server’s file system and SharePoint itself.

- **Consistent solution.** As we’ve already mentioned, this is a tool already in use throughout Foo Corp’s servers, and there is definite value for Foo Corp in being able to reuse a solution that is consistent with its existing backup strategy.
What Needs to Be Improved

As with most IT solutions, there are certain risks and trade-offs to account for with this approach. There’s nothing wrong with that, and in fact our purpose in this and the other examples to come is to highlight some of those risks and drawbacks so you can better understand them and their potential impact on your own SharePoint 2010 disaster recovery solution.

- **Single point of failure.** There is a definite risk in using only one solution to back up an environment. Even though the IT administrator is comfortable with the Windows Server Backup tool and used it successfully in the past, that doesn’t mean its backup tool is going to work 100 percent of the time. The problem for Foo Corp is that it doesn’t have the means to augment its IT staff with someone who has experience with other options like SQL Server backups or PowerShell scripting, so it’s a risk the company is willing to take. (Also, the company wasn’t aware of outstanding resources on the subject that may be available, such as *SharePoint 2010 Disaster Recovery Guide.*) Foo Corp does plan to at least schedule simple full-farm backups with the `Backup-SPFarm` cmdlet on a regular basis for additional coverage, but it needs to resolve some storage space issues before setting that up.

- **Lack of expandability.** Microsoft recommends against using Windows Server Backup to protect SharePoint data in multiserver farms. Should Foo Corp choose to expand its farm down the road, the company needs to decide if it wants to modify its disaster recovery strategy to go against Microsoft’s recommendation and update its use of the Windows Server Backup tool to back up the SharePoint farm so it can accommodate multiple servers. Alternatively, it could use a different solution to back up its SharePoint 2010 environment and retain the Windows Server Backup tool to protect the other files and platforms on its servers.

- **Requires manual storage management.** Because there is no built-in functionality in the Windows Server Backup tool to archive or delete out-of-date backup files, Foo Corp’s administrator must handle this task in addition to running backups by hand. Beyond simple management of the storage files, Foo Corp is struggling to find enough space to store more than two or three full backups at a time, something it wants to do to preserve a historical view of its content. Foo Corp is currently evaluating Amazon’s Simple Storage Service (also known as Amazon S3) as a possible remote storage location for its historical backup files.

Wrapping It Up

It looks like doing its homework has really paid off for Foo Corp. By leveraging the RegisterWssWriter operation for STSADM and the Windows Server Backup tool, the company is able to protect its SharePoint 2010 farm with a solution that it’s already using to protect other servers in its environment. Sure, there are some long-term issues with expandability and storage management that Foo Corp needs to address, but RegisterWssWriter meets the company’s current needs—something that a lot of smaller organizations have to focus on to survive. Foo Corp must understand that as the usage of its SharePoint farm increases, so does the need to re-examine its SharePoint disaster recovery plan, but for the time being, it is in a pretty good position.
The Two-Tier Farm Environment

Now we begin to add a few more moving parts into the equation by separating the hosting of SharePoint’s databases to a separate server and leaving all of SharePoint’s roles on the other. Here are the details of the environment:

- **Web server.** Microsoft IIS 7.5.
- **Authentication.** Active Directory is used to authenticate users, but these servers are not domain controllers.
- **User base.** 50–100 users accessing the server via LAN connections.

The Situation

The organization using this SharePoint Server 2010 environment, we’ll call it Bar LLC, is a small business that has implemented SharePoint for its document management and collaboration capabilities to make it easier to update and share documents among its employees. Teams have their own sites, and a central site has been created to contain corporate news and information. Bar LLC has kept the environment simple, with no third-party customizations or code having been deployed to it. Bar LLC doesn’t have a large IT staff; the farm was built by its lone IT administrator who is well versed in Windows Server and SQL Server administration but has limited experience with SharePoint. Their users are storing business-critical content in the environment and using it more and more heavily on a daily basis, so essential data and resources that need to be protected are being stored in SharePoint.

Bar LLC’s farm consists of one SharePoint Server 2010 server, and its databases are hosted in Bar LLC’s central SQL Server 2008 instance. It has one content SharePoint 2010 Web application that hosts all the site collections that its users collaborate in and upload content to. This Web application hosts five site collections: one serving as an intranet home page, three departmental home pages with collaborative subsites, and one site collection hosting collaboration and documentation for a short-term project to implement a new customer relationship management (CRM) product for Bar LLC. The intranet home page is set as the default browser home page for all of Bar LLC’s employees. The home page and the three departmental sites are accessible by all of Bar LLC’s employees and contain critical business data that cannot be lost.

Much like Foo Corp from the previous example, Bar LLC has made sure to set its expectations for the SharePoint farm realistically. Again, there is no anticipation of HA or resiliency. Bar LLC also has lenient requirements for the time needed to resolve an outage, although its requirements are more stringent than Foo Corp’s. But because the farm is used to support Bar LLC’s business
operations rather than directly implement them, there is no direct urgency in how quickly an outage can be resolved or when a restoration needs to be completed in the event of a disaster. Bar LLC requires content to be backed up every 24 hours for the farm’s RPO, but its RTO is more flexible, allowing for up to two weeks to restore the SharePoint 2010 environment after a disaster.

The Solution
Yes, this is another fairly simple problem to solve, but it’s important to understand that the equally simple solution for Bar LLC can be used in a variety of situations, as long as requirements, expectations, and configurations are scaled back to meet the same requirements for simplicity. In this case, the best solution for Bar LLC is to back up its farm’s databases directly in SQL Server using the database platform’s built-in backup tools. (For more information on these tools, see Chapter 7, “SQL Server 2008 Backup and Restore.”) Given the limited resources and experience with SharePoint, this solution allows Bar LLC to focus on protecting the most important aspect of its SharePoint environment—its contents—using a tool its administrator is comfortable with.

What’s Good About the Solution
This approach works well for Bar LLC for several reasons:

- **Content protection.** It’s been mentioned a few times already, but the critical component of this system that needs to be backed up is its content. There are no customizations to take into account. SQL Server backups of the farm’s databases ensure that its contents are preserved on a regular basis and that its other databases are available either for restoration (in the case of the configuration database; see Chapter 10, “SharePoint 2010 Command Line Backup and Restore: PowerShell,” for more information on how to accomplish this with PowerShell) or for reference (such as service application databases).

- **Scheduling.** Because SQL Server backups can be scripted and scheduled, the process of backing up the farm’s databases can be automated. Automated backups remove the human element from the disaster recovery process, ensuring a higher level of consistency and success.

- **Consistent with Bar LLC’s other disaster recovery solutions.** Some of Bar LLC’s other platforms are already backed up through SQL Server, allowing for possible reuse of scripts or processes to back up the farm’s databases. This also makes the backup process more manageable, because administrators should already be aware of this process for other platforms.

- **Familiar technology.** Yes, scripted backups of the SharePoint 2010 farm can be created using SharePoint 2010’s new PowerShell cmdlets (again, see Chapter 10 for more information), but Bar LLC’s administrator is far more comfortable with SQL Server than SharePoint 2010 and PowerShell.
What Needs to Be Improved

But that’s not to say that SQL Server backups are a perfect solution. There are some drawbacks to the approach, which Bar LLC is currently at ease with. However, you should re-examine this approach as use and adoption of the SharePoint 2010 environment grow over time.

- Single point of failure. Just like Foo Corp, Bar LLC faces a definite risk in using only one solution to back up an environment. What happens if the SQL Server backup process fails or its backups become corrupted? Although the likelihood of this is low, Bar LLC plans to address this issue by thoroughly testing and implementing scripted SharePoint backups via PowerShell cmdlets in the next six months as it builds greater competency and comfort with SharePoint administration.

- Requires generous RTO. Because the focus of this approach is on protecting the farm’s content, it does not consider how to completely cover its configuration and setup data. To restore the farm using SQL Server backups, take the time to install SharePoint on a new server, build a new farm, and restore these backups into it. In this circumstance, you must measure recovery time in days, not hours or minutes, to account for the time necessary to rebuild the farm.

- Places high emphasis on change management. Again, because of the emphasis on safeguarding the farm’s content, you must take other means to track the way it is configured. Yes, you can recover some data from a farm’s configuration database (see Chapters 9 and 10 for more information), but not all of it. Bar LLC must record and track all configuration changes made in SharePoint so that they can be replicated should it be necessary to rebuild the environment.

- Not full coverage. Several aspects of SharePoint are not backed up via SQL Server backups, such as the SharePoint root and the farm’s Internet Information Services (IIS) Web sites. It may not be possible to restore items such as these to their proper state during a rebuild activity. This is another area where Bar LLC has the same challenges as Foo Corp and needs to work on implementing an approach for protecting the aspects of the farm not included in its databases. The information provided in Chapter 5 should be highly useful for this exercise.

Wrapping It Up

So there we go: one possible solution for how Bar LLC can protect and recover its SharePoint Server 2010 farm should a disaster strike. Yes, it’s a simple solution, and it is not without its drawbacks, but it meets Bar LLC’s current needs and allows the farm’s disaster recovery solution to grow and mature as the farm grows and matures. It’s just as important to match a SharePoint disaster recovery solution to a company’s resources as it is to develop a solution that meets all its needs. Using SQL Server backups plays the strengths of Bar LLC’s IT administrator, but Bar LLC also identified several areas for improvement or enhancement for this solution during its
creation. The company is now planning to deliver on those items in the coming months and adapt to allow for the way increased usage and adoption can change the requirements in that period.

The Small Three-Tier Farm Environment

Now we’re getting somewhere—a more complex scenario with a lot more moving pieces and things to consider. This time we’re going to look at Acme & Co., a large national company that’s seriously off and running with SharePoint. Let’s examine the details of Acme’s environment:

- **Database.** Microsoft SQL Server 2008 R2 Standard Edition.
- **Web server.** Microsoft IIS 7.5.
- **Authentication.** Active Directory is used to authenticate users, but these servers are not domain controllers.
- **User base.** 2,000 users accessing the server via LAN and wide area network (WAN) connections.

The Situation

Acme & Co. has really gotten behind SharePoint; the company has been using the software for a long time now and has moved several of its critical business functions to the platform. Yes, Acme definitely uses it for its corporate intranet and collaboration, but it’s also taking advantage of SharePoint’s diverse development platform to host custom line of business (LOB) applications designed to manage orders and track permit compliance, among other things. Because of the vital business data that Acme stores in SharePoint, it has set up its environment so that it is highly available and backed up.

To ensure that Acme’s farm is able to both deliver the performance that its users require and do so consistently, the company has separated its farm into three tiers and made a point of provisioning multiple servers in each of these tiers. At the top of the stack is the Web tier, where it has placed two virtualized SharePoint servers and assigned each of them both the Web front-end (WFE) and Query roles within the farm. It is using Windows Network Load Balancing (NLB) to distribute traffic between the two servers. Next is the application tier, with one virtualized SharePoint server dedicated to hosting the functionality of Acme’s LOB and service applications and another physical SharePoint server assigned the Crawler search role. Finally, Acme’s SharePoint databases are in the third tier, hosted primarily on one SQL Server 2008 R2 instance and mirrored over to a second SQL Server 2008 R2 instance (both running on physical servers) to ensure that the farm’s databases are able to still be available should the primary server suffer an outage.
In Acme’s approach to configuring its SharePoint 2010 farm, we can see some of the more advanced HA solutions that are covered in Chapter 6, “Windows Server 2008 High Availability” and Chapter 8, “SQL Server 2008 High Availability,” like Windows NLB and SQL Server database mirroring. These are both concepts that Acme is comfortable with, as they have an experienced IT department that supports a range of applications running on Windows servers. Acme also knows that it cannot afford to lose the content and data that it is placing in its SharePoint environment. Acme may be able to live without it for a few days (it has an RTO of three days), but the company’s ability to do business could certainly not withstand its entire loss, so it wants to make sure that it’s taking good care of it.

**The Solution**

We’ve already talked a little about how Acme is making critical aspects of their SharePoint farm more highly available to withstand outages with NLB to protect the user-facing WFE servers and mirroring to safeguard the farm’s databases. Therefore, let’s focus on how Acme is backing up their data and documents to ensure that they can be preserved should anything catastrophic happen to the farm. Acme has decided to take a two-pronged approach to backing up its SharePoint farm by targeting both the content of its SharePoint farm and the configuration of its custom applications running within it. Because the content is so important to the company, Acme has implemented two methods to back up that content: SharePoint farm backups via scheduled PowerShell scripts and SQL Server database backups via scripted maintenance plans. Acme is also using multiple solutions whenever it is possible to protect the farm’s configuration data and customizations: the file systems of every server in the farm are backed up with the Windows Server Backup tool with scheduled PowerShell scripts. Also, copies of the virtual hard drive files are made for every virtual server in the farm, which includes customizations because each version of a SharePoint solution package is saved on the farm’s Application server once it has been deployed to the farm.

**What’s Good About the Solution**

Now let’s look at why this approach works so well for Acme.

- **Content protection.** The most important thing about SharePoint to users is the content that they put into it, which is why it’s the first and most important aspect of the environment that an effective disaster recovery strategy needs to account for. Acme is not just accounting for its content; it has established multiple methods for backing it up, through SharePoint and SQL Server’s built-in tools.

- **Automation.** Acme also understands that it needs to automate the backup process to ensure that it is run on a consistent basis to provide constant protection. It has developed custom SQL Server maintenance plans and PowerShell scripts, and it has thoroughly tested those scripts, that meet the company’s specific needs for coverage, storage, and frequency of execution. Automation is also present in Acme’s database HA solution. By making SharePoint aware of the mirroring capabilities of its SQL environment, Acme is able to
ensure continuity of service through automatic failover should something happen to the primary database host.

- **Redundant.** Acme has been running its own internal IT long enough to know that eventually all systems and solutions encounter an outage or an error. To more fully protect its SharePoint farm, Acme has made redundancy a key aspect of its disaster recovery strategy. The farm’s WFEs are load-balanced to distribute traffic and allow the farm to remain available if one server goes down. The databases are mirrored so that they can remain online should the principal become unavailable. And Acme is making its backups redundant, too, both for the content of the farm and the setup of its servers.

- **Comprehensive coverage.** By making a point to back up the file systems of its farm’s servers, Acme is doing as much as it can to reduce the time it takes to restore service should an outage strike. Because of SharePoint’s unique architecture and design, backups of a SharePoint server’s file system are not a major shortcut, but they do capture the state of the farm’s IIS Web sites, the SharePoint Root directory, and the files for any solution packages that may be stored on a server prior to being deployed to customize the farm. These backups are especially important for those customizations, because they must be redeployed if a new farm has to be built after a disaster. Acme is also considering implementing configuration-only backups to capture these customizations and in general reduce its RTO, but first the company wants to test the process so it understands exactly what is and is not delivered by those backups.

**What Needs to Be Improved**

So what are some of the things that Acme needs to be mindful of and working to improve about its disaster recovery solution?

- **Good but not great.** Due to budget constraints, Acme had to make some trade-offs in how it designed its solution. The company considered hardware load-balancing applications, because they are more customizable and full featured than NLB, but ultimately they were too pricey. Acme also thought seriously about implementing clustering for its SQL instances instead of mirroring, but it couldn’t afford to make that kind of investment in its storage hardware. Acme is aware that it doesn’t necessarily have the best solution money can buy, but it feels the compromises were worthwhile and able to meet the company’s core needs.

- **Places emphasis on change management.** Even though Acme is backing up its entire farm and its databases, as well as the file system of every server in the farm, it knows that restoring SharePoint is not necessarily a simple matter of restoring all those backups. There is still a need for Acme to closely track and record all the changes it makes in the SharePoint environment so that it can replicate those changes in a restored environment after a disaster.

- **Not truly redundant.** Acme has only one datacenter and has decided not to invest in an off-site backup solution at this time. Yes, there is some redundancy in Acme’s solution, but it is not truly redundant if there are still single points of failure in it. By putting all its resources
in a single datacenter, Acme runs the risk of losing its entire environment should something happen to that datacenter, such as a power outage, network outage, or catastrophic disaster.

**Wrapping It Up**

Acme & Co. has done a pretty good job of building multiple layers of availability and protection into its disaster recovery solution. Most importantly, there’s a lot to like about how this solution uses NLB to make SharePoint sites redundantly accessible and isn’t putting all of Acme’s eggs in one basket with a single backup tool for valuable content. Yes, there are some things the company could stand to improve on, but what it has is by no means shabby. It protects its valuable content as well as the customizations required to implement its LOB applications. The farm’s key roles, WFE and database host, are set up to guard against the failure of a server and provide continuity of service. Acme should address some definite aspects of the system in the future, but what it has today is serviceable and cost effective. Acme also needs to challenge its IT organization to not become lax in its change management procedures, so that it can maintain useful and thorough records of how its environment has changed and grown over time and be able to restore the farm to its most recent known state should a disaster occur.

**The Multiple Farm Environment**

Finally, we turn to the circumstances of the Umbrella Corporation, a large multinational pharmaceutical company. It’s an interesting case because the regulations and oversight that it has to operate under put a much greater emphasis on having a comprehensive disaster recovery solution for its SharePoint environment. Because of that burden, Umbrella has to take several more steps to make sure that its farm is continually up and running. Here are the specifics of what Umbrella is doing to make that happen:

- **Database.** Microsoft SQL Server 2008 R2 Enterprise Edition.
- **Web server.** Microsoft IIS 7.5.
- **SharePoint.** SharePoint Server 2010 Enterprise Edition.
- **Authentication.** Active Directory is used to authenticate users, but these servers are not domain controllers. Employees of Umbrella’s partners also access the environment via an ASP.NET forms-based authentication provider.
- **User base.** 20,000 users accessing the server via LAN, WAN, and virtual private network (VPN) connections.

**The Situation**

Documentation is something that Umbrella must, and does, place a good deal of emphasis on. Whether it is capturing minute details of clinical trials or detailing the proper way to evacuate a
manufacturing facility, Umbrella needs to make sure that everything is properly categorized, secured, and discoverable for its employees. SharePoint was the tool that Umbrella selected early on to allow it to move to a digitally based solution for documentation, rather than a paper one. That has been a positive result for Umbrella, but it creates a great deal of dependence on SharePoint for its ability to do company business, meaning that Umbrella cannot allow for much downtime within the environment.

Because of that fact, Umbrella has made a serious investment in making its SharePoint environment redundant and highly available. Multiple servers are present in each tier of the farm, and no server is hosting more than one SharePoint role at a time. WFE servers are clustered behind hardware load-balancing devices. The farm also uses multiple Crawl servers to distribute the task of indexing its extensive content and redundant dedicated Query servers to respond to users’ search queries. The farm’s databases are hosted by multiple clustered SQL Server instances, allowing for instantaneous failover as well as distributing load across multiple instances. Finally, those databases are log-shipped to a second farm in a remote datacenter, which allows for even greater protection of Umbrella’s content and provides a warm standby farm if needed.

Obviously, Umbrella is sparing little expense when it comes to its SharePoint farm, and rightly so. Even losing it for a day can translate into a loss of more than $1 million, a nontrivial sum even for the largest company. Umbrella is doing several of the same things we saw in the case of Acme & Co., but it’s taking the approach several steps further in terms of redundancy and complexity of the solution. The approach gives Umbrella the solution, but it is not without its own drawbacks.

The Solution
As we alluded to previously, Umbrella Corporation’s solution is similar to what Acme has implemented, but with more of pretty much everything. From a load-balancing perspective, Umbrella has implemented hardware load balancers to provide clustered IP addresses for the farm’s four WFEs. Its three application servers are automatically load-balanced by SharePoint, and search crawling is handled by two Crawl servers. SQL Server is highly available, but with Windows failover clustering rather than database mirroring. The big difference for Umbrella is that the farm’s databases are replicated to a second SharePoint farm in a remote datacenter via SQL Server log shipping, which we’ll dive into deeper later. In addition to that log shipped failover farm, Umbrella is backing up the farm and the file systems of each server in it with scheduled PowerShell scripts and the Windows Server Backup tool.

While designing its disaster recovery strategy, Umbrella realized that its RTO was best measured in hours, or even minutes, not days or weeks. It was also apparent that the company needed to ensure its data was fully protected from a range of possible problems or calamities. Umbrella determined that the best course of action was to set up a second failover farm in a remote location and replicate its SharePoint content to that farm, giving it both a duplicate set of content and a fully functional environment to host it in if needed. This gave Umbrella a solution that could address both of those critical needs, but it was not without its detractors.
What’s Good About the Solution
Before we get into some of the issues in this multiple-farm solution, let’s look at why it works for Umbrella:

- **Content protection.** Umbrella’s documentation and data is protected both by SharePoint’s backups and by the database log shipping that replicates it to the failover farm.

- **Automation.** Most aspects of this disaster recovery solution are automated: load balancing, scripted backups, failover clustering, and log shipping. Umbrella has been able to successfully deploy a solution that drastically reduces the opportunity for human error.

- **Redundant.** Redundancy is something that Umbrella takes seriously, and it shows. No aspect of the system is allowed to be a single point of failure, whether it is a server, a switch, a storage device, a backup process, or a SharePoint role.

- **Coverage.** Because of that redundancy, as well as Umbrella’s implementation of a number of diverse backup solutions, the company is able to assure itself of complete coverage of its entire environment.

What Needs to Be Improved
So what are some of the things that Acme needs to be mindful of and working to improve about its disaster recovery solution?

- **Cost.** It comes down to this: solutions as exhaustive and redundant as what Umbrella is doing are not cheap. There’s the cost of every server, every license, every appliance, the storage, not to mention the space in the second datacenter and the bandwidth required to connect the two datacenters. In addition, there’s the cost to operate, administrate, and maintain each of these resources.

- **Complexity.** By adding all these moving parts and additional systems into the equation, Umbrella has really increased the complexity of the SharePoint environment. It has put several different technologies in place to support its disaster recovery strategy, and each one has to be managed and maintained in a different way. Adding this kind of complexity to a SharePoint environment makes it more difficult and more costly to operate, support, and maintain.

- **Difficult to maintain.** The most challenging aspect of this solution is the need to keep the primary and failover farms in sync with one another as the environment changes and grows. As changes are made to pieces within the primary farm, such as SharePoint patches applied to servers in the farm or a new solution package being deployed, the same changes must be made in the failover farm to ensure that the failover farm is always at the same level of functionality as the primary should a failover be required. This is much easier said than done and makes it much more challenging to administer the environment.
Wrapping It Up
The solution we’ve described for the Umbrella Corporation can certainly be effective, but it is definitely going to require an expansive budget for hard costs such as hardware, licensing, and hosting, as well as soft costs such as configuration, administration, and troubleshooting. The solution provides a high level of protection and redundancy, but it also comes with a great deal more complexity. This is not a solution that a company with a small, inexperienced IT staff could implement or maintain; it is a much better fit for a large organization such as the Umbrella Corporation. And remember, it is necessary because Umbrella is in a heavily regulated industry. It is important to remember factors such as those when designing a disaster recovery solution.

Conclusion
Well, that’s it. The examples we’ve walked you through are just four specific situations out of the hundreds that can commonly occur when implementing SharePoint, and the solutions we outlined are far from the only ways that you can handle the situations. You should not interpret these solutions as the absolute or best way to protect those SharePoint environments, because it’s not our intention to say that they are. Our goal with this chapter is to get you thinking about how to put the content from the first 12 chapters of this book to use in real situations, to show how to apply the tools and solutions in an actual SharePoint environment rather than just give you a how-to on each of them.

The other thing to keep in mind when reflecting on these examples is that one of the things that every organization wants for its SharePoint environment is for its usage and adoption to grow and expand over the life of the system. If that happens, your environment is certain to grow as well to accommodate that adoption. Our point is that if you feel like you have an environment that has more in common with the first or second example, don’t let that keep you from taking a close look at the larger environments as well. Who’s to say that your SharePoint farm isn’t going to be bigger a year or two from now? You may not want to skip over one or more of the more complex examples in this chapter. There could be more in common with your “simple” SharePoint environment (actually, there’s no such thing as a simple SharePoint environment) than you think.

Finally, we hope that this chapter has emphasized the importance of considering SharePoint disaster recovery as much more than just backing up a farm or load-balancing some servers. There are so many configurations to consider to create an effective solution; just trying to throw a tool at the problem isn’t going to fix it, regardless of what tool that is. You need to think about who is using your environment, how they’re using it, and when they need it. You need to make sure that you know how fast you have to have it back online in case of an outage. You need to know how much you can spend to make all of this happen and how much it takes to keep it all up and running. Long story short, it’s not simple and it’s not easy. But it’s also a heck of a lot better to get it right now before you truly need it, rather than later when things are going bad and they’re only going to get worse.