

Part 1

Preparing for Microsoft SharePoint Foundation 2010

Chapter 1: SharePoint Foundation 2010 Under the Hood

Chapter 2: Installation Standalone

Chapter 3: Complete Installation

Chapter 1

SharePoint Foundation 2010 Under the Hood

SharePoint Foundation (SPF) is a nifty web-based collaboration, data management, communication, idea-creating, problem-solving tool that costs you nothing (assuming you already have a licensed server). SharePoint Foundation 2010 needs to run on Windows Server 2008 (Standard Service Pack 2 or higher).

SharePoint Foundation has its needs, its shortcomings, and its weaknesses, but overall, it is a surprisingly useful, flexible, and 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 for the 2010 version: SharePoint Foundation 2010 (SharePoint Foundation) and Microsoft SharePoint Server 2010 (SharePoint 2010). SharePoint Foundation is a free download and falls under the server's license model. However, SharePoint 2010, which during installation installs SharePoint Foundation and then installs its added components, costs thousands of dollars (depending on volume license) and requires at least one Client Access License (CAL) for each user.

The free version doesn't require separate CALs for each user (assuming each user already has a standard CAL for server access) and is the foundation for SharePoint. The paid-for version just adds more functionality to the foundation. So yes, SharePoint Foundation 2010 is free and (as the name implies) is the foundation for the more expensive SharePoint 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 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 in what is called a SharePoint *server farm*.

Fundamentally, SharePoint is a collection of web pages containing 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, team sites, blogs, wikis, site collections, workflows, 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, co-authoring, 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. Some lists and libraries can be set up with their own email accounts, so people can email entries without going to the SharePoint site. Existing external data can be used to populate certain lists. And when integrating with Office 2010, users can seamlessly work on files and documents using Outlook, PowerPoint, Word, Excel, and the rest.

This chapter will give you an idea of what you need to know to prepare for installing SharePoint Foundation. This kind of product does require planning. This isn't an "install it and then think about what you want to do with it later" kind of product. To make sure the initial installation and configuration of your SharePoint implementation goes smoothly, it is a good idea to know what you are going to need for success before you start.

In this chapter, you'll learn to

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

Hardware Requirements

Trying to pin down the exact hardware requirements for a product like SharePoint is tough (And when I say SharePoint, I generally mean SharePoint Foundation in this book unless I specify otherwise.). There are many different ways to use it; therefore, there are many ways to configure the resources.

Microsoft has some suggested hardware requirements. This time around, Microsoft seems to be hedging its bets and beefing up the requirements. Your mileage may vary, but chances are good that these suggestions will easily handle an average server load. There is no suggested minimum anymore, only "recommended" requirements.

Processor 64-bit, multi-core (4 preferably), 2.5 GHz per core minimum.

RAM 4 GB for developer or evaluation installations (usually meaning single-server, testing situations, not production loads), 8 GB or more for production use. Microsoft is serious about the 8 GB recommendation. SharePoint uses IIS Web Sites and application pools, which, for each one, use a considerable amount of RAM. So, the more IIS Web Sites (which correspond with a SharePoint "web application") you need, the more RAM you'll need.

Disk 80 GB, NTFS. More disk space is recommended, depending on your storage needs, such as SQL databases (if you are going to do a single-server install) or anything else running on the server.

DISK SPACE: BIGGER IS BETTER

I have noticed that, for the virtual machines I am running for this book, about 28 GB are used just for Server 2008 R2 and SharePoint Foundation. Generally, plan for about 28 GB just for OS and SharePoint, if nothing else. Also keep in mind that log files and indexes can grow to be unexpectedly large very quickly, so 80 GB is a good suggested minimum.

Planning for storage is particularly important if you are running SQL and SharePoint on the same server (as is the case in a single-server environment). You will need to plan for the storage space of SharePoint pages in IIS, SMTP mail store (if you enable incoming email), indexing files used for search, all storage space the databases would use for site lists and libraries, and all other databases SharePoint uses for additional services, such as Logging and Business Data Connectivity. 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 This is not really required for SharePoint but is useful.

Display Microsoft avoids mentioning a recommended resolution. I've found 1024×768 on the client is a functional minimum. Pages don't display well at lower resolutions. To avoid any scroll bars, many pages in SharePoint are now better viewed at resolutions closer to 1152×864 or higher.

Network Microsoft, again, doesn't mention a real requirement here. But I find that 1 Gbps is a good recommendation.

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.

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.

For this version of SharePoint, these prerequisites are pretty lengthy. So to make installing SharePoint Foundation easier, most of the software prerequisites (after you install the operating system, of course) can be installed automatically on the server during setup by selecting to install prerequisites on the preinstallation screen. Keep in mind that this version of SharePoint is all about 64-bit; it cannot be installed on a 32-bit operating system, nor can its databases be stored in 32-bit SQL.

Operating System SharePoint requires a 64-bit operating system. In production, the server needs to be at least Server 2008 SP2 (during installation, if the server is not up to Service Pack 2, it will be installed—you have been warned). It is suggested to use Server 2008 R2 or higher.

The prerequisites installer will also install some hotfixes to support claims-based authentication (a security token-based authentication method now supported by this version of SharePoint). For Server 2008, it is KB976394. For Server 2008 R2, it is KB976462.

Windows Server 2008 and 2008 R2 come in different versions, and SharePoint Foundation will install on Standard, Enterprise, and Datacenter. It will not install on Core, Web Server, or Foundation. Installing SharePoint on a domain controller is not supported.

WINDOWS 7

For the sake of developers, Microsoft has finally made it possible to install SharePoint on a Windows 7 workstation. It takes additional modifications, but it can be done. This is for development and testing purposes only, but there are numerous blogs and articles detailing the tweaking necessary to make it work.

Web Server and Application Server Roles For Server 2008, IIS 7.0, or 2008 R2, IIS 7.5, with IIS 6.0 compatibility must be installed. This makes sense. SharePoint is web-based because IIS allows Windows Server (2008 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, which hold web pages. In SharePoint, IIS Web Sites are considered to be web applications 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 Web Sites and their management (for example, specifying application pools and whether anonymous access is allowed). Those settings may be configured in SharePoint but are applied to the IIS Web Site (aka 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 be enabled in IIS.

Database SharePoint obviously requires a database on the back end in order to store data. Currently, SharePoint only supports SQL Server for its content and configuration databases (although external data can be accessed from other types of data sources in other ways).

SQL Server SharePoint Foundation requires the 64-bit version of either SQL Server 2005 SP3 with Cumulative Update (CU) 3, SQL Server 2008 SP1, CU2, or SQL Server 2008 R2. This pricey SQL 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. It can also be clustered for failover scenarios (which SharePoint can support). SQL Server is possibly overkill for small offices that are considering SharePoint. However, if you already have SQL Server 2005 SP3 CU3, 2008 SP1, CU2, or SQL 2008 R2 on your network, then by all means use it.

SQL Server 2008 Express 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 Standalone." This will install SQL Server 2008 Express during SharePoint's initial setup. SQL Express is a free, local-only database (meaning that it cannot be remotely accessed). With SQL Express, SharePoint can create and manage its databases just fine. The catch is that the Express version of SQL cannot support any other SharePoint servers accessing it and has a hard database size limit of 4 GB (you can upgrade it SQL Express 2008 R2, which can support 10 GB after install). It is not as robust as its big brother SQL 2008, and it has no graphical tools built in with which to manage and update it. The previous version of SharePoint used a different kind of single-server database engine, which had no database size limitations.

MIGRATING?

If you are considering upgrading to SPF from WSS 3.0 and you have a basic or single server installation, you will need to do a few extra things to prepare for the upgrade because of this database version change. See Chapter 15, "Migrating from WSS 3.0 to Windows SharePoint Foundation 2010," for details.

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*, discussed in Chapter 8, "Introduction to Libraries"). 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 that other SharePoint servers can access the same database. Being able to share the SQL databases 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. It 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 SQL Express to accomplish that. A server farm requires a remote SQL server that all SharePoint front-end servers can share.

You also may notice that there are a lot of references to service packs, cumulative updates, and hotfixes. At this point, Microsoft has so many products out in the middle of its release cycles, and those products have numerous fixes and improvements, that some of the features SharePoint requires must have underlying technologies with specific modifications applied. So, keep that in mind if you have other technologies that share the same resources (such as SQL) but require those modifications *not* be made. There may be incompatibility issues to contend with.

This version of SharePoint has greatly improved integration features with SQL, as well as more functionality in terms of claims-based authentication, communicating with external data sources, command-line scripting, and more. To facilitate that, this version of SharePoint has a number of additional features and components it needs in order to function properly.

The following components can be installed (if you have Internet access) during the installation process, by clicking Install Prerequisites on the SharePoint Foundation installation screen (Figure 1.1).

Windows PowerShell 2.0 This feature is extremely important for SharePoint Foundation, because SharePoint uses PowerShell to run a number of things in the background, and PowerShell is swiftly replacing STSADM as SharePoint's fundamental command-line interface. A number of SharePoint commands and capabilities depend on PowerShell 2.0.

SQL Server 2008 Native Client Even if you're not running that version of the server for your databases, you need it anyway. This is used to create new applications or enhance existing applications that need to take advantage of new SQL Server 2008 features. It's needed to support some new, under-the-covers database capabilities in SharePoint.

Microsoft Windows Identity Foundation Sometimes also known as Windows Identity Framework, it extends .NET's Cardspace support and makes it possible for SharePoint to support claims-based, security token system authentication. It particularly supports claims-based authentication for ASP.NET applications and Windows Communication Foundation services.

FIGURE 1.1
The SharePoint
Foundation instal-
lation screen



Microsoft Sync Framework Runtime v1.0 (x64): This improves SharePoint's ability to synchronize data with ADO.NET-enabled data sources or FeedSync feeds (such as RSS or Atom).

Microsoft Chart Controls for the Microsoft .NET Framework 3.5 Installs new assemblies for ASP.NET and Windows Form Chart Controls. It's particularly useful for visualization with Visual Studio.

Microsoft Filter Pack 2.0 This installs and registers ifilters with the Windows Indexing Service, which integrates with search services to index the contents of Microsoft-supported files, such as .docx, .zip, .pptx, .xls, .zip, and so on.

Microsoft .NET Framework 3.5 SP1 .NET (and subsequently ASP.NET) are required by SharePoint. This service also contains the Windows Workflow Foundation, Communication Foundation, Presentation Foundation, and CardSpace. It also supports ASP.NET Ajax, Language Integrated Query (LINQ), and web protocol support for WCF services (including Ajax, JSON, REST, POX, RSS, ATOM, and so on), as well as new classes in .NET Framework's base library. This is required for Server 2008 but already built into Server 2008 R2.

Microsoft SQL Server 2008 Analysis Services ADOMD.NET This improves data analysis, business intelligence capabilities, and data warehousing capabilities of SharePoint using SQL (it is backward compatible with earlier versions of SQL).

ADO.NET Data Services Update for .NET Framework v3.5 SP1 This supports Windows Communication Framework (WCF) in accessing data and exposing that data via web services over HTTP. This capability, along with the fully fledged WCF, enables SharePoint Foundation to perform Business Connectivity Services, discussed in Chapters 3 and 16. This is installed with Analysis Services.

Microsoft Speech Platform Runtime (x64) and Language – TELE (en-US) An optional service but installed automatically by the prerequisite installer, it assists SharePoint accessibility with speech recognition and speech synthesis. It also requires an appropriate language pack. There is a little more configuration needed (especially if you are going to use additional language packs) concerning a registry key change. Search TechNet for more details.

SQL 2008 R2 Reporting Services SharePoint 2010 add-in Also an optional service, this add-on is convenient if you are using SQL 2008 R2 with Reporting Services installed, have

SharePoint installed on the same server, and want to use Report Builder (or another tool) to make reports that can be stored in (and read from) a SharePoint library.

Of course, from the client side, users will need a browser to access the SharePoint sites.

Browser support comes in two levels. Level 1 describes browsers that are fully supported, and SharePoint is considered optimized for their access. Not surprisingly, the only browser that supports all of SharePoint's advanced features is Internet Explorer 7 or higher, 32-bit (yes, 64-bit IE is not entirely compatible with all SharePoint features). This is largely because of SharePoint's use of proprietary ActiveX controls. Mozilla's Firefox 3.5 or higher is a quasi level 1 browser, but some ActiveX features, such as Datasheet view, do not function.

Level 2 browsers are mostly supported, but they are intended to be used to do rudimentary things with SharePoint, such as reading and writing in the SharePoint sites, and to do basic SharePoint administration. Anything that truly requires ActiveX will not work for level 2 browsers.

So, what browsers are considered level 2? Any browser that isn't IE 7, 8, or higher, or isn't Firefox 3.6. That means any IE version older than 7 (or IE for Mac), Safari 4.04, Opera, and so on. IE version 6 and lower are definitely not supported.

The bottom line is that Microsoft wants you to use the most recent versions of Microsoft IE to use SharePoint—that and Office 2010, of course.

Office 2010 is really integrated with SharePoint; half the things you can do with SharePoint you can do *better* with Office 2010. Don't get me wrong, though; Office 2007 can integrate too, but not as completely as Office 2010. Keep in mind that SharePoint prefers you use the 32-bit version of Office 2010 to integrate; not all features are supported with 64-bit Office at this point (sadly enough).

There you have it. That's SharePoint Foundation under the hood—a Windows Server operating system, IIS 7.0 or higher with 6.0 support, PowerShell, .NET Framework 3.5, a handful of additional components that extend .NET 3.5, filtering, and SharePoint's use of SQL, specifically 64-bit SQL Server 2005 SP3 CU3 or higher (or you can let SharePoint install SQL Server 2008 Express). 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.

INSTALLING SHAREPOINT: SINGLE SERVER OR SERVER FARM

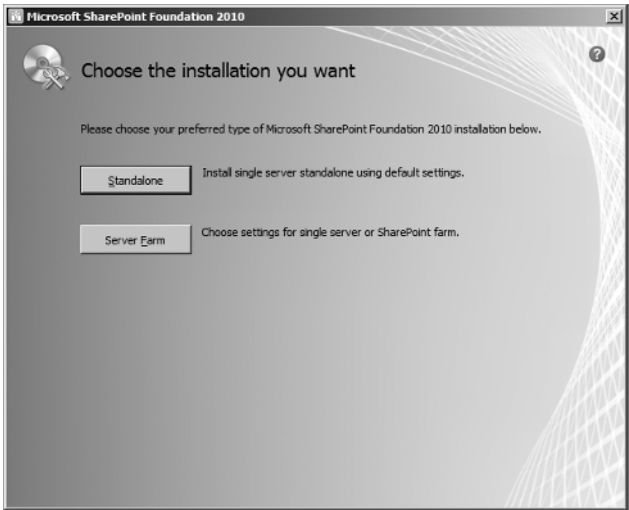
Now that you know what you need to have on the server before you even consider installing SharePoint, let's take a look at what you need to know *about* the installation process itself, which I'll take you through in Chapters 2 and 3.

There are essentially two ways to install SharePoint. Either you don't have a copy of SQL running on your network (or even on the same server) that you can use to host SharePoint's databases or you do.

SharePoint may come in two sizes (Standalone and Server Farm; see Figure 1.2), but it can actually be installed three different ways: Standalone, Standalone Server, and Complete. The last two options are under the heading Server Farm (see Figure 1.3) and indicate that the SQL Server 2008 Express database won't be installed locally; instead, the installation process will prompt for SQL server and database information.

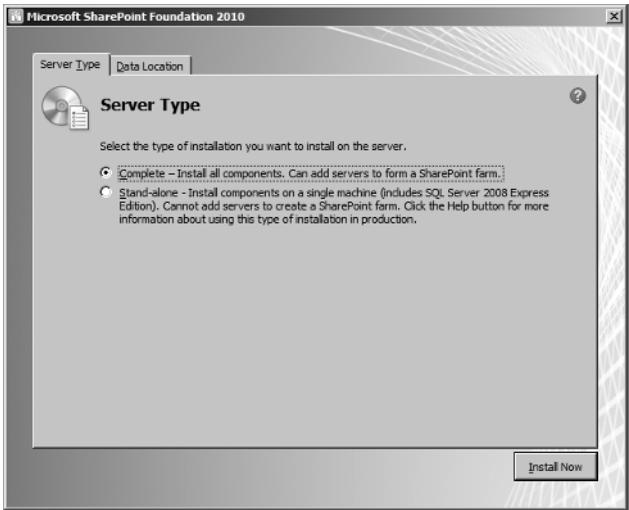
Standalone The Standalone install assumes that you are going to use only one server *ever* to run SharePoint and that you don't have a copy of SQL handy to use for its databases. What it does in that case is install SharePoint assuming all necessary services are going to run locally and that you need it to install the free SQL Server 2008 Express version of SQL on the same server.

FIGURE 1.2
SharePoint Foundation installation types



Server Farm, Stand-alone This installation is essentially the same as the Standalone install you get by just clicking the Standalone button. Use this installation method if you intend to install SharePoint on one server only (you cannot use it as a server farm server), and you want SharePoint to install and use the SQL Server 2008 Express database engine on the same server. The only difference between this install type and Standalone is that it gives you the option to specify the location of your search index files.

FIGURE 1.3
The two Server Farm options for installation



Server Farm, Complete This installation method is intended to, at its very simplest, let you specify to SharePoint, during installation, the SQL Server deployment you want it to use instead of installing SQL Server 2008 Express locally. Even if a full version of SQL Server is

installed on the same machine as you are installing SharePoint, SharePoint will not notice if you don't specify that you want a Complete installation.

That's because Standalone and Server Farm Stand-alone install SQL Server 2008 Express without your involvement; they don't give you a chance to specify anything. This means that if you have SQL Server *anywhere* on your Windows network (2005 SP3 CU3 or 2008 SP1 CU2) and you want to use it to house your SharePoint databases, then the Server Farm Complete install is the only type that lets you specify where your databases will go.

The other reason to use a 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 scaled up to a more complex topology, such as numerous SharePoint servers (generally called web front-end servers) and a 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, comprising 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; typically this means having multiple SharePoint servers and even multiple, clustered database servers. You can also separate roles such as incoming email, central administration, or services such as Search, Business Data Connectivity, Sandboxed Code, or the Subscription Settings Service.

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 you are adding it to an existing server farm. 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 a 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.

Server farms work because the databases that hold all the information about SharePoint and its server farm configuration settings (including which other databases on the server contain what data for what sites) already exist in a configuration database on the SQL server. All you have to do at that point is specify which configuration database 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 to do load balancing, 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 has been little documentation about how to do load balancing. So, check out Chapter 16, "Advanced Installation and Configuration," for a brief demonstration of how to do simple 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 SQL Express database engine, you really need to understand those differences before you install SharePoint.

SharePoint Sites and Databases

This section briefly outlines the IIS Web Sites and the databases that SharePoint will create during installation.

SHAREPOINT IIS WEB SITES

SharePoint needs at least two different IIS Web Sites (otherwise known as SharePoint web applications) to function. Most of the web applications contain the web pages that you will access to either administer SharePoint or actually use SharePoint's lists, subsites, and libraries. In addition, SharePoint Foundation adds another IIS Web Site during installation to support web services, such as Business Data Connectivity, Security Token Service, and Topology.

At the minimum, say for a simple Standalone installation, you will have these three IIS Web Sites available on the SharePoint server:

The Central Administration V4 This web application is used to control the configuration and administration of all servers on the server farm, as well as all web applications. This site is set up on a unique port, completely different from 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 Standalone installation, the port will be chosen at random for you during installation and configuration (or you can specify your own). 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. SharePoint tends to want to name its web applications with SharePoint, a dash, and the web application's port number (or host header and port number). It will contain the first top-level site for SharePoint, just to get you started (or in a Server Farm installation, you can create it yourself—or have the configuration wizard do it). Web applications were meant to contain site collections, which are collections of sites. Each one starts with a top-level site, but they 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.

SharePoint Web Services This web application, on a random port for both HTTP and HTTPS, is the one that does not contain any web pages to be used for any standard reason. This web application is used exclusively by SharePoint web services to use all of the nifty features and components available for SharePoint. It allows them to make external connections to data not stored in SharePoint's databases, handle security token for claims-based authentication, or manage the farm's topology.

Keep in mind that these are the web applications that are created during simple Standalone SharePoint installation. You can create more if you'd like. If you inherit a SharePoint server and find that more than three web applications are being used by SharePoint, that's fine. Someone probably added more for a good reason (see Chapter 10 for more information about how and why to create additional web applications) or enabled additional services, and now you are responsible for them. Congratulations.

THE SHAREPOINT DATABASES

SharePoint, of course, creates databases during the course of its installation. 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. In addition, there will be a logging database, as well as a database for Business Data Connectivity.

This means that six databases will be created when SharePoint is installed as a Standalone server (meaning most services will be enabled and configured automatically). The following list describes them using the default names. When you do a Complete installation, you can choose to specify the database names if you need a different naming convention. However, even for the Complete installation, SharePoint will suggest the default names below.

SharePoint_Config_(GUID) This is the configuration database for SharePoint. It holds all the configuration data and settings for the entire server farm. What makes separate SharePoint servers all members of the same server farm is that all of them use the same configuration database. This makes it possible for all those servers to have the same configuration settings. When you do a standalone server installation, the database will be called `SharePoint_Config_`(a string of random alphanumeric characters to generate a unique global ID or 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 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 usually the content database for the first web application made in SharePoint for SharePoint sites (it's the default, unless you are doing a Complete installation and choose to name it something else). It will contain information about the site collections that the web application contains, and it will contain all 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 it easy to 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.

EXTRA DATABASES

This version of SharePoint does come with some additional services that, when enabled and configured, require their own databases. Their default names are as follows:

WSS_Logging This database is used by SharePoint's diagnostic logging and usage to store logging and usage data.

Bdc_Service_DB (GUID) This is the database used by SharePoint to store Business Data Connectivity data. If you use external data sources to have external lists and list fields in SharePoint, this database is critical for storing the information SharePoint needs to make those connections happen.

Subscription Settings Service Database This database doesn't necessarily have a default name like the others do, because it has to be configured by hand in PowerShell (so you explicitly have to name it). It is not enabled in a Standalone server installation by default. However, if it is enabled in your environment, chances are good the person who did so will give the database a name related to the service. The Subscription Settings service can be enabled to offer multi-tenant, subscription-based hosting on the SharePoint server to separate customers, departments, whatever group or organization that requires isolated administration, authentication (if configured), and feature management. For more information about it, see Chapter 16.

Although each web application gets its own content database initially, you can add more content databases to a web application if necessary. 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 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 it's 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) so that you can be warned when a database is getting too big and be prepared to add a new database. Site collections themselves can have quotas that limit their size in megabytes to give you further control. This is particularly important should you be using a Standalone installation.

Overall, this means that SharePoint uses IIS Web Sites to drive the web applications that hold site collections. Those site collections can each contain a lot of data. Additionally, a SharePoint server farm can have a number of web applications, each with several content databases. This means your SharePoint implementation can contain numerous 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 the only one. It is shared by each of the SharePoint servers in a farm configuration. That's why, during installation, if you choose to do a Complete installation and you specify you'd like to add the server to an existing farm, you just indicate the SQL server and the configuration database, and you're basically done.

SharePoint Service Accounts and Services

After it installs, SharePoint creates and enables a number of services and application pools in order to work properly. To be able to do their jobs, these services need to run with some sort of account context. Some of those services do work only on the local machine and therefore can get away with using local accounts. But some will need to access the SQL server or other servers on the network and therefore should not use local accounts. There is also a service or two that, because of the work it does (or from a troubleshooting standpoint) should be a unique account, not shared by any other service.

Depending on how you install SharePoint, you may have to create domain accounts to apply to those services. As a matter of fact, SharePoint has a health analyzer in which there are rules that certain services must have unique domain accounts. To understand SharePoint and keep it in good working order, it helps to know what those services are, what they do, and what access their accounts need while remaining secure.

Service Accounts

Depending on how SharePoint is installed, you may have the following service accounts:

Setup Account (Standalone Install) 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 (or the equivalent). 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.

With a Standalone installation, all other service accounts used by SharePoint are set up automatically (using local system or network service accounts). It really is the easiest installation, in addition to being the cheapest. Although it is not really scalable, it is convenient. It is also a great way to get an understanding of what SharePoint and its services look like when running, and it gives you a chance to simply get started using SharePoint. Once you've explored its functions, it makes it easier to then do a Complete installation and configure the services manually, because you will know how they work.

THE CHEESE STANDS ALONE

You don't have to install SharePoint in a domain environment. You also can install SharePoint on a stand-alone server in a workgroup with no domain controller.

Just install SharePoint using the Standalone option on the server (or, if you don't want to use SQL Server 2008 Express, install SQL on the server, and then install SharePoint using the Complete option).

If you choose a Standalone install, the databases and services setup 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.

Having both SharePoint and SQL on the same server means that all the database and service management can be done without needing to access anything on a different server and therefore only need to use local accounts.

Local users and groups will be used to give users access to SharePoint in that scenario, rather than going through a domain controller. You can still support incoming email in that scenario by enabling SMTP in IIS and then setting it up in Central Administration.

It just goes to show that SharePoint is scalable down as well as up.

Setup Account (Complete Install) 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: Login, SecurityAdmin, and DBCreator.

Farm Account Also known as the *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 farm account to the SQL server's Logins, DBCreator, and SecurityAdmin roles. This is why the farm account ends up being the owner (DBO) of most of the SharePoint databases.

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 managing alerts). This account should be guarded and not used for anything else (except for the one rare occasion of setting up a super-admin account in PowerShell).

THE DBO EXCEPTION

Oddly enough, the farm 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 farm account holds an owner role. This also means that, in a pinch, the setup account can be used to do farm administration in PowerShell, if necessary.

Content Database Access Account Also known as the *content database account*, *web application account*, or web application *application pool account*, this is the account that uses the content database(s) 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 the web application it is working for.

If you are going to have more than one web application, you may want to consider creating a content database access account for each of them. This helps give the account least privilege (if it is compromised, it can only affect that web application; if it fails, it causes only one web application to fail), and it is easier to troubleshoot if each web application has its own content

database access account. However, each application pool does use server resources to function, so some organizations actually require there be a limited number of application pools for a SharePoint implementation. It can be a balancing act, but it's something to keep in mind when planning accounts for SharePoint.

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 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. Often administrators just use the search account for both search and content access services.

ADDITIONAL SERVICES MIGHT NEED LOVE TOO

If you enable the new Business Data Connectivity service, Sandboxed Code, or Subscription Settings service, they will need their own service accounts as well. Keep in mind that content database and service accounts like Search (but not Index, strangely enough), will require that their account be registered in SharePoint as a managed account before being applied. So, it is a good idea to plan for the accounts, create them in Active Directory, and then register them as managed accounts soon after SharePoint installation.

Optional SharePoint Admin or SharePoint utility Account I also suggest you consider a general-purpose SharePoint administrator account. This account should be a domain admin (or at least a 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 be used as an 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 administrator 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).

PREPARING FOR POWERSHELL

New with this version of SharePoint are SharePoint-specific shell cmdlets (pronounced "command-lets"). Microsoft is going all out with PowerShell, hoping that customers will prefer to use it rather than the old favorite, the STSADM command-line tool.

With STSADM, the permissions required to run a command depend on what you want to do. It has to be run on the SharePoint server locally (or at least on a server in the farm), so the account needs to have local administrative rights on the server. If you are doing farm-related commands, such as creating a web application or starting a service, the account also needs to be a farm administrator. If the commands are only for a site collection, the account needs local admin rights on the server and needs to be a site collection admin for the site collection being worked on.

With PowerShell, things are different. Maybe because the tool is so new or maybe because it has been more the focus of developers than administrators, PowerShell requires the accounts that use it to be very powerful, possibly insecurely powerful. For an account to use PowerShell, it must be a local administrator of the SharePoint server. Then it must have ownership rights to the farm's configuration database, as well as be a farm administrator (meaning that the account will be added to the WSS_Admin_WPG group on the SharePoint server). If that account also has to do *any* work in a specific web application (site collection, and so on), it must also be an *owner* of the content database in SQL related to the web application (or site collection). So, for a farm administrator to be able to work on anything the farm needs, the account needs to own (have the owner permission in SQL of) all content databases and the configuration database of the farm.

There is a PowerShell cmdlet (otherwise known simply as a *command*) that is used to give accounts PowerShell admin rights, `add-spshe ll admin`. This command has to be run by an account with the rights to do so in order to apply the correct permissions to the added account so it too can use PowerShell.

This causes a chicken-and-egg scenario: you need to give shell admin rights to accounts to work in PowerShell, but there is no account available to run the command to give the rights to other accounts...except for the farm account.

The farm account should never, ever, ever be logged in and used as a normal account. This account, if made (very temporarily) a local administrator of a SharePoint server, could open the SharePoint Management shell (Start menu > All Programs > Microsoft SharePoint 2010 Products > SharePoint 2010 Management Shell) and be used to give shell admin rights to the accounts or AD security group that need it. Then, when you are done using it, log the account out, log back in as a normal administrator, and remove the farm account from the local administrators group (SharePoint hates it when the farm account is a local admin, and there may be an error saying so, if it notices what you've done).

When an account is added as a shell admin account, it is added to the configuration database in SQL with the owner role and a SharePoint-specific role called `SharePoint_Shell_Access`. If you want to have an account also be able to manage a content database of a web application via PowerShell, they need to be added as a shell admin specifically to that database (or databases). In doing so, the cmdlet adds the `SharePoint_Shell_Access` role to the database and then gives the account rights to it. Conceivably, you can have different shell admins with rights only to make changes to the configuration database (by owning it) and to only a certain content database that you have specified. Usually I use an account that is added as a shell admin to all databases for the farm. I consider it my PowerShell super-admin account.

Keep in mind that site collection administrators should not be given PowerShell capabilities if you only want them to be capable of managing their own site collection. PowerShell admins are able to manage everything in a web application they are given rights to (which contains site collections, so they'd manage all of them) or nothing. Site collection administrators will still have to use STSADM for their command-line work.

There is an entire chapter dedicated to getting you up to speed with PowerShell (Chapter 14, "STSADM and PowerShell"), but I wanted to give you some advanced warning when planning for SharePoint, that there may be the need to give a few accounts a lot of power to do damage, not just to every database related to SharePoint but potentially to SQL as well (since every account will need a login role).

In my case, I plan to give my SharePoint admin account the right to be a shell admin for the whole farm in order to be able to use it, at will, to do work—making it my super-admin account (see Chapter 14 for more about how that’s done). Initially, we will be doing a lot of work either in Central Administration or with STSADM. Later in the book, as you get the hang of using SharePoint, we will begin to do more and more with PowerShell.

SharePoint Services

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

SPAdminV4 (SharePoint Foundation 2010 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 service runs as the WSSADMIN process in Task Manager.

SPTimerV4 (SharePoint 2010 Timer) This is the service in charge of actually triggering and running jobs for SharePoint. Because it uses the farm 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 SPAdminV4 account to do it. This service runs as the OWSTRIMER process in Task Manager.

SPSearch4 (SharePoint Foundation Search V4) This is the Search service for SharePoint. It runs on the SharePoint servers that are running the Search service. This service runs the mssearch and mssdmn processes in Task Manager.

SPTraceV4 (SharePoint 2010 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.

SPWriterV4 (SharePoint 2010 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 starts only when necessary. So, it’s not always running.

SPUserCodeV4 (SharePoint 2010 User Code Host) This service is specifically for something called *sandboxed solutions*. Basically for developers, SharePoint now allows certain controls and limits for running solutions. This service executes code in a “sandbox.” In the past, solutions could unintentionally take up a lot of server resources because of unhandled exceptions, abnormal process termination, and the like. So, now solutions that are installed on the server farm, if they are scoped to be sandboxed solutions, can be managed (limited or stopped) by this service. This service runs as the SPUCHostService process in Task Manager. It is not always enabled by default.

BAD SOLUTION, NO DINNER FOR YOU...

Solutions can be pointed to the User Code service and, using something called *solution affinity*, be made to run only on the servers that have this service started. Any sandboxed solution (which now includes site templates) is allowed only so many “resource points” within which to run. If a solution exceeds its allowed points, it’s turned off. The user using the solution may not know what’s going on or be able to do anything about it, but now you can sleep safely, knowing that at least it isn’t crashing the server.

A few services that SharePoint uses straddle the line between a SharePoint capability, service application (which uses an application pool identity to function), and a simple service. They include the following:

Business Data Connectivity Service (BDC) This Business Data Connectivity Service application enables external connection types, such as non-SharePoint SQL databases, which allows SharePoint to surface that non-SharePoint data in external lists. It uses the Web Services web application in IIS and has its own application pool identity to function. Although it isn’t a separate service running in the Services console or Task Manager, it has its own database and an identity, which you can specify. This service is also known as Business Connectivity Service (or BCS) in some documentation.

KEEPIN’ IT BRIEF

Most of these services, if you were to look at them in Central Administration, would start with the words *Microsoft SharePoint Foundation* and then the actual unique task the service does, such as User Code Service. I’ve taken the liberty of shortening these titles to avoid too much repetition.

Application Discovery and Load Balancer Service Application This service simply supports load balancing and discovery of farm-scoped applications. It doesn’t really have its own service account and has no configuration settings to speak of.

Subscription Settings Service This service, for the most part, is not surfaced for configuration in Central Administration. A number of service applications normally used by SharePoint Server 2010 have been moved to SharePoint Foundation, and this is one of them. The Subscription Settings service applies to something called *multi-tenancy*. This kind of SharePoint setup requires configuration via PowerShell; it cannot be done in Central Administration. It basically uses a GUID to put site collections together in a “subscription.” Very much the way SharePoint online works, it allows for a SharePoint implementation to be configured in such a way that each department or client has their own SharePoint “deployment” with limited administrative surface. This service also allows the farm administrator to create “feature packs” or group together features and then apply them to certain subscriptions, giving those site collections only the features available in the pack. For more about multi-tenancy, see Chapter 16.

Workflow Timer Service A subservice of the SharePoint Timer service that runs the farm, this service simply sets the number of workflow events that are processed every timer interval for the server.

Central Administration and Web Application Both are services you'll see in every SharePoint farm. The Central Administration service runs only on SharePoint servers that are hosting the Central Administration site. On most server farms, only one server needs to host that site. Web Application is the service that lets SharePoint have web applications, serve pages, and so on. It is fundamental to SharePoint, and every SharePoint server runs it. If you enable Central Administration on a SharePoint server that did not originally have it running, it will generate a web application on the server locally to support Central Administration's site collection. If you disable the Web Application service, the server will stop answering user requests for web pages. This is useful if you want to run services, such as Search or BDC, but not waste that server's resources offering pages to users.



Real World Scenario

USER ACCOUNT MODES

When you install SharePoint, it automatically defaults to using Active Directory (AD) to supply the user accounts to be used as users for the SharePoint sites. This means that you need to have the user account in AD (or on the local server in a non-domain, standalone environment) before it can be added as a user in SharePoint. This user account mode is called Active Directory Domain Account mode.

However, there is another user account mode available, called Active Directory Account Creation mode (ADAC). This lets you create the account in SharePoint *first* and then adds it to an organizational unit (OU) that you set up specifically for SharePoint in AD. This mode has limitations; the account has to be added as an email address, the same email account cannot be added as a user to more than one site collection, and it disables several settings in Central Administration, particularly those that have to do with configuring or managing site collections so that they can only be run in the command interface (with STSADM or PowerShell). This mode focuses quite a bit on applying and isolating accounts per site collection.

Enabling ADAC is an advanced setting and can be done only during the installation of SharePoint. It is a one-shot thing; it defines the way user accounts are applied to SharePoint, period. There is no easy way to undo the choice, because it is locked in as the user account mode in the configuration database for the whole SharePoint farm by the time installation is complete.

You get the chance to select the ADAC account mode by clicking the Advanced Settings button during configuration. If you miss that button and complete the installation, the default Domain Account mode will be applied.

Although SharePoint Foundation still supports ADAC (SharePoint Server 2010 does not), it has been overshadowed by the capabilities of the Subscription Settings service, which uses multi-tenancy to isolate site collections more effectively and can either isolate users in their own OU in AD or use forms-based authentication (FBA), which lets you use a SQL database to store user accounts for web applications (and the site collections within them) instead of AD.

Because of this, I will point out the Advanced Settings button during Chapter 3, "Complete Installation," but I will be focusing more on multi-tenancy in this book instead (Chapter 16). FBA is rather fiddly and outside the scope of the book, but it can be applied per web application (or extended web application) and, like multi-tenancy, is better than the "all-or-nothing" approach of ADAC.

Authentication Types

In conjunction with IIS, SharePoint supports several ways to allow users to authenticate. They are not exclusive; you can 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.

ASP.NET Impersonation Authentication This authentication type allows an ASP.NET application to use a specified account to act as its identity or run as a user authenticated by IIS (such as IUSR if Anonymous Access were enabled).

Windows Integrated Authentication This authentication type 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 is 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 type will send authentication information across a network as clear text, which is obviously not a great idea. It is sometimes required by mobile devices.

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

Usually ASP.NET Impersonation and Windows Integrated Authentication are enabled by default for SharePoint web applications.

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 is the simplest to use.

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 database access accounts 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.

Although Kerberos can be the best option for authentication, NTLM is still suggested in many cases, 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 it to authenticate can be a problem due to time synchronization. Another consideration is that, in some situations, the Index service used by the Search service (discussed next) cannot authenticate using Kerberos (on custom ports in particular) and therefore cannot index sites that require it. That said, Kerberos can be a little faster and is more secure than NTLM. Further, it can also be used for those service applications that do pass-through authentication. For more information about Kerberos and how to configure it, see Chapter 16.

Claims-based authentication Not technically an authentication method, SharePoint now also supports claims-based authentication. This allows SharePoint to support standard Security Assertion Markup Language (SAML) tokens, giving it the option to support non-Windows accounts in a Kerberos ticket kind of way. It also makes it possible to use security tokens to better protect authentication using forms-based authentication. If you enable claims-based authentication for a web application, you can still use NTLM or Kerberos, but claims-based is the option that supports forms-based authentication as well.

SharePoint Search

The Search service for SharePoint Foundation is basically the same one used for the previous version (WSS 3.0). The interface has changed a little and is now called SharePoint Foundation Search.

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 the 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 and then merged 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. When installing SharePoint using either the Complete or Server Farm Stand-alone installation, you can use the option to save the index files to a different location. Consider setting aside a different drive or partition on the server to handle the possible large amount of files. (If you do specify where the index goes and if you decide to move it later to a different server, be sure the same path is available for the index files on the new server to avoid issues.)

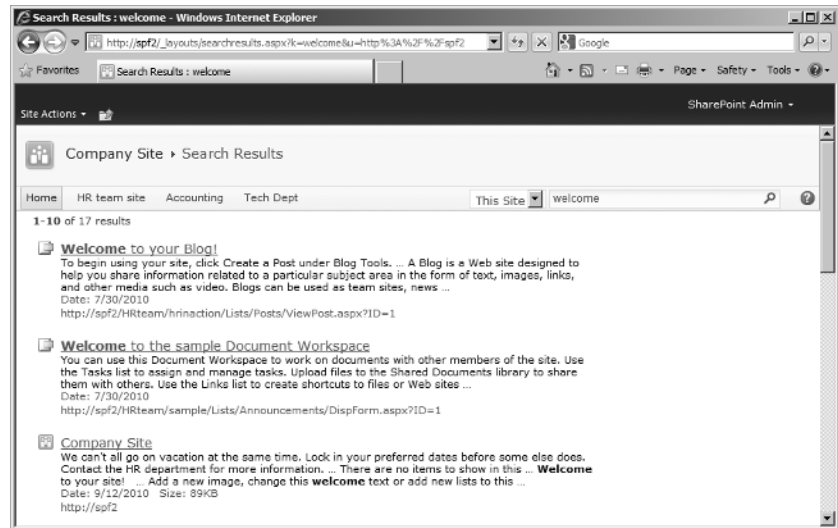
Search has some strengths and weaknesses that you should know about before you install SharePoint:

- ◆ Search doesn't have much of an administrative surface. The GUI settings are limited to what service accounts are used, 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 or PowerShell. See Chapter 14, "STSADM and PowerShell," 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 SharePoint Server 2010 (which, for the added cost, can search multiple external sources or even multiple SharePoint server farms), Microsoft Search Server 2010, or installing Search Server 2010 Express.
- ◆ 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 is security filtered and searches along a path. Therefore, it generally 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 considered a hard-search boundary because of two things. Site collections are usually at the top of a path, like `http://server/sites/sitecollection1`. Everything within and under the `sitecollection1`'s top-level site would be included in a search. Then those results are further filtered by the querying user's permissions. Normally a user is a member of one site collection but not the other. This usually works fine for most users, but if you are, say, the owner of more than one site collection in a path, check the URL of your search results to be sure they are from the correct 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*. However, strangely, the word *or* is neither ignored nor recognized as part of the query logic and is treated like part of the query text itself.
- ◆ Unfortunately, Search for SharePoint Foundation 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 file type, content type (used for libraries particularly), author, title, or subject. To filter in the search field by property, the syntax is `property:query`; for example, `filetype:txt` will result in all text files in the site collection. Searches are scoped in a way (but not as well as the previous version). Search

will be scoped depending on where you initiate the search. If you are on a list or library, even though the search field will say “Search this site...,” it will actually only return results for that list or library. However, on the search result page, you can then change the scope to “This site” (which strangely also places you one step from the site’s home page, regardless of the list you were originally on when you first initiated the search), and it will search the whole site for your query.

- ◆ The search results are displayed on a page organized by relevance. Results are displayed with the link to it and some summarizing information (see Figure 1.4).

FIGURE 1.4
The Search
Results page



- ◆ In a server farm, there is usually only one Search service and one Index (content access) service running. However, if you have a large and busy server farm, it might be good to have a server dedicated to searching and indexing, 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 (make sure the site is set to always search .aspx pages). In addition, the Index service prefers NTLM authentication, so it can have problems accessing a web application that requires Kerberos if it is using a nonstandard port (ports 80 and 443 are considered standard). Search generally uses the default zone address for the web applications it indexes, so it might be best to keep that zone’s authentication method to NTLM (and to consider using extended web applications to host Kerberos or anonymous access).
- ◆ Search does 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.
- ◆ Sites and lists can be excluded from indexing if you’d like them to be unsearchable. This can be useful for lists with item-level security (which can confuse security trimming), and may cause some items to be displayed in the search results for those who can’t open them.

INDEXING AND GATHERING

The Search service's Index service is an old hand-me-down from SPS 2003 and MOSS. The Index service is a powerful feature that you don't need to monitor. It takes care of itself and does its own thing. (SharePoint Server 2010 has extensive 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 64 MB of indexed words per document. 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.

The ifilters that come with SharePoint can handle Office 2007/2010 file types, text files, HTML, and TIFF files (the file type usually created by scanning faxes and documents). If you'd like to be able to index other types of files, a number of additional ifilters are available from their manufacturers.

SharePoint and Email

SharePoint integrates easily with email. However, it does take some consideration concerning 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 can 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 (or library) without going to the SharePoint site if you know that list's email address. And you can do all of this from the comfort of your email program. There's no need to open a browser.

To manage incoming email, the SharePoint server should have the SMTP service set up locally (make sure it has started). It is best to have SMTP 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 feature called SharePoint Directory Management Service (DMS). This service integrates SharePoint with Active Directory and Exchange. 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 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 it works best with Exchange 2003, because it integrates so deeply with AD. Later versions of Exchange can support DMS but may require considerable additional configuration.

DON'T LET THEM SCARE YOU

Despite occasional documents stating otherwise, SharePoint Directory Management Service does not have to be running for SharePoint to handle 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.

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 your users to be able to access it the same way they do other Web sites or 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 (AAM) is about mapping a SharePoint web application to an alternate address other than the default. This way, you can have an internal, default name of `http://spf2` and an Internet URL of `http://SharePoint.dem0tek.com`, both pointing to the same server (and more importantly, to the same content).

AAM specifies alternate access to 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 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. An extended web application is just an IIS Web Site that points to the same content database as an existing web application. This is done if you want to use some other URL, security settings, or authentication type to access the same data (essentially if you want to use different IIS Web Site settings to access the same content, like anonymous or Kerberos). That way, users can have more than one way to access the same data, especially if you want to have different types of authentication for the content, depending on what URL the user uses.

Because an 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 in the web application list 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. You might want to note that there are a limited number of AAM zones available to extend (Intranet, Internet, Custom, Extranet) per web application. The Default zone is the original web application's URL, so obviously that is not available to be used for extending.

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 SPF2, then its URL can be `http://SPF2`, 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* wildcard managed path for

additional site collections. The URL for that path would be, on the same server, `http://spf2/sites/`. What this means is if you create that additional site collection, it can be something on that path, such as `http://spf2/sites/something`.

You can, of course, create your own managed paths, 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, and so on.

A site collection makes a good user account or permissions boundary because you can add users once to the top-level site, apply permissions to them either individually or in groups, and if the subsites inherit permissions (which they do by default), those users will be able to access subsite resources with those permissions 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 websites or web software you want to run in the same IIS Web Site virtual directory, SharePoint automatically ignores anything that is on a path not specified as a managed path.

User Accounts and Permissions

For anyone to use SharePoint, there must be users. SharePoint leans toward organizing users and permissions based on the users' roles. So, a site owner would need to have full control of the site, but a member would only need to be able to contribute.

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

Individual Active Directory (AD) users can be added to SharePoint, but you can also simply add domain security groups as well. Doing so lets you add numerous users to SharePoint that might require the same permission levels at one time. It is also easier for SharePoint to handle because it has limitations on how many separate security principals it can manage at one time. It's actually considered a SharePoint best practice to use AD security groups to add users rather than individual domain users for that reason (the same applies if SharePoint is on a non-domain server, only they'll be local server users and security groups).

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 (permission levels are combinations of individual permissions). 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. Default permission levels include Full Control (all permissions), Read (permissions that only allow a user to view the site and its contents, but not add, edit, or delete), and Contribute (permissions that allow users to read, add, edit, or delete site content). So when planning your user management strategy, keep permissions, permission levels, and SharePoint groups in mind.

Another thing to consider is that you can apply user policies to a web application that affect all the site collections contained in them. A user policy is when you add a user to the web application and explicitly allow or deny them permissions to access the web application (and therefore all site collections contained therein). A user policy at the web application level overrides permissions at the site collection level. This means a user account, given the correct permissions in a user policy, can log into any site collection in the web application, even if that account is not

a member of the site collections in any way. This also works from the standpoint of denying a certain user or security group permission; so even if they are added to a site collection within a web application, they will not be able to make use of the denied permissions. Keep this in mind as you plan for user accounts and permissions while designing for your web applications and site collections. For more on users and permissions, see Chapter 12.

Performance Planning

By now you may have made some plans concerning what OS you'd like to have on your SharePoint Foundation server; what installation option you'll use; and what accounts, services, user account mode, and authentication you'll implement. You may have decided how you'll handle Search (such as putting the index files on a separate drive), set up email, manage paths, and alternate access mapping, and you've mapped out the user accounts, permission levels, and groups you'll need.

Now you need to determine whether your server is big enough for the job—today and into the future.

KEEPING UP-TO-DATE

Planning for performance and storage is more an art than a science. And as time passes (and more people use the most current version of SharePoint), the ideas about the current best method for performance or storage planning change to suit. So, be sure to check online to see what the most recent best practice is for planning for SharePoint.

How do you plan for that? There are really two points of concern, performance and storage. In terms of performance, it's good for you to determine how many operations per second (OPS) your server will need to do under normal (or even extreme) loads. (Storage is measured in input/output operations per second, or IOPS, which is a little different.)

There are probably as many different ways to plan for performance as there are people using SharePoint, but for a ballpark, general estimate, there is a tried-and-true formula from WSS 3.0 to help you with your plans. This formula is very simple, but it can give you a good idea of how to avoid being underpowered in terms of simple user activity.

Essentially, you need to answer the following questions:

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 do they do on average (how many documents edited, list entries added, searches done, and so on)? (Operations)
4. How many hours do the users work in SharePoint on an average day? (Work hours)
5. Does an average work day have particular peaks in performance? (Peak factor)

To calculate the operations per second, multiply items 1, 2, 3, and 5 together and 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 $\times$ 60 minutes per hour $\times$ 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 suggested 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 ALWAYS ALL-INCLUSIVE

Many businesses don't always allow every employee access to SharePoint. Therefore, when you determine who will use the SharePoint sites, don't just include everyone in the company by default. 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 as follows:

Users $\times$ Usage $\times$ Operations $\times$ Peak $\times$ (360,000 $\times$ Work hours)

Or in our case:

$1,000 \times 60 \times 50 \times 4 \times (360,000 \times 9)$

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 shown previously, using a quad-core, 2.5 GHz per core processor and 8 GB of RAM should be enough to handle at least 10 user 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.

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. Having more web applications means more RAM. Indexing is also RAM-intensive. Keep in mind that SharePoint often rapidly increases in use and increases 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. If you can afford it, consider calculating your OPS requirements and then doubling them to prepare for the inevitable, large increase in use.

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 five minutes or less. Instead, it's better to index every hour or at certain times of the day. This can be difficult if you expect SharePoint to index and search new items almost instantaneously; just keep it in mind if you are trying to squeeze as many operations per second as you can from your server. Indexing can be RAM-intensive. If your server is taking a long time to index documents and indexing is peaking your RAM usage, consider increasing the RAM on the server doing indexing.

Usage and Health logging SharePoint can analyze site usage and deliver detailed reports. However, analyzing the usage logs takes a considerable amount of processor power. Try to schedule the analysis to occur during a long downtime, usually sometime around 3 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.

Web Applications Although web applications may not, by themselves, increase processor load, they do take up about 50 MB or more apiece in RAM on the SharePoint server (especially if they are using different application pool accounts). This is one of the reasons that you might want to consider consolidating site collections into as few web applications as possible.

Features and Solutions Custom features and solutions can be added to SharePoint and scoped at the farm, web application, or site collection level. Be sure to test those added components to know how many resources they use when active. Realize that although now there can be sandboxed solutions, or solutions that are deeply throttled in terms of the resources they're allowed to use, and scoped to be added and activated at the site collection level, no matter how frugal they are, if you have many solutions or features they can impact the server's CPU and RAM over time.

Storage Planning

When you're considering performance issues, don't forget to plan for adequate storage. If you plan to have SharePoint and the SQL Server 2008 Express 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 databases, which holds all SharePoint's precious content—are stored on a different SQL server, planning for storage is still important.

Consider that the maximum default size allowed for document uploads is 50 MB. A large multimedia Word document can often be about 5 MB, so a maximum of 50 MB is usually more than sufficient. Of course, you can adjust the size; this is just a good default. And of course, if you upload more than just 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. For example, assume they are creating marketing materials to send out every quarter, and they are storing and collaborating on the materials in a document library. If they create five major documents each quarter, that would be 20 large documents per year, possibly up to 10 MB per document. That could be 200 MB of space for those documents alone. If other people manage the images for the document in a picture library and the material had 10 large, full-color pictures per document, that could be 2,000 MB (2 GB) 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 1.4 GB 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 kilobytes, if that. However, if you enable attachments for the lists or libraries, those files (by default less than 50 MB) 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 very likely be returned to the original library, and the workspace will probably be deleted when the project is done. Until then, however, that document (and its workspace) is yet another thing requiring storage. You can also allow users to create their own site collections (with Self-Service 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 64 MB of stored words for a single document. That means if you have 20 documents in a library, the search database can have (maximum) 1.3 GB of entries for that library alone. Of course, if the documents themselves never exceed 50 MB (which should have less than 64 MB of unique words to store) and Search sticks to its 40 percent

limit for each document, then that would be no more than 20 MB of indexed entries per document and therefore (going with our scenario) about 400 MB 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.
- ◆ Also, on the SharePoint server, always leave some room for the paging file; try to go for at least the same amount as the server has in RAM.

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. There is a standard formula going around that might help in estimating for user storage:

$$\text{Database size} = ((D \times V) \times S) + (10 \text{ KB} \times (L + (V \times D)))$$

D = documents

S = average size of documents

L = list items (harder to average, but smaller; suggested three per document)

V = estimated number of versions

10 KB = constant (the estimated amount of metadata used by SharePoint)

So, in this case:

D = 200 yearly (that's 20 documents and 10 pictures per document)

S = 10240 KB (that's 10 MB per document/picture, in kilobytes)

L = 600 (rounding to three times the documents, a rule of thumb considering that there is likely to be a couple discussion items and a calendar entry per document)

V = 7 (estimating the number of versions for the documents)

So the formula would be as follows:

$$((200 \times 7) \times 10240\text{KB}) + (10\text{KB} \times (600 + (7 \times 200)))$$

$$\text{or } 14,336,000 + 20,000 = 14,365,000\text{KB}$$

$$14,356,000\text{KB is about } 13.6909 \text{ GB (rounded up to } 14\text{)}$$

This averages to a storage size of about 14 GB (estimating high of course) per year. Then you need to factor in the storage space that Search will need (about 40 percent of the size of stored files, or in the case of 14 GB, factor in an additional 5.6 GB).

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 20 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.

Storage is well worth the price you pay for it (and is often inexpensive). Use RAID to make your storage fault tolerant; mirror the web servers, and cluster your SQL servers if you can (especially now that SharePoint is failover aware). If there is drive failure, you'll be grateful you did.

Software Limitations

In addition to its hardware limitations, SharePoint has some software limitations. Microsoft beat the heck out of some servers to see how they performed; and they found that when certain objects reached a maximum number, performance degraded significantly. Previously, the list of limitations for SharePoint was considered a guideline of acceptable performance. Now it is considered software boundaries and limitations.

These boundaries and limitations come in three types:

Boundaries Hard limits that cannot be exceeded by design.

Thresholds Limits that have a default value for best performance, but that value can be changed.

Supported Limits Recommended limits, based on testing. Surpassing these limits could result in performance issues and possible “deleterious” effects.

Tables 1.1–1.3 list the 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. These are not the full and comprehensive tables (which are available on TechNet; just search for *SharePoint 2010 software boundaries and limitations*); they simply list the most relevant to SharePoint Foundation in a standard installation. (Note that much of what is available online is related to the SharePoint Server product more than SharePoint Foundation.)

Table 1.1 contains the software boundaries for this version of SharePoint.

TABLE 1.1: Software boundaries for SharePoint Foundation 2010

OBJECT	MAXIMUM VALUE	TYPE OF OBJECT	DETAILS
Zone	5 per web application	Web application limit	This limit is hard-coded per web application as Default, Intranet, Internet, Extranet, and Custom.
List Row Size	8,000 bytes per row	List and library limits	Each list or library item can take only up to 8,000 bytes (actually 7,744) in the underlying database.
File Size	2 GB	List and library limits	The default maximum file size that can be uploaded is 50 MB, but that can be changed. The absolute maximum is 2 GB.
Bulk Operations	100 ops per bulk operation	List and library limits	The user interface only allows for a maximum of 100 items to be selected at once for bulk operations (such as copying to a library from Explorer view).
ECT (External Content Type) in Memory	5,000 per web server	BDC service limit	Only 5,000 different ECTs can be in memory per server at any given time.
External System Connections	500 per web server	BDC service limit	The default maximum is 200, but the boundary is 500.

Table 1.2 describes a number of the software thresholds for SharePoint Foundation. Remember that thresholds are default settings, many of which can be changed. These thresholds are suggested for best performance.

TABLE 1.2: Software thresholds for SharePoint Foundation 2010

OBJECT	MAXIMUM VALUE	TYPE OF OBJECT LIMIT	DETAILS
List View Lookup	8 join operations per query	Lists and libraries limit	Any list view can only have eight lookup fields (and that includes People And Groups and Workflow progress fields). More than that will be blocked. Can be changed in Resource Throttling per web application.

TABLE 1.2: Software thresholds for SharePoint Foundation 2010 (CONTINUED)

OBJECT	MAXIMUM VALUE	TYPE OF OBJECT LIMIT	DETAILS
List View	5,000 items	Lists and libraries limit	The default maximum number of items that a query can process at one time. Can be changed in Resource Throttling per web application.
List View for Auditors and Administrators	2,0000 items	Lists and libraries limit	The default maximum of items that an object model database can query at one time. Works with the Allow Object Model Override setting. Also a Resource Throttling setting.
Subsites	2,000 per site view	Site limit	Enumerating the number of subsites for a site collection starts to degrade performance above 2,000; it also degrades the performance of the All Site Content page and Tree View.
Co-Authoring	10 concurrent editors per document	Lists and libraries limit	The recommended limit is 10, but the boundary is 99. More than 10 coeditors on a document can cause conflicts and degradation in performance.
Security Scope	1,000 per list	Lists and libraries limit	This is the maximum number of unique security settings per list or library. A security scope can contain the ACL (access control list) for an object and the security principals related to that ACL as well.
Single Line of Text	276 per list	Column limit	There should be no more than 276 single-line-of-text-columns per list.
Multiple Lines of Text	192 per list	Column limit	
Choice	276 per list	Column limit	
Number	72 per list	Column limit	
Currency	72 per list	Column limit	

TABLE 1.2: Software thresholds for SharePoint Foundation 2010 (CONTINUED)

OBJECT	MAXIMUM VALUE	TYPE OF OBJECT LIMIT	DETAILS
Date and Time	48 per list	Column limit	
Lookup	96 per list	Column limit	
Yes/No	96 per list	Column limit	
Person or Group	96 per list	Column limit	
Hyperlink or Picture	138	Column limit	
Calculated	48	Column limit	
Web Parts	25 per wiki or web part page	Page limit	This is an estimate based on simple web parts. Above this threshold, and performance suffers. If the web parts are more complex, expect performance degradation sooner.
Workflow Postpone	15	Workflow limits	15 workflows are allowed to be executed against a database at any given time. After that, additional workflows will be postponed. Those new requests count against this threshold.
Workflow Timer Batch Size	100	Workflow limits	This is the standard limit that the workflow timer job can pick up at any one time. Configurable. Additional workflow timer job instances can be started to handle more batches if necessary.
Database Items Returned per Request	2,000 per database connector	BDC service limits	The default maximum of 2,000 is used by the database connector to restrict the number of results that can be returned per page. The boundary is 1,000,000.

Table 1.3 describes a number of the supported limitations for SharePoint Foundation. Remember that supported limitations are ones that, although they are not hard boundaries or thresholds that can be changed, are suggested limits because tested performance has been found to degrade after these limits are reached.

TABLE 1.3: Software limitations for SharePoint Foundation 2010

OBJECT	MAXIMUM VALUE	TYPE OF OBJECT LIMIT	DETAILS
Content Database	300 per web application	Web application limit	With 300 content databases per web application, end users may not see a change in performance, but administrative operations such as creating a new site collection will experience a decrease in performance. Using PowerShell for administration might help.
Managed Paths	20 per web application	Web application limit	Managed paths are cached on the web server, and beyond 20, there is a degradation in performance.
Application Pools	10 per web server	Web server, application server limit	It does depend on the server's RAM and the workload the farm is servicing. However, this is a good rule of thumb. If you need more, test thoroughly for performance issues.
Content Database Size	200 GB per database	Content database limit	There is no hard limit, but it is strongly suggested to limit individual content databases to 200 GB apiece. Larger databases are acceptable for a single site repository, such as a records center, but it is not suggested.
Site collections per Content Database	2,000 recommended; 5,000 maximum	Content database limit	This limit is based on how long it takes to upgrade. Also, the more site collections there are in a content database, the smaller they should be.
WebSite	250,000 per site collection	Site collection limits	Note that creating many subsites simultaneously may fail at this limit. Sites have a way of increasing quietly; it is easy to have more than you think.
Site Collection Size	100 GB per site collection	Site collection limits	Site collection size should never exceed 100 GB unless it is the only site collection in a content database.
Documents	30,000,000 per library	List and library limit	This value depends on how well you organize the data in the library, using folders, nested folders, and using views.

TABLE 1.3: Software limitations for SharePoint Foundation 2010 (CONTINUED)

OBJECT	MAXIMUM VALUE	TYPE OF OBJECT LIMIT	DETAILS
Major Versions	400,000	List and library limit	Passing this limit can have surprising performance effects, causing issues with file open, save, delete, and view history.
Items	30,000,000 per list	List and library limit	You can create very large lists using standard views that filter by meta-data and by using folders. This value can also vary depending on number of columns and how the list is used.
Number of SharePoint Groups a User Can Belong To	5,000	Security limit	This is consistent with Active Directory guidelines. Can be affected by size of tokens, the number of groups SharePoint is caching, and how long it takes to do security checks.
Users in a Site Collection	2 million per site collection	Security limit	You can add millions of people to a website using security groups (instead of individual users). The limit particularly affects the manageability and ease of navigation in the user interface.
Active Directory principals/ Users in a SharePoint Group	5,000 per SharePoint group	Security limit	Activities affected by this limit (or exceeding it) are fetching users to validate permissions and rendering the membership of a view.
SharePoint Groups	10,000 per site collection	Security limit	When greater than 10,000, the time it takes to execute operations increases significantly
Security Principal: Size of the Security Scope	5,000 per ACL	Security limit	The number of security principals per access control list for an object. The size of the scope affects the data used for security check calculations. There is no hard limit, but the bigger the scope, the longer the calculation takes.

TABLE 1.3: Software limitations for SharePoint Foundation 2010 (CONTINUED)

OBJECT	MAXIMUM VALUE	TYPE OF OBJECT LIMIT	DETAILS
SharePoint Search Service Applications	20 per farm	Search limit	Multiple servers can run Search. More than 20 (particularly if there are other service applications running on the farm) degrades performance.
Indexed Items (and Crawl Log Entries)	Recommended 10 million, 100 maximum per Search service application	Search limit	Although search can index 100 million items, it is suggested to keep the limit to 10 million for best performance.
Alerts (that can be searched)	1 million per search application	Search limit	This is the tested limit.
Blog Posts	5,000 per site	Blog limit	Tested limit for maximum number of blog posts per site
Blog Comments	1,000 per post	Blog limit	Tested limit for blog comments per post, per site

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 do so. 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, capabilities, limitations, and services. Now you are ready to get started installing it. Chapter 2 covers preinstallation steps, Standalone installation, and post-installation steps; Chapter 3 covers Server Farm Complete installation and runs through the post-installation steps as well.

The Bottom Line

Determine the software and hardware requirements you need for installing SharePoint Foundation 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 Server components and services in order to function.

Master It What software architecture is required for both the server OS and SQL to successfully install and run SharePoint?

Identify the three ways of installing SharePoint Foundation Choose the best three ways of installing SharePoint Foundation 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 SharePoint Foundation allows SharePoint to access data from external data sources?

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 storage requirements that a SharePoint server would need in a given environment?

