

Part I

Preparing for Windows SharePoint Services 3.0

Chapter 1

Windows SharePoint Services 3.0 Under the Hood

You've heard of Windows SharePoint Services 3.0. You've probably seen webcasts and presentations about it, but what is it and why should it matter to you?

Windows SharePoint Services (WSS) 3.0 is a nifty web-based collaboration, data management, communication, idea-creating, problem-solving tool that costs you nothing. Windows SharePoint Services, which is usually referred to in the singular, needs to run on Windows Server 2003 (Service Pack 1 or higher, or Release 2 if you have it) and should be a server role in Server 2008 (which is in beta at the writing of this book).

WSS has its needs, its shortcomings, and its weaknesses, but overall, it is a surprisingly useful, flexible, powerful web-based tool for any administrator. The best part is that using it doesn't require any web-development skills at all. As a matter of fact, this book is being written for IT admins specifically because they seem to be the people who are ultimately responsible for managing SharePoint, without really being trained for it. This book should help fill in some of those holes in training.

So what is SharePoint? SharePoint comes in two flavors: WSS 3.0 and Microsoft Office SharePoint Server 2007 (MOSS). WSS 3.0 is free and is considered a Windows Server 2003 server role which falls under the server's license model. However, MOSS, which installs on top of the free version, costs thousands of dollars (depending on volume license) and requires a Client Access License (CAL) for each user. The free version doesn't require separate client licenses for each user and is the foundation for SharePoint. The paid for version just adds more functionality to the foundation. So yes, Windows SharePoint Services is free and the foundation for the more expensive MOSS components.

What does SharePoint do? It presents a web interface for people to collaborate, communicate, and share data in an environment that is consistent, easy for administrators to control, designed to store data and documents, and is very scalable. SharePoint can be installed on a single server or it can be installed on numerous web front end servers sharing the client load on what is called a SharePoint *server farm*.

Fundamentally, SharePoint is a bunch of web pages with web parts and lists on top of a database. However, SharePoint takes advantage of that simple framework and uses it to offer lists, libraries, workspaces, wikis, blogs, and web parts. With these tools, you can offer shared calendars, discussions, file libraries, surveys, and more. For process management, you can require document checkout, content approval, and versioning. You can even establish workflows to trigger alerts and other changes based on where documents or list items are in a process. Lists and libraries can be set up with their own email accounts, so people can email entries without going to the SharePoint site.

SHAREPOINT DOESN'T DO SHARE POINTS?

WSS uses *content* databases to contain its data. It's a great way to store and organize large numbers of records, documents, photos and more. However, it is not intended to be a web front end for aggregating file shares, despite the fact that file shares are also sometimes called share points. So SharePoint does not have anything to do with share points. There are web parts and other page attributes that can point to file shares, but that is not the primary purpose of SharePoint.

This book will cover the ins and outs of Windows SharePoint Services 3.0 to give you the best bang for your buck and the most information about what you can get from the free version before you buy the expensive versions.

In this chapter, you'll learn how to:

- ◆ Determine the software and hardware requirements you need for installing SharePoint Services 3.0
- ◆ Identify the three ways of installing SharePoint Services 3.0
- ◆ Set up the necessary accounts that SharePoint needs to run
- ◆ Recognize the new features and requirements of SharePoint

Software Requirements

To make all that SharePoint goodness possible, the following roles and technologies must be installed and running on the SharePoint server. These are the underlying technologies that make SharePoint function. Without them, SharePoint won't even install.

Internet Information Services (IIS) 6.0 (or Higher) SharePoint is web-based because IIS allows a Windows Server (2003 or higher) to host websites and service HTTP requests from clients. Many SharePoint capabilities are dependent upon and colored by the functions and needs of IIS. For example, IIS contains Web Sites, that hold web pages. In SharePoint, IIS's Web Sites are considered to be *Web Applications*, formerly called *virtual servers* in WSS 2, and contain web pages organized into sites and subsites, called *site collections*. SharePoint Web Applications are considered containers and security boundaries for those site collections, largely because of the built-in properties of IIS's Web Sites and their management (for example, specifying application pools and whether or not anonymous access is allowed). Those settings may be configured in SharePoint, but are applied at the IIS Web Site (ala web application). This explains why anonymous access is enabled at the web application level and then trickles down to each site collection contained within. The IIS server role must be installed before SharePoint can be installed.

An additional SharePoint feature that depends on IIS is incoming email, which requires that the SMTP service is enabled in IIS.

ASP.NET 2.0 ASP.NET is required to create and run web parts and other components of SharePoint web pages (as well as compile the pages themselves). It must be installed and enabled in IIS before SharePoint will install properly. ASP.NET 1.1 will also need to be installed for backward compatibility. ASP .NET 2.0 can be installed separately, or as part of the .NET Framework 3.0 installation.

.NET Framework 3.0 Required in order to install SharePoint, this service contains the Windows Workflow Foundation, a useful part of list management and document processing.

Windows Workflow Foundation Although not something that can be installed by itself, this part of the .NET Framework 3.0 is required for SharePoint to work properly.

IT'S NOT FOR WORKSTATIONS

SharePoint can't install just anywhere on just any operating system. It requires Windows Server 2003 SP1 (Standard, Enterprise, Data Center, or Web Edition) or higher. It also requires NTFS. It is not recommended to install, it won't install on FAT32. It supports x86 and x64, although the installer for either version is still sharepoint.exe, so be careful what you download. SharePoint on a domain controller. Also, Windows Server 2003 Web Edition cannot host a database, but it can hold SharePoint. Therefore you can install SharePoint as a web front end server on the Web Edition of Windows Server 2003, but you cannot install the standalone version.

In addition, somewhere on the network, depending on how you install it, there has to be a version of SQL server for SharePoint to access. There are basically two types of SQL you can use:

SQL Server SharePoint supports either SQL 2000 with at least SP3 or SQL 2005. This pricey package is a database powerhouse. Network aware, it can be made to support clustering and more. It is ideal for handling the huge amounts of data a large server farm might generate. SQL Server is possibly overkill for small offices who are considering SharePoint. However, if you already have SQL Server 2000 SP3 or 2005 on your network, then by all means use it.

SQL Server 2005 Embedded Edition This edition is also called the Windows Internal Database. If you don't have SQL handy (and don't want to shell out the cash to install and use it), you can do the poor man's single SharePoint server install, as discussed in Chapter 2, "Installation." This will install SQL Server 2005 Embedded Edition (SSEE) during SharePoint's initial setup. SSEE is a local only (cannot be remotely accessed), free database, which is a modified version of SQL Server 2005 Express and essentially the newer version of WMSDE. With SSEE, SharePoint can create and manage its databases just fine. The catch is that the embedded version of SQL cannot support any other SharePoint servers accessing it. It is not as robust as its big brother SQL 2005, and it has no graphical tools built in with which to manage and update it.

CLIENT-SIDE SHENANIGANS

Of course, from the client side, users will need a browser to access the SharePoint sites. Microsoft says that SharePoint has two levels of browser support: Level 1 and Level 2. Level 1 browsers support Active X controls, namely Internet Explorer 6.0, 7.0, or higher. Level 2 refers to all other browsers.

SharePoint is optimized for Level 1 browsers (no surprise there) and supports everything you might ever want to do in SharePoint. Level 2 browsers support only non-Active X activities and are generally limited to reading and adding text to fields. Your performance may vary, depending on what customization

and development has been done on your sites. The bottom line is, Microsoft wants you to use IE to use SharePoint—that and Office 2007, of course.

Office 2007 is incredibly integrated with SharePoint; half the things you can do with SharePoint you can do *better* with Office 2007. Office 2003 can do integration too, but not as completely as Office 2007.

It's important to realize how pivotal SQL is to SharePoint. In addition to hosting nifty-looking websites, SharePoint's real primary purpose is to store and access data from its databases. SharePoint is really an extensive database front end. It's all about lists (and a special kind of list called a Library). Lists contain data in records and fields (or, visually, rows and columns). Therefore, SharePoint logically requires databases on the back end to hold all that data.

As you know, SharePoint does not necessarily need to be installed on the same server as the databases themselves, although it can be if you need it. That is the beauty of SQL server: it can be accessed remotely. This means that a SharePoint server just needs to be pointed at a nearby SQL server to create and use a database there. This is convenient for several reasons, such as separating resources and storage, helping eliminate the SharePoint server as a single point of failure, and scalability. If a SQL database can be accessed by one SharePoint server, then it stands to reason (with maybe a little tweaking) that other SharePoint servers can access the same database. This is what makes server farms possible. Using this approach, multiple installations of SharePoint can be pointed to the same configuration and content databases, so they can do load balancing and share the same consistent configuration and administration settings.

This is obviously why SharePoint requires SQL. This is also where you see a functional split between installing SharePoint to be hosted by a single server and installing SharePoint to be managed across a server farm. Single server installations only need local access to a database, and they can easily use SSEE to accomplish that. A server farm requires a remote SQL server that all SharePoint front end servers can share.

So there you have it, that's SharePoint's foundation; IIS 6.0 or higher, ASP .NET 2.0, .NET Framework 3.0, and SQL Server 2000 SP3 or higher (or you can let SharePoint install SSEE). These roles and technologies, working in tandem, power SharePoint. The strengths and weaknesses of this underlying infrastructure lend their particular traits to SharePoint. Knowing about them teaches you both how SharePoint works and how to manage it, especially when it comes to troubleshooting.

Now that you know SharePoint's critical components, there are other considerations you need to cover before you install it.

Installing SharePoint: Single Server and Server Farm

SharePoint may come in two sizes, but it can actually be installed three different ways: Basic, Stand-alone Server, and Server Farm. The last two options are under the heading "Advanced."

Basic The Basic install assumes that you are going to use only one server *ever* to run SharePoint and that you don't have a copy of SQL handy to use for its databases. What it does in that case is install SharePoint assuming all necessary services are going to run locally and that you need it to install the free "Windows Internal Database," which is Microsoft's nickname for its SQL Server 2005 Embedded Edition database (SSEE).

DESKTOP DATABASE PRIMER

MSDE was Microsoft's free desktop database engine, originally for developers to run on their workstations to develop SQL applications without having to have a copy of the expensive version of SQL. It had a 2GB limit, could have a maximum of only five concurrent users, could not be accessed remotely, had no search indexing capabilities, and had a few more limitations.

WMSDE, or MSDE (Windows), was created by Microsoft to be the built-in database back end for some of their free but necessary products, such as Microsoft's Windows Server Update Service (WSUS) and Windows SharePoint Services 2.0. The WMSDE version of MSDE unlocked the 2GB limitation, but it still did not have full-search indexing or remote access capabilities.

WMSDE was sort of "embedded" in those free Microsoft products, meaning it was transparent, installed invisibly; and was a critical part of those products, so much so that they installed it automatically (well, WSS 2.0 did if you told it to).

This explains why the newest version of WMSDE that installs with the SharePoint version 3.0 Basic install is called SQL Server 2005 Embedded Edition (SSEE) or Windows Internal Database (WID). It is a slightly modified version of the WMSDE update, called SQL Server 2005 Express Edition, which has been surprisingly improved and supports search, remote access, and Windows authentication.

If you perform the SharePoint Basic installation, the SSEE database cannot be used by any other SharePoint server on the network. You will not have remote access. There is absolutely no way that you can do a more complicated, multiple server installation of SharePoint using the SSEE database. Unfortunately, an SSEE database is not quite as robust as SQL server databases. For most people, this only means that as the databases fill up more quickly and become more awkward and slower, so it is important to closely monitor the content database of a Basic installed SharePoint server (use database site quotas and quota templates for site collections judiciously). That being said, a lot of small to medium businesses use the Basic install of SharePoint without any problems, and they have the bonus of saving so much money by not buying SQL. Basically, if you plan to never have more than 10 separate web applications in your company, and plan to have only one SharePoint server, then using SSEE would be fine for you. Web applications (and their databases) will be discussed in greater detail in Chapter 8.

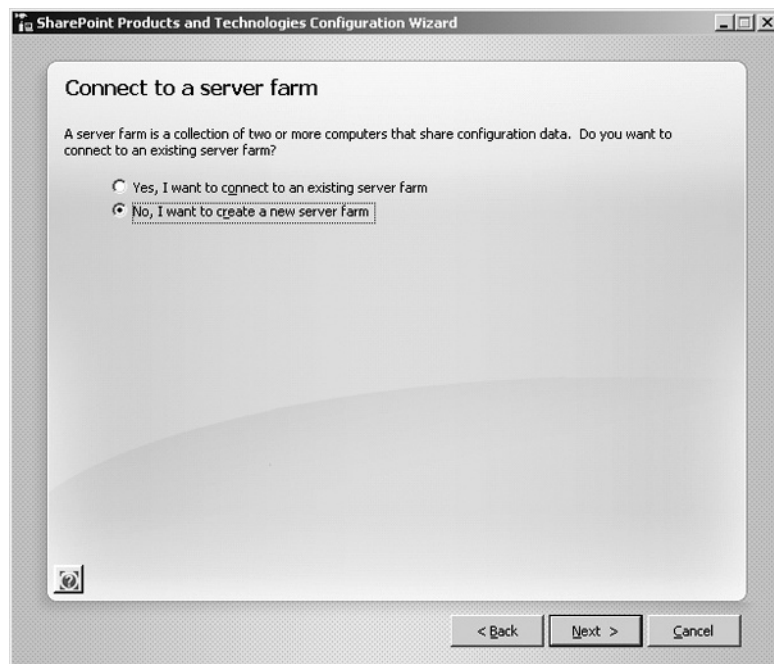
Advanced *Stand-alone.* This installation is essentially the same as the Basic install. Use this installation method if you intend to install SharePoint on one server only, and you want SharePoint to install and use the SSEE database. The only difference between this install type and Basic is that it gives you the option to specify the location of your index files, as well as define your feedback (because you may want to let Microsoft know about your day to day SharePoint experience).

Web Front End (Server Farm). This installation method actually includes a few kinds of SharePoint topologies. At its simplest, this is the method of installation you use if you don't want SharePoint to install SSEE because you have, and are going to use a SQL Server. That's because Basic and Stand-Alone install SSEE without your involvement. If you have a SQL server on your Windows network (2000 SP3 or 2005) and you want to use it to house your SharePoint databases, then the Server Farm install is the only type that lets you specify where your databases will go.

The other reason you would use the Server Farm installation method would be if you want a server farm topology. A SharePoint server farm uses more than one server to support SharePoint. This can be simply one SharePoint server and one SQL server; or it can be scale up to a more complex topology, such as several SharePoint servers (generally called web front end servers) and an SQL database cluster. The simplest server farm consists of a database server and a server with SharePoint installed on it, so the two functions are separated between two servers. Together they are a server farm. Of course, there is more to it than that. Usually, people create bigger server farms which means more SharePoint servers all using the same SQL databases. This is appropriate if they have a lot of SharePoint sites and they want to spread HTTP requests between servers to improve performance; meaning multiple SharePoint servers, and even multiple, clustered database servers.

If you choose to do a Server Farm installation, you can specify whether the SharePoint server you are installing is the first on the farm or if you want that server to be part of an existing server farm (see Figure 1.1). The first SharePoint server on a server farm is kind of like the first domain controller in a domain. Because it's the first, it tends to hold all the services and is the one used to set up the databases. Choosing to add the server to an existing server farm means that the installation will install only the files needed to make that new server a web front end server to help support the first server with client requests.

FIGURE 1.1
Starting a new server farm, or connecting to an existing farm



Server farms work in this configuration because the databases that hold all the information of the SharePoint sites already exist on the SQL server. All you have to do at that point is specify which configuration databases the new server will share with the first server, and presto change-o, you've got a new SharePoint server with the same configuration and content.

THE GORILLA IN THE ROOM

Something that isn't mentioned much is that server farms, in addition to having front-end servers that all access the same databases, are usually configured using Windows Network Load Balancing software, DNS round robin, or a hardware load-balancing device. Real, server farm, load-balancing functionality requires additional setup using something other than SharePoint. Installing additional SharePoint front-end servers is only one part of it.

To make matters worse, there is little current documentation about how to do load balancing. So check out Chapter 15 for a brief demonstration of how to simply do network load balancing with SharePoint.

The differences between the kinds of SharePoint installations are not the stuff of rocket science. However, if you intend to do more than run everything on one server, or if you don't want to end up with the SSEE database, you really need to understand those differences before you install SharePoint.

SharePoint Sites and Databases

SharePoint needs at least two different IIS Web Sites (otherwise known as SharePoint web applications) to function. These web applications contain the web pages that you will access to either administer SharePoint or actually use SharePoint's lists, sites, and libraries.

The Central Administration Web Application This web application controls the configuration and administration of all servers on the server farm, as well as all web applications. This site is set up on a completely different and unique port than the standard one for HTTP. If you do a server farm installation, you can specify the port or use the one suggested. If you do a Basic or Stand-Alone installation, then the port will be chosen at random for you during installation and configuration. The range is somewhere between 1023 and 32767. The unique port helps obscure this site from anyone surfing the standard ports on the server.

The SharePoint Site The default name for the first SharePoint web application (that isn't dedicated to Central Administration) is usually SharePoint-80. It will contain the first top-level site for SharePoint, just to get you started (or in a server farm installation, you will have to create if yourself). Web applications were meant to contain site collections, which are literally collections of sites, starting with a top-level site, but can also include additional subsites. Web applications can contain as few as one site collection with one top-level site, or many site collections, each with multiple subsites. Because a web application is essentially a container for your SharePoint sites, when you configure settings at the web application level, they can affect *all* sites contained therein.

I THOUGHT THERE WOULD BE ONLY TWO

Keep in mind that these are the web applications that are created during SharePoint installation. You can create more if you'd like. If you inherit a SharePoint server and find that more than two web applications are being used by SharePoint, that's fine. Someone probably added more for a good reason (see Chapter 8 for more information as to how and why to create additional web applications), and now you are responsible for them. Congratulations.

Each SharePoint web application needs at least one content database to contain its data. The Central Administration web application also accesses the server farm's configuration database (which stands to reason, because that is where all the configuration settings are for SharePoint). Because SharePoint is capable of performing full-text, site-collection wide searches, Search also has its own database.

The Sharepoint Databases This means that four databases must be created when SharePoint is installed. I am using the default names, but you can change them depending on how you install SharePoint.

SharePoint_Config_(GUID): This is the configuration database for SharePoint. It holds all of the configuration data and settings for the entire server farm. The thing that makes separate SharePoint servers all members of the same server farm is that each of them use the same configuration database. This makes it possible for all of those servers to have the same configuration settings. When you do a single server installation, the database will be called SharePoint_Config_(GUID). If you do a server farm installation, the suggested default (which you can easily change) is simply SharePoint_Config.

WSS_Search_Servername This is the database that contains all of the search data acquired when the index (or content access) service crawled through the SharePoint site collection. Search is an interesting beast in SharePoint, both overly simple and potentially complex.

WSS_Content This is the content database for the first web application made in SharePoint for SharePoint sites. It will contain information about the site collections that the web application contains, and it will contain all of the list, library, and web part metadata, documents, and attachments. Keep in mind that you can have more than one content database for a web application, and chances are good that you will grow out of the first one pretty quickly.

SharePoint_AdminContent_(GUID): This is the content database for the Central Administration web application. Because the Central Administration website is just like any other SharePoint website, it is prone to the same strengths and weaknesses. Site settings can be changed, including those for the master page. Novices should not do this. As a matter of fact, no one should. They could potentially delete the document library folder containing the Help files and more.

Content Databases

Although each web application gets its own content database, web applications can contain more than one site collection, and each site collection can contain multiple sites that can contain lists and libraries that *can* get really big (I'm not guaranteeing anything, I'm just saying, over and over, that they *can*). Frankly, using a single database to contain large sites full of data can be an invitation for that database to become really slow and unwieldy. There is always a reasonable limit to how much any one database can hold, and its surprising how quickly that limit can be reached. Don't think of it as a bad thing; it just means that people are using the sites. To help you cope, SharePoint allows you to add extra content databases to web applications to keep up with the ever-increasing data load. This is why it is possible to have several content databases for one web application. In addition, you can configure database capacity settings (by limiting the number of site collections per database and the size in MB of the site collections themselves), so that you can be warned when a database is getting too big and be prepared to add a new database.

Overall, this means that SharePoint uses IIS Web Sites as web applications to hold site collections. Web applications can contain multiple site collections, each site collection can hold

many sites, and each site can have many lists and libraries. As a result, a SharePoint server farm can have many web applications, each with several content databases. However, there can only be *one* configuration database for each server farm. The configuration database specifies the configuration for the whole farm and, therefore, must be only one.

SharePoint Service Accounts and Services

After it installs, SharePoint creates and enables certain services and application pools. To be able to do their jobs, these services need to run with some sort of account context. Depending on how you install SharePoint, you may have to create domain accounts to apply to those services. If you want SharePoint to work, it will help to know what the services are, what they do, and what access those accounts need while remaining secure.

Service Accounts

Here are the accounts you need for SharePoint to install and work:

Setup Account (Basic or Stand-alone install) In order to install SharePoint, you must be logged in on the server with an administrative account. If your server is not in a domain, this account needs to be the local Administrator. In order to install SharePoint, you must be logged in on the server with an administrative account. On a domain, the account can be a domain admin. The account must be able to install software locally, and should also be allowed to add and start services on the server.

All other service accounts used by SharePoint are set up automatically (local system or network service) with a Stand-alone or Basic installation. It really is the easiest installation, in addition to being the cheapest. Although it is not super scalable, it is convenient.

THE CHEESE STANDS ALONE

You don't have to install SharePoint to support multiple servers in a domain environment. You also can install SharePoint on a stand-alone server in a workgroup with no domain controller. The easiest way to do this is to install SQL server (or let SharePoint install the SQL 2005 Embedded Edition for you) on the server that will house SharePoint. Then it can do all the database management it requires without needing to access anything on a different server. To use incoming email features, the server will also need to have SMTP enabled. Local users and groups will be used to give users access to SharePoint in that scenario, rather than going through a domain controller. It just goes to show that SharePoint is scalable down as well as up.

Another bonus of a single-server install with SQL 2005 Embedded Edition is that you don't really need to worry about specifying domain permissions or specific SQL permissions of the SharePoint service accounts. If you choose a Basic install, database and services set up will be done for you by SharePoint using the administrative account you used to log in. It will specify that all services will run using local system or network service server accounts.

Setup Account (Server Farm) In a domain Server Farm install, the setup account should be a domain admin (you can use local Administrator accounts to install SharePoint on each individual server, but it is easier simply to use one setup account that is a domain admin). This account should be allowed to install SharePoint on any server in the domain, and it must be able to access the SQL server that SharePoint will be using to build databases.

On the SQL server, the setup account must have these SQL server security roles on the target SQL server: Logins, Securityadmin, and Dbcreator.

Database Access Account Also known as the *server farm account* or *configuration database account*, this account is powerful and critical to SharePoint. It does not need to have administrative privileges; but it should be a domain account. All other rights for this account will be configured automatically by the setup account during installation. The setup account adds the database access account to the SQL server's Logins, Dbcreator, Securityadmin roles. This is why the database access account ends up being the owner (DBO) of most of the SharePoint databases.

THE DBO EXCEPTION

Oddly enough the database access account does not become the DBO of the configuration database for the server farm because the setup account creates that database during installation and then assigns ownership of it to the database access account. This means that, by default, the setup account is the DBO, but the database access account holds an owner role.

This account is the Central Administration application pool identity. This means that it is *the* account that accesses and changes the configuration database for the server farm. It is also the account used to power the SharePoint Timer Service, which is in charge of any jobs that need to be started and stopped at different times (such as getting incoming mail, managing quotas, and alerts). This account should be guarded and not used for anything else.

Content Database Account Also known as the *web application account*, or *web application application pool account*, this is the account that uses the content database of a web application. There should be one of these per web application—although under some circumstances (as is the case in businesses with security policies that limit service accounts), web applications can share an account. This account should be a domain user and otherwise is given (and requires) database ownership of all content databases associated with a web application.

Search Account This account should be a domain user. It directly accesses the Search database. Because it takes the questions entered into the Search field in SharePoint and queries the Search database records with them, it is considered the query account.

Content Access Account Also known as the *index, gatherer, or crawler account*, this account analyzes all of the content in SharePoint site collections. It must be a domain user, and it will automatically be given full read rights to all web applications. It also has access to the Search database to write in the information it has gathered.

Optional SharePoint Admin Account I also suggest you consider a general purpose SharePoint administrator account. This account should be a domain admin (or at least local admin

for each SharePoint server), so it can install tools locally on all SharePoint servers on the farm, run the SharePoint command line tools, and can be used as a default administrator for central administration and new site collections you may create. It comes in handy for me when I need to troubleshoot a site or a setting in Central administration. I always know that account's name and password, and it is usually the first owner of most site collections I create (of course, this may not be allowed to remain after handing the collection over to its rightful owner, but it's convenient during setup).

IF THE SQL DBA DOESN'T PLAY NICE

If the person in charge of the SQL databases is not comfortable giving the SharePoint server admin the power to create databases on their SQL server, don't worry. If you prebuild the databases that the SharePoint server will require, SharePoint will happily connect at the correct points of setup to the preexisting database without unduly empowering non-database administrators to create databases of their own. Check out the "Deploy using DBA-created databases" TechNet article for more information about how to configure the databases and handle service accounts in that situation. There are versions of the document for both WSS and for MOSS, but they are basically identical. Go to the Windows SharePoint Services 3.0 Technical Library, Deployment for Windows SharePoint Services 3.0 technology, End to end deployment scenarios, "Deploy using DBA-created databases."

SharePoint Services

The following services are created and required by SharePoint. It might be handy to know what they are before you conduct your first installation.

SPAdmin (Windows SharePoint Services Administration) This is the administrative service for SharePoint. It runs on every SharePoint server locally and is in charge of checking the configuration database for changes. It keeps track of what server on a server farm is running what service, and is used by sharepoint to access local resources per server. This services runs as the WSSADMIN process in Task Manager.

SPTimerV3 (Windows SharePoint Services Timer) This is the service in charge of actually triggering and running jobs for SharePoint. Because it uses the database access account identity, it usually doesn't have administrative permissions on the local server; however, it does have ownership permissions to do what it needs to do on both the configuration and content databases. If it needs to do something administrative on the local machine, it calls on the SPAdmin account to do it. This service runs as the OWSTIMER process in Task Manager.

SPSearch (Windows SharePoint Services Search) This is the Search service for SharePoint. It runs on the SharePoint servers that are running the Search service. This service runs the mssearch process in Task Manager.

SPTrace (Windows SharePoint Services Tracing) This service also installs on each SharePoint server locally. It is used for error tracking and analysis, and controls the trace logs. This service runs as the wsstracing process in Task Manager.

SPWriter (Windows SharePoint Services VSS Writer) This service integrates with SQL's VSS writer service, inherited from SPS 2003, and works with SharePoint's backup and recovery capabilities. It makes it possible to use Windows Volume Shadow Copy when doing backups. This service runs as the SPWRITER process in Task Manager, and only starts when necessary. So it's not always running.

SQL SERVICES

SharePoint is dependent upon SQL, so it should go without saying that if it is installed to use the SSEE database locally, that version of SQL should be running locally as well.

User Account Modes

Most people don't even realize they have a choice when it comes to selecting a user account mode. By default, SharePoint will install using the Active Directory Domain Account mode. However, hidden deep behind an Advanced Settings button at the end of SharePoint's installation is the option to choose a different option when it comes to how SharePoint handles user accounts. When they say "advanced," they mean it. Setting up account modes is a one-shot deal. You get one chance to choose your user account mode when you install the first SharePoint server in the server farm (or a Stand-alone server). Then that information gets locked in the configuration database of the whole farm, affecting the whole farm with no way to change it. *You cannot undo the account mode decision once you make it.* So choose with caution. There are two choices and they both are based on Windows Active Directory user accounts.

There are two User Account Mode options. The default user account mode is the one with which we are all familiar—Active Directory Domain Account Mode. The other account mode (Active Directory Account Creation Mode) is more complicated to set up and is actually considered a different SharePoint deployment. That other user account mode is the one that must be selected during installation in order to be enabled—otherwise the default, Domain Account mode will be enforced.

Domain Account Mode This mode is selected by default during SharePoint setup, SharePoint lets the administrators add users to SharePoint based on their Active Directory or Local Users accounts. In other words, first you have user accounts in Active Directory (which is what you probably have already), and then you can add them to SharePoint.

Active Directory Account Creation (ADAC) Mode When this mode is selected during SharePoint setup, SharePoint allows administrators to create user accounts in Active Directory when they add them to SharePoint. That's right. When you add users to SharePoint, it adds the users to a special Organizational Unit (OU) in Active Directory. This was meant for ISPs or companies that had a lot of external partners or offsite users who needed authenticated access to the server content. Therefore, instead of creating user accounts in Active Directory and then adding them to SharePoint, Active Directory Account Creation mode does it the other way around by adding the user to SharePoint, which adds them automatically to Active Directory. See Chapter 15 for more information as to how to use ADAC and what happens when you do.

Sounds interesting, doesn't it? But keep in mind that it is a potentially complicated procedure from which there is no going back. See Chapter 15 for a step by step look at Active Directory Account Creation mode.

Authentication Types

In conjunction with IIS, SharePoint supports several different ways to allow users to authenticate. They are not exclusive; you can choose to apply multiple types of authentication to a web application. IIS will apply the most restrictive method first. If that fails, it will try the second most-restrictive method, and so on until it finally refuses the client or lets them log in.

A Rose by Any Other Name . . .

You may have noticed that SharePoint uses the same terminology in several different ways in several different places. Here is a quick rundown on some of them:

Authentication Provider: (sometimes referred to as Membership Providers): Usually refer to the services that provide authentication like SQL Forms based authentication or Windows authentication.

Authentication Method: The method that authentication is sent to the Windows Authentication Provider, such as NTLM or Kerberos. In IIS, authentication methods also refer to using other authentication types like basic, digest, and integrated Windows to authenticate users to a particular Web Site.

Authentication Source: Where the authentication accounts are stored and accessed by the Provider. Examples of this are Active Directory or the database used by Forms based authentication.

Because SharePoint has gone through several different versions by this time, and over time it has changed its terminology. However, the SharePoint command line tool, STSADM, does occasionally reflect some of the older terms. So don't be surprised if you see, when working at the command line, the following examples:

- ◆ Sites collections are called "Sites."
- ◆ Sites are often called "Webs."
- ◆ Subsites are referred to as "Subwebs."
- ◆ Server Farms are often referred to as Web Farms.
- ◆ Web Applications are called "Virtual Servers."

Windows Integrated Authentication This authentication method requires the user to have a domain account or a local account on the SharePoint server. This, of course, is the method that Microsoft prefers and the one used throughout this book.

Digest This also works with Active Directory, but it sends the username and password as hashed values. It can be used if Windows Integrated Authentication is blocked by a firewall or not being passed by a proxy server. It is also available on WebDAV servers.

Basic This method will send authentication information across a network as cleartext, which is obviously not a great idea.

Anonymous Access This method allows users to establish an anonymous connection with IIS by using an anonymous or guest account.

Authentication Methods

In addition to those authentication types, SharePoint offers two Windows authentication methods during installation. These protocols don't just govern how authentication data is passed on the network for users trying to access SharePoint; they govern how SharePoint service accounts themselves access resources:

NTLM This secure protocol encrypts usernames and passwords over the network. It simply sends data to the authenticating authority and back. This protocol does not require additional configuration, and it is suggested for most SharePoint scenarios.

Kerberos This secure protocol encrypts data but handles authentication differently than NTLM. Kerberos is based on *ticketing*. A username and password are passed to an authentication server, which sends back a ticket to allow the authenticated user to access network resources. The user *and* the authentication server (or Key Distribution Center) must trust each other. This means that service principal names must be set for the SharePoint servers and the database access account so resources on the network can be accessed by SharePoint on behalf of the user. The account and the servers must be trusted for delegation in some circumstances.

Microsoft suggests using NTLM, because using Kerberos requires the database access account to have a service principal name, which could be a greater danger to the network if that account is compromised. And even though outside the network, authentication is tighter with the mutual authentication process of Kerberos, using to authenticate can be a problem due to time synchronization. There is one catch though: in some situations, search's index service cannot authenticate using Kerberos and therefore cannot index sites that require it. For more information about Kerberos and how to configure it, see Chapter 15, "Advanced Installation and Configuration," for more details.

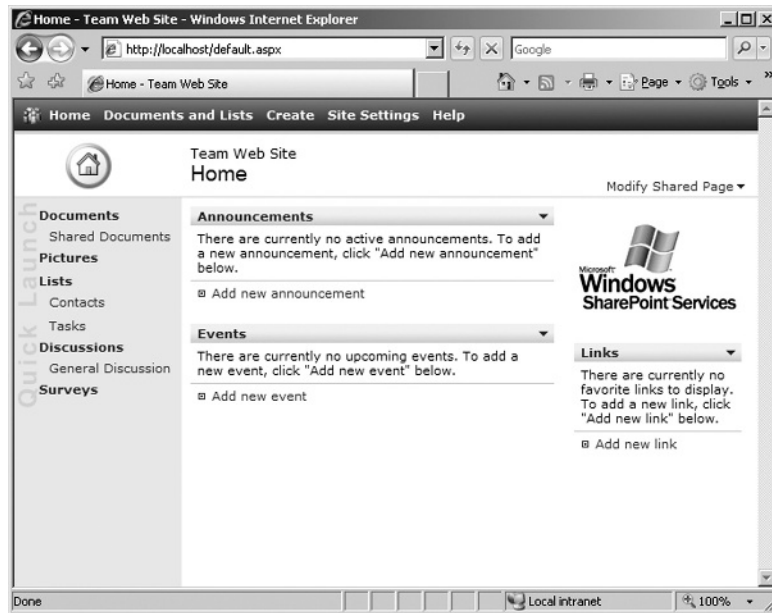
IFILTERS

Don't despair if you are thinking about collaborating on files other than those made in Microsoft Office using SharePoint. Some vendors use Index Filters (IFilters) so that Search can index their document types. Check with your vendor first to see if they have an IFilter you can use to recognize their file types for searching.

SharePoint Search

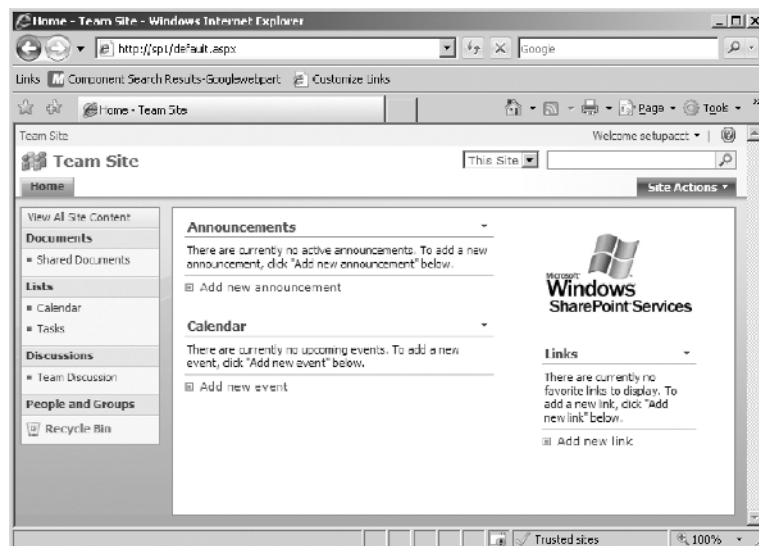
The Search feature is new to Windows SharePoint Services. In the old days, you could enable Search in SharePoint (WSS 2 and lower) only if you were using a full-blown SQL server. This was because SharePoint was simply using SQL's built-in full-text search and indexing features to do searches, which could add a significant performance load to the SQL server. This meant that if you did the Typical install of SharePoint, which is now called Basic, and used WMSDE for the database, the Search field simply would not be available in SharePoint (see Figure 1.2). WMSDE does not have the search capabilities of SQL server.

FIGURE 1.2
In WSS 2.0, a Typical
install could not search



WSS 3.0 uses the same SharePoint search service that the older version of MOSS used. This means that you can perform full-text searches of site collections, including Office files, *even* if you did a Basic install of SharePoint and are using the SSEE database (see Figure 1.3). SharePoint is doing the searching now, not SQL. This helps lower the performance stress of the SQL server as well.

FIGURE 1.3
In WSS 3.0, a Basic
Installation has a search
field



Search basically does two things:

- ◆ It responds to search queries.
- ◆ It crawls through site collections and indexes data.

This is why Search has two services, the search service and the index service (or content access service), and their corresponding service accounts. Both services use a Search database; the index service merges its collected data with it, and the search service queries it. Only one index service can exist on a server farm, but there can be more than one server running the search service on a farm. (Each server would share the index service.) The index service requires read access to all content databases of all the web applications that will be searched. When a web application is being created, you can assign a search server to service its content database. This is useful if you have more than one server running search.

The index service will scan the content databases of the web applications per the schedule you set up when you enable Search. The changes that it finds, are temporarily stored in index files on the SharePoint server that is running the index service, then merges them with the Search database after a set period of time. Meanwhile, the search service, when responding to a user query, will check the index files and the database to be sure that all results are accurate. This is why there can be only one server running the index service on a farm, because those files have to be in one place.

Search has some strengths and weakness that you should know about before you install SharePoint:

- ◆ Search only returns search queries per site collection. That means if you are looking for a document and you have several site collections, you need to know what site collection it's in or search each site collection until you find it. Site collections are a hard-search boundary.
- ◆ Search doesn't have much of an administrative surface. The GUI settings are limited to what service accounts use, the Search database name, and how often the site collections will be indexed. Indexing is primarily incremental, but even that can strain resources if you do it too often. What little management you can do with search is through the SharePoint command line tool STSADM. See Chapter 13, "STSADM: A Look at the SharePoint Command Line Tool" for more details.
- ◆ Search can search *only* site collections (or more precisely content databases). It cannot search file shares, email servers, or other locations. If you want to search content outside site collections, consider shelling out the money for either MOSS or MOSS for Search (which for the added cost, can search multiple site collection or even multiple SharePoint servers).

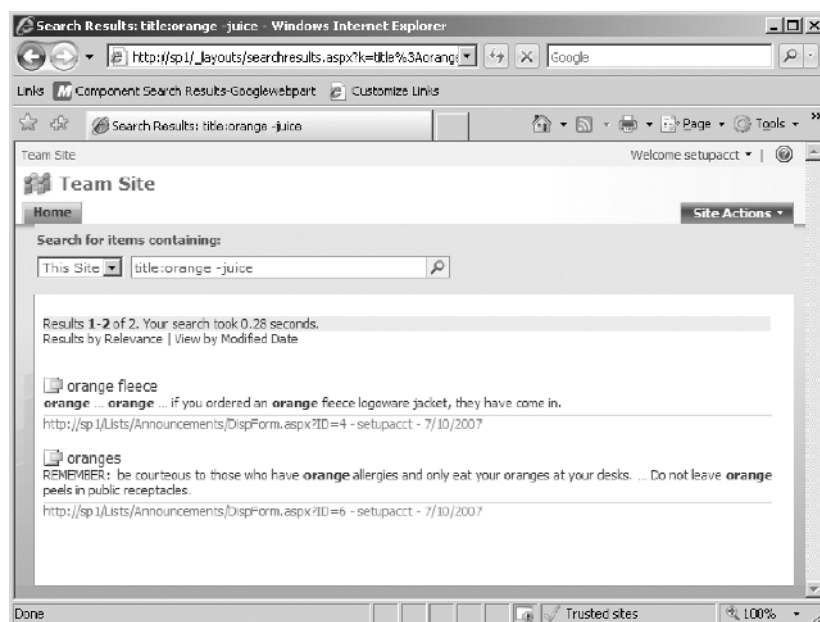
Search uses a top-down approach. When you conduct a search query on a site, it will search that site and all subsites under it. If you conduct a search query on a site at the top of a site collection (the first site created in a site collection), it will search the data contained in its Search database and index files for that site and then systematically check all other subsites below it. However, if you are already on a subsite and start to search, it will search from there and work its way down the subsites below it, ignoring the sites above it in the collection. In other words, Search always searches *down* and never *up*. Unless you absolutely know which subsite has the data you are looking for, you should always perform searches from the top level of a site collection.

- ◆ Search does whole word, exact match queries. If there are multiple words in a query, AND is implied between the words (orange juice is considered orange AND juice, and would return only results that contain both values). Punctuation is ignored, as is the word "and itself."

However, strangely, the word OR is neither ignored nor recognized as a part of the query logic and is treated like part of the query text itself.

- ◆ Unfortunately, Search doesn't accept wildcards or Boolean logic, but it does allow for keyword exclusions or additions by using the plus (+) or minus (-) signs. Search will also support property filtering. Property filtering means that search can recognize some field names and properties, such as filetype, contenttype (used for libraries particularly), author, title, or subject. To filter in the search field by property, the syntax is property:query, such as filetype:txt will result in all text files in the site collection.
- ◆ Searches can be scoped. This is a simple concept that just means that when you are in a list, library, or folder, the little dropdown list next to the Search field offers you the option to search that one location or the entire site.
- ◆ The search results are displayed on a page organized by modified date or relevance (the default is relevance). This can further allow you to narrow down the search query. Results are displayed with the link to it, and some summarizing information. The page even displays the length of time the query took to complete (Figure 1.4).

FIGURE 1.4
Search results page



- ◆ In a server farm, there should be only one search service account and one index (content access) service account. However, if you have a large and busy server farm, it might be good to have a server dedicated to searching, or you could run more than one server with the search service enabled. Search prefers Windows Authentication and may cause errors in an anonymous environment. In addition, the index service prefers NTLM authentication, so it can have problems accessing a web application that requires Kerberos.

- ◆ Sites and lists can be excluded from indexing, if you'd like them to be unsearchable such as lists with item level security, which may cause some items to be displayed for those who can't open them.
- ◆ Search can perform *security trimming*, which means it includes security information when it is indexing site collections and excludes items from a query based on the permissions of the person asking.

INDEXING AND GATHERING

The search service's index service appear to be hand-me-downs from SPS 2003 and MOSS. This is why WSS 3.0's Search is independent from SQL. The index service is a powerful feature that you don't need to monitor. It takes care of itself and does its own thing with WSS. (MOSS has added configuration features for indexing.) Its only content sources are the content databases that SharePoint uses. It uses IFilters and protocol handlers to parse documents, filter out formatting, and find words in documents. It can distinguish between relevant words and irrelevant words or "noise." It can handle only 64MB of indexed words. If it maxes out, it doesn't really notify you; it just doesn't index any more of the document, which is another reason to keep uploads and document files from becoming bloated. It allows whole word searches and keyword inclusion/exclusion, but it doesn't support stemming, wildcards, or Boolean logic.

The IFilters that come with SharePoint can handle Office 2007 file types, text files, HTML, and TIFF files (which is the file type usually used for scanning faxes and documents).

SharePoint and Email

Sharepoint integrates with email more than ever. This is why you should consider how you'll configure email when you're planning to install SharePoint.

In addition to being capable of sending alerts and notifications (which requires properly configured outgoing email), SharePoint needs to be set up to receive incoming email. This is because several lists and libraries can be enabled to receive email. The primary benefit is that you can send a new item to the list without going to the SharePoint site if you know that list's email address. And you can do of this from the comfort of your email program. No need to open a browser.

To manage incoming email, the SharePoint server needs to have the SMTP service set up locally. You should have that enabled before you install SharePoint. When SharePoint receives email, it pulls it from the default drop directory that SMTP uses or from the directory you specify. It gives it to the correct list or library, which parses the email for the subject line, message body, and other pertinent header information. It then applies the information to the appropriate fields in the list record.

Incoming email has another interesting new feature called SharePoint Directory Management Service. This service integrates SharePoint with Active Directory. To use it, you need to create a unique OU, give the server farm account extensive access to it, and assign the content database accounts local administrative rights to the SharePoint's server. SharePoint can allow users to create distribution lists that show up in the OU and add the list and library incoming email aliases to the Exchange global address list (GAL). Of course, this obviously requires Exchange, and more specifically Exchange 2003 because it integrates so deeply with Active Directory.

DON'T LET THEM SCARE YOU

Despite occasional documents stating otherwise, SharePoint Directory Management Service does not have to be running for SharePoint to be able to do incoming email. In its simple straightforward way, incoming email works fine without it. If you don't want to increase the complexity of your SharePoint install, don't use SharePoint Directory Management Service. It is an option, not a requirement. Its biggest strength is that it allows Exchange users to have the SharePoint lists' addresses show up in the global address book. It simply adds contact records and distribution lists to the specified OU in Active Directory. For more about Directory Management Service, see Chapter 15.

Alternate Access Mapping

When you initially install and start using SharePoint, accessing it by using the NetBIOS name of the server works fine, but what if you want to be able to access it from the internet? You can't resolve that server name among all the other machine names on the internet, so you need it to resolve to a DNS name. Alternate Access Mapping is about mapping a SharePoint web application to an alternate name other than the default. That means that you can have an internal, default name of `http://sp2` and a different, internet URL of `http://sharepoint.dem0tek.com`, and both actually point to the same server.

Alternate Access Mapping (AAM), specifies alternate access of a web application by internal URLs, public URLs, and zones. An Internal URL is what the web application responds to. A Public URL is what the web application returns to the user in the address bar, and in the links for all search results. Web applications can have five public URLs associated with it (at which point they are called *zones*). So you can have a Default zone (that's the default URL for the web application which is usually the root path for all the site collections it might contain), an Intranet zone, Internet zone, Extranet zone, and a Custom zone.

There is also another use for AAM, extending web applications. Sometimes you might want to have two web applications using the same content database (and therefore accessing the same websites, lists, etc.). This can give users two ways to access the same data and is useful if you want to have two different types of authentication for the content, depending on what URL the user uses. Because the extended web application is just sharing the same content database as an existing web application, it is considered just another URL used to access the first web application's content. This is why an extended web application is not given its own name but is considered a *zone* of the existing web application. In that case one of the Public URL zones is taken up with the URL of the extended web application.

So when planning your URL structure and how users are going to access SharePoint, keep AAM in mind.

Managed Paths

When planning for SharePoint it's a good idea to keep in mind how you would like to structure your site collections. Site collections are composed of a top-level site and all the sites that stem from it (called subsites). The top-level site is usually accessed by using the web application's URL and then the path to the top-level site's home page. When creating a site collection, you must decide what its URL path will be. When you create your first site collection in a web application, you can give it the *root* address for that web application, or you can specify a path. What this

means is if you create the first web application on server SP2, then its URL can be `http://SP2`, using port 80, which is the root address for the URL. But if you create a second site collection in that web application, it needs to have a different path, because it can't use the same URL. This is where Managed Paths comes in. By default SharePoint has a "sites" managed path for additional site collections. The URL for that path would be, on the same server, `http://sp2/sites/`. What this means is if you create that additional site collection, it can be something on that path, such as `http://sp2/sites/something`.

You can, of course, create your own, depending on your required topology. This is useful if you are planning to have one web application, say, per region, and then site collections for each office. Then you might consider creating a managed path for the London office, Beijing office, Helsinki, etc.

Site collections are useful for being a user account or permissions boundary because you can add users once to the top-level site, apply their permissions, and they are available as users in all subsites as well; but for that site collection only. The other site collections are unaffected by the comings and goings of users in any other site collection.

Another thing to consider with managed paths is that if you have additional non-SharePoint Web Sites or web software you want to run in the same IIS Web Site virtual directory, SharePoint automatically ignores it if it is on a path not specified in Managed Paths.

User Accounts and Permissions

In order for anyone to use SharePoint, there must be users. SharePoint leans toward organizing users and permissions based on the users' roles. So if you have a site owner, he would need to have full control of the site, but a member would only need to be able to be a contributor.

SharePoint controls the user permissions that can be applied at the web application level. So if necessary, you could actually block certain permissions entirely from ever being applied to users in the site collections the web application contained. At the site and site collection level, permissions can be combined to create permission levels, which are then applied to users or groups.

Individual Active Directory Users can be added to SharePoint, but you can also simply add domain security groups as well. Doing so lets you add a number of users to SharePoint that might require the same permission levels, at one time. It is also easier for SharePoint to handle because has limitations on how many separate security principals it can manage at one time. It's actually considered SharePoint best practice to use AD security groups to add users rather than individual domain users for that reason.

SharePoint uses SharePoint Groups to organize users. There are three SharePoint groups built in: Members, Visitors, and Owners; but you can also make your own. When you create a SharePoint group, you assign permission levels to the group. Then, when you add a user, you choose the SharePoint group they should belong to, and that group's permission levels automatically apply to that user. So when planning your user management strategy keep permissions, permission levels, and SharePoint groups in mind.

Hardware Requirements

Trying to pin down the exact hardware requirements for a product like SharePoint is tough. There are many different ways to use it; therefore, there are many ways to configure the resources.

Microsoft has some suggestions for SharePoint server's recommended and minimum requirements. These recommendations are for average server loads. In my experience, the recommendations work pretty well as long as your network is healthy and well configured.

Processor 2.5GHz minimum, dual processors, 3GHz recommended.

RAM 1GB minimum, 2GB or more recommended.

Disk 3GB, NTFS. More disk space is recommended, depending on your storage needs. The 3GB reflects only what SharePoint needs and does not include the needs of the operating system, any SQL databases (if you are going to do a single server install), or anything else running on the server.

ALL IN ONE

Disk space is a particular issue if you are running SQL and SharePoint on the same server, as they would be in a Stand-alone installation. You will need to plan for the storage space of the SharePoint pages in IIS, SMTP mail storage (if you enable incoming email), the indexing files used for search, all the storage space that your site's lists and libraries will use, and all the other databases SharePoint uses. As you can see, the space that SharePoint might need for its files is not the only space you'll need. In this case, everything is stored in one place. Size it well and guard it carefully.

DVD Drive Not really required for SharePoint, but useful.

Display 1024 × 768 on the client (800 × 600 is too small. It forces some pages to require way too much scrolling.)

Network 56 kilobits per second (Microsoft's minimum), 1 gigabit per second is suggested.

These recommendations are just starting points; however, they are more than adequate for most simple SharePoint Server Farm installations. Most single server or simple Server Farm installations can probably handle 1,000 people creating an average load on the SharePoint server, without seeing a lag in operations per second. Commonly, each gigahertz of processing power in a SharePoint server can handle about nine operations per second.

Performance Planning

You might be wondering how you determine operations per second? There are formulas to help you figure that out.

Essentially, you need to know:

1. How many people are supposed to use SharePoint? (Users)
2. What percentage are *really* going to use it? (Percent active users)
3. How many operations per day they do on average (how many documents edited, list entries added, searches done, etc)? (Operations)
4. How many hours do the users work in SharePoint on average? (Work Hours)
5. Whether an average work day has particular peaks in performance? (Peak Factor)

To calculate the operations per second, multiply items 1, 2, 3, and 5 together, then divide that number by the number of hours those people are going to be working a day by 360,000 (which is 100 percent conversion × 60 minutes per hour × 60 seconds per minute). Altogether that will show you how many operations per second your server needs to efficiently handle.

To show you what I mean and illustrate that the above hardware requirements are probably adequate for your needs, assume your office has 1,000 people who are going to use SharePoint and 60 percent of them will be actively using SharePoint daily. You estimate that each user probably performs about 50 operations a day. (Most of them will spend more time editing a document than retrieving it from the document library or uploading it.) Let's say your office has, at maximum, 9 hours of work time a day and a peak factor of 4. *Peak factor* is a scale between 1 and 5 that refers to how often or how likely there are to be peaks in normal daily usage. One indicates that there is practically no particular time of peak usage during a business day, and 5 indicates that practically the entire day is a peak use time. I never go less than 4, just in case.

MEMBERSHIP IN CLUB SHAREPOINT IS NOT ALL-INCLUSIVE

Many businesses do not need to allow every employee access to SharePoint. Therefore, when you determine who will use the SharePoint sites, don't just include everyone in the company. To help ensure that your calculations are as accurate as possible, consider exactly who will do what.

Let's summarize the data we have:

Users: 1,000

Percent active usage: 60

Operations: 50 (per person, per day)

Work hours: 9

Peak factor: 4

And the formula that uses that information is:

$$\text{Users} \times \text{Usage} \times \text{Operations} \times \text{Peak} \div (360,000 \times \text{WorkHours}), \text{ or in our case,}$$
$$1,000 \times 60 \times 50 \times 4 \div (360,000 \times 9)$$

That will bring you to the operations per second that your server needs to deliver for your users. In this case, that number is 3.7 operations per second (OPS).

Given the standard formula above, 2.5GHz and 1GB of RAM should be able to handle at least 10 operations per second (Microsoft has mentioned that it feels confident that that hardware can handle 18 operations per second). All *you* need is 3.7 operations per second for 1,000 people doing 50 operations a day. You can see why I think the starting hardware requirements are sufficient for most small to medium businesses.

Of course, I don't really trust Microsoft's ideal that a 1GHz server can handle 9 operations per second. However, under normal circumstances, I could comfortably see at least 10 operations per second being safely handled by the 2.5GHz starting specifications given—especially with 1GB or more of RAM. Remember, just like the processor, RAM is important, if only so the server can render pages efficiently. Keep in mind that each web application a server hosts does increase the amount of RAM the server uses. More web applications, means more RAM.

Be cautious though. SharePoint often rapidly increases in use, and an increase in the percentage of people using it. As SharePoint catches on, you might find yourself at peak usage more often than not. That's why you need to monitor how your SharePoint server handles the stress of use, just in case.

PERFORMANCE MONITORING

You might assume that SharePoint has performance monitoring tools—but it doesn't. It doesn't need them. Windows Server 2003 already has a Performance Monitor for that sort of thing. SharePoint does have a Usage Analysis utility, but it only reports usage activity on sites and web pages. Performance Monitor is easy to use, well-documented, and should be a regular part of your server maintenance arsenal.

As an added bonus, there are performance counters specific to SharePoint that can be used in conjunction with the usage analysis data to manage your SharePoint server. They give you all the more reason to use Performance Monitor to monitor your SharePoint servers. See Chapter 12, "Maintenance and Monitoring," for more details.

ADDITIONAL PERFORMANCE CONSIDERATIONS

You'll want to keep an eye on these items that will increase your processor's load.

Alerts Users can set alerts on changes in a list or library. Alerts are scheduled and, therefore, keep the SharePoint Timer Services busy. Limit the number of alerts your users can have running at any given time. It will save your processor. Alerts can be configured with a user limit, or disabled altogether.

Indexing The server that will be indexing site collection content will have to support the increased load on the processor. If you can, try not to index every 5 minutes or less. Instead, consider indexing every hour or at certain times of the day, which would be better. This can be difficult if you expect SharePoint to almost instantaneously index and search new items; just keep it in mind if you are trying to squeeze as many operations per second as you can from your server.

Usage Analysis Sharepoint can analyze site usage, and deliver detailed reports. However, analyzing the usage logs takes a considerable amount of processor power for the SharePoint. Try to schedule the analysis to occur during a long downtime, usually sometime around 3:00 A.M.

Web Parts Your developers may go crazy with the power of web parts. Be careful; some web parts (depending on what they do and how they were coded) can be resource hogs. Stay well below 50 web parts per page—and that includes the hidden ones. Home pages, where web parts are usually found, can be overwhelmingly busy.

Storage Planning

When you're considering performance issues, don't forget to plan for adequate storage. If you plan to have SharePoint and the SSEE database on the same server, you'll need extra RAM because SQL uses quite a bit. But more specifically, it will require much more storage space than SharePoint alone. Even if your SharePoint databases—particularly the content database, which holds all of SharePoint's precious content—are stored on a different SQL server, planning for storage is still important.

Consider this, the maximum default size allowed for document uploads is 50MB. In my experience, a 100-page Word document is about 5MB. So, a maximum of 50MB is usually more than sufficient for a Word document. Of course, you can adjust the size; this is just a good default. But of course, if you upload more than Word files, you may need to change that limit.

It goes without saying that storage needs will depend on how your users will use the lists and libraries on your SharePoint sites. Assume they are creating marketing materials to send out every quarter, and they are storing them and collaborating on them in a document library. If they create five major documents each quarter, that would be 20 large documents per year, possibly up to 10MB per document. That could be 200MB of space for those documents alone. If other people manage the images in a picture library, and the material had 10 large, full color pictures per document, that could be 2,000MB (2GB) per year for that picture library in addition to its related document library. You could need gigs and gigs of hard drive space—and that doesn't include versioning.

If you have Versioning enabled in your document libraries, there will be multiple copies (as many copies as you allow when you set up versioning) of each document. Therefore, if versioning (say four major versions, and three minor versions per document) were enabled in the previous scenario, then at least 1.4GB per year would be needed for versioning in the marketing document library alone. Keep in mind that versioning can be allowed for most lists as well.

Most list entries, when stored in the content database, are tiny—just a few KB, if that. However, if you enable attachments for the libraries, those files (by default less than 50MB) will be saved with those list items, increasing the size of your content database in ways you may not have intended. And don't forget about incoming email. If you configure an incoming email enabled list or library to save original emails, those emails (including attachments) need to be stored in the content database too.

You also need to consider that, depending on what you allow, users can easily create their own document workspace subsites from a document if they need additional team work to collaborate. When a document workspace is spun off of a document, it takes a copy of the original document with it. An additional site will need to be stored in the content database, and a copy of that document with its own versions will be stored on that site. That document will eventually be *merged up* to the original document workspace. However, until then (and until you delete the document workspace when it's done) that document (and its workspace) is yet another thing requiring storage. You can also allow users to create their own site collections (with Self-Site Creation), this adds yet more storage overhead to the SharePoint content databases.

Finally, remember that the more stuff you have in SharePoint, the more stuff you will have in the Search database. It holds the indexed search data for documents, list entries, and page content (it *does not* index attached files); that data is stored on the SharePoint server itself and merged regularly into the Search database. To make sure that it returns only the entries that the user making the query is allowed to see, Search also records the Access Control List information for every indexed entry.

Generally Search is only allowed to store indexed word entries that equal about 40 percent of the original document's size, with a maximum of 64MB of stored words for a single document. That is well over the 50MB limit, but that's a maximum hard limit regardless. That means if you have 20 documents in a library, the search database can have (maximum) 1.3GB of entries in the Search database for that library alone. Of course, if the documents themselves are never over 50MB, and Search sticks to its 40-percent limit for each document, then that would be no more than 20MB of indexed entries per document, and therefore (going with our scenario) about 400MB stored in the Search database for that one document library.

When you're deciding how much storage space your SharePoint server should use in SQL, consider this:

- ◆ You need to have an idea of what your users are going to do. Estimate how many documents they are going to be collaborating on and storing. Think about what lists they will be using, and how they will be used.

- ◆ Plan how you are going to manage attachments and versioning.
- ◆ Plan how you are going to manage user websites—especially ones generated for document and meeting workspaces.
- ◆ Plan on using site collection quota templates to keep site collection storage in check (in addition to limiting site collections per content database). Remember the Recycle Bins as well. The End user Recycle Bin contents at the site level are part of the site collection's quota, so keep an eye on it. But the second stage, site collection level Recycle Bin can have a quota that is a certain percentage of its site collection's quota, but keep in mind that is in *addition* to the site collection's quota. That can cause an unexpected increase in storage requirements if you aren't prepared. Remember to empty your recycle bins to save space.

Once you can estimate what you need, double that space. At least, always have 25 percent more space than you expect to need. Always leave room to bloat. You will never go wrong.

It's great if SharePoint works, but if you have no more room to store SharePoint's data, the users will be upset.

Keep in mind that your environment may be different; after you install your SharePoint server, make sure you monitor the activity. Create a test group that represents a small but measurable sample of your expected users. See how many of them use the server, when they use it, how they use it, and how much they store on the server. Then multiply the increase in resources based on their activities by an estimate of how many more users will be doing the same sorts of things when the server goes live. If you don't think the suggested hardware will be up to the task, improve it. Plan for at least 10 percent more growth than you expect—just in case. It's better to find out that your system is not adequate now than to find out when everyone is using it.

For goodness sakes, storage is cheap. Use RAID to make your storage fault tolerant; mirror the web servers. If there is drive failure, you'll be grateful you did.

SPEAKING OF STORAGE

Although I am primarily referring to SharePoint's hardware needs, do not neglect SQL's needs. If you are going to use SQL, understand that it is as important as the SharePoint server itself in the performance of your SharePoint sites. Do not skimp on the hardware, particularly RAM and storage. Using RAID drives and even clustering are great ideas to help keep all that important data available.

Software Limitations

In addition to its hardware limitations, SharePoint has its software limitations. Microsoft beat the heck out of some servers to see how they performed; they found that when certain objects reached a maximum number, performance degraded significantly. This list of limitations is referred to as the *guidelines of acceptable performance*. These guidelines are something to keep in mind if your simple SharePoint Server Farm install becomes a large, busy server farm. These limitations are probably caused by a combination of the OS, IIS, and SQL performance limitations impacting SharePoint. These limits are something to remember when you are planning your SharePoint objects, such as site collections, lists, and users.

Table 1.1 provides a list of object limitations you need to know. At this point you may not really realize the importance of some of these objects, but you will. It's always good to know up front what limitations there might be for something in case you might end up being responsible for it.

TABLE 1.1: Guidelines for Acceptable Performance

Object	Number for Acceptable Performance
Website	250,000 per site collection, but performance can degrade as more sites are added.
Subsite	2,000 per website. This limit is due to the fact that enumerating the subsites of a site degrades after 2,000.
Document	5 million per library maximum, depending on the size of the documents. Keep in mind that viewing items in a large list or library can slow performance. Use list indexing, and consider making list or library views that filter content to 1,000 items or less per view.
Document Size	Generally, the maximum is 50MB. This can be set to a larger number, but it is not suggested.
List	2,000 per website. They become difficult to enumerate past that point.
Field Type	256 per list (not a hard limit, the performance just degrades at that point).
Web Part	50 per page. If they are complex web parts, the maximum decreases.
Columns	2,000 for libraries, and 4,096 for lists. Not a hard limit, but performance does degrade.
Users in Groups	2 million per website. Do not add users individually if you can help it, because many more can be added using MS security groups.
Security Principals	2,000 per website. The ACL size limits the number of users and groups in a website, but it does not affect the number of users in a group.
Indexed Documents	50 million per search index. One index server is supported per search server or server farm.
Search Servers	One search server can support up to 100 content databases. The number of search servers on a farm is based on the number of web applications being supported.
Content Databases	100 per web application. Performance degrades after adding 100. Consider creating a different web application before that point.
Site Collections	50,000 per web application (or about 50,000 per database). This is a soft limit but it causes performance degradation.
Web Applications per SharePoint Server	64. This is an IIS limitation, not SharePoint directly. It also could vary depending on load.
Web Server to Database Server Ratio	8 web servers to 1 database server. The performance degrades, but it can vary depending on environment.
Web Server to Domain Controller	3 web servers per DC, depending on how much authentication is being done.
Web Applications per SharePoint Server (Basic Install)	10 (the approximate number based on performance limitations of the embedded database).

Enumerating content in libraries and lists can be resource intensive. To ease that burden, if you have more than 2,000 items per list or library, you can limit the number of items viewed by default to 1,000, index a field in the list (only index if the list is very large, because indexing speeds up viewing a list but does add a resource load), or consider breaking up the flat list or library by using folders to organize the items to improve performance (it's okay if this makes little sense now; it will come up again).

These hardware and software factors should help you avoid the slow decay of your SharePoint server's performance. Remember to monitor, monitor, monitor. It does no good to have logs if you don't read them. Be prepared for the need to scale out or upgrade before someone else has to tell you to. If you ever overestimate the performance requirements, it's good to know that too.

So that's it. You've seen behind the curtain of SharePoint and learned about its requirements, limitations, and services. Now you are ready to get started.

The Bottom Line

Determine the software and hardware requirements you need for installing SharePoint Services 3.0 SharePoint has some stringent software and hardware requirements. Be sure you know what you need before you become the proud owner of your own SharePoint server or servers. SharePoint depends on Windows 2003 SP1 server components and services in order to function.

Master It What software must be on the server before you install SharePoint?

Identify the three ways of installing SharePoint Services 3.0 Choose the best of the three ways of installing SharePoint Services 3.0 for you. With SharePoint, how you choose to install it defines how it works. Making the wrong choice can come back to haunt you. Know what you're in for and choose the correct installation type for your business.

Master It If you were going to install SharePoint on one server (no existing SQL server) for a small business of about 50 people, what installation type would you choose?

Set up the necessary accounts that SharePoint needs to run When SharePoint is installed on a domain, it needs user accounts to assign to its services. Knowing what permissions and roles those accounts require will help you avoid problems when installing and running SharePoint.

Master It What is a Database Access Account? Is it known by any other names?

Recognize the new features and requirements of SharePoint SharePoint has features that require additional planning and setup to function properly. Make sure you know what they are and what they require.

Master It What new feature of WSS 3.0 requires SMTP to be running locally on the SharePoint server?

Plan for hardware requirements Don't let SharePoint outgrow its hardware before it really gets started. Prepare for growth. Establish your company's baseline operations per second and storage needs before installing SharePoint.

Master It What is the formula to calculate the operations per second that a SharePoint server would be doing in a given environment?

