1

Introduction to SharePoint

by Mike Walsh

When PCs were new, there were two main computer magazines — *Byte* from McGraw-Hill and *PC Magazine* from Ziff-Davis. The older *Byte* covered not just PCs, but also Apples, and was a tough read in parts, whereas *PC Magazine* concentrated entirely on PCs and was placed at the ability level of most of its readers. Ziff-Davis covered the "professional" market with a magazine called *PC Tech Journal*, and despite hardly being able to understand much of it, I subscribed to this because I thought that in time I would. Two years later, I was just about able to start following the articles, so naturally Ziff-Davis then pulled it and that was the end of that.

If you bought this book without having been responsible in some way for any previous SharePoint product, you are roughly at the level with SharePoint where I was with PC computing more than 20 years ago — that is, you have a good knowledge of computing in general, just no specific experience of the SharePoint products. Because I'm assuming that you don't have a couple of years available to you while things click into place, this first chapter is for you. This chapter provides the basic knowledge you need to be able to follow the rest of the book.

> *This chapter is not for people with existing knowledge of SharePoint products. You have the choice of going directly to Chapter 2 or (and this seems most likely) picking and choosing among the chapters that follow. Some chapters will talk about SharePoint areas that are completely new to you, and others will add to your knowledge of other areas.*

I start by briefly showing you how "SharePoint" became the multifaceted set of applications we have to choose from today. I equally briefly outline the differences between the various applications before showing how to install the simplest SharePoint 2007 version in the simplest way possible, so that you quickly will have something concrete to look at when reading the rest of the chapter.

The rest of the chapter goes through what is installed and is available for your use when you do an installation of Windows SharePoint Services (WSS) 3.0 (that "simplest version").

Looking at the Old and New

Let's start by ensuring that you know how we reached the present complicated set of applications and what these are. There are lots of different (and yet similar) names for the various SharePoint products, so this section should make you are aware of both the old and the new names. By the end of this section, you should know at a glance which SharePoint product and version people are discussing when you see some advice on the Internet.

As shown in Figure 1-1, in 2001, there were three "Microsoft" products of what later became SharePoint 2007. One of these, Content Management Server (CMS), was a product that had been acquired by Microsoft during the acquisition of a company. Another product, SharePoint Portal Server 2001 (SPS 2001), started out in the Office team using the Exchange storage engine. A third product, SharePoint Team Services (STS), was something that started off as an internal piece of code written by members of the FrontPage team as a super-set of Front Page Server Extensions. This piece of code was so useful that its use spread like wildfire throughout Microsoft. At some stage, it was decided to make a Microsoft product out of it.

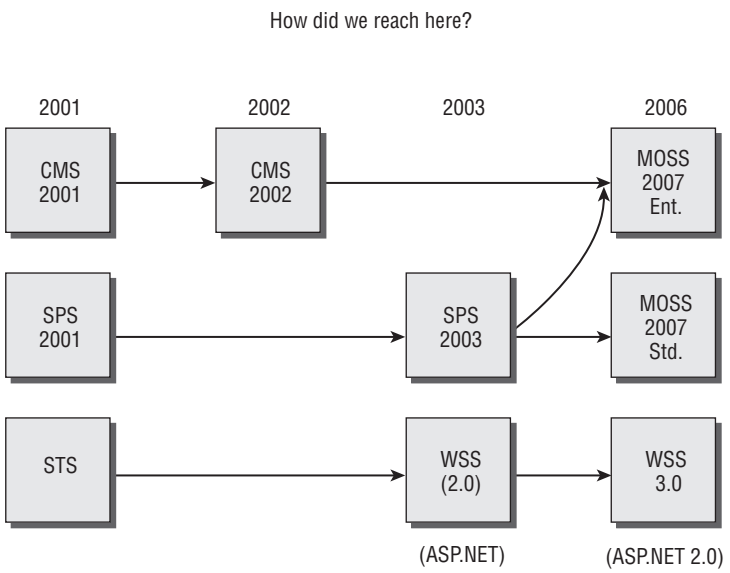

Figure 1-1: Product progression timeline

It's important to note that the use of SharePoint in the product name of STS was purely a marketing one. The product and the techniques it used had nothing in common with the slightly earlier released product SPS 2001.

So, by the end of 2001, there were three completely different products, all of which could be used for a server-based Internet or for an intranet. CMS was the big product, because it typically required a set of computers for staging information from test to production. SPS 2001 didn't have staging, but usually needed a set of servers because its design was based heavily on search, and so it needed servers for indexing and search. STS was, by comparison, a small product that could run on a single server, or a single front-end server with a database server back-end, although it, too, was extensible in a limited way (more front-ends; clustered database system).

The discussions in this chapter use common abbreviations, including the following:

Version 1 (v1) — SharePoint Team Services (STS); SharePoint Portal Server 2001 (SPS 2001)

Version 2 (v2) — Windows SharePoint Services (WSS or WSS 2.0); SharePoint Portal Server 2003 (SPS 2003)

Version 3 (v3) — Windows SharePoint Services 3.0 (WSS 3.0); Microsoft Office SharePoint Server 2007 (MOSS 2007)

Common mistakes are made with the use of SPS 3.0 (which equals MOSS 2007); WSS 2007 (which equals WSS 3.0); and STS (when actually WSS is meant).

In 2002, CMS was refreshed (and became CMS 2002), but it still had nothing to do with the two "SharePoint" products. In October 2003, the new versions of these products came out after major rewrites. These were SPS 2003 and Windows SharePoint Services (WSS) 2.0.

Note that only Microsoft called it WSS 2.0. Most other people called it simply WSS because, after all, it was first called Windows SharePoint Services.

WSS was a rewrite of STS that moved most of the data that had previously been stored in the file system to the database. Also, by using ASP.NET, SharePoint Web Parts became available.

SPS 2003 was a hybrid product. It had two layers: a *WSS layer* (which was virtually identical to that used by WSS itself), and an *SPS layer* (which looked similar to SPS 2001). The big index/search was still there, but it only worked at the SPS layer, and a simpler search was in use for data in the WSS layer. There was much more additional functionality with SPS 2003 that I won't go into here because it is beyond the scope of this chapter. It is important to note, however, that WSS used SQL DB Full Text Search, whereas now WSS 3.0 uses the same search methods as MOSS 2007.

So, by the end of 2003, there were no longer three different product streams. But there weren't only two either — more like two and a half. These were all still providing server-based Internet access or intranetworks.

What was released at the end of 2006 as a "2007" set of products is finally one stream based on ASP.NET 2.0 and WSS 3.0. In this stream, the difference between the various products is the amount of Microsoft-written code that comes with each product. For example, as you pay more, you get more built-in Web Parts, but you will also get additional functionality provided by Microsoft code that does not come in the form of Web Parts.

A *Web Part* is a piece of code that performs a set of functions. Web Parts, once written, can be added to a library (called a *gallery*) in a SharePoint product, and then used in Web pages of that SharePoint product. Writing your own Web Parts is discussed in greater detail in Chapter 7.

For the version 2 (v2) time frame, the people experiencing the most adjustment problems were those who had been running SPS 2001 and had now moved on to SPS 2003 (because, suddenly, they had a WSS layer to learn). The people experiencing the most adjustment problems for version 3 (v3) products are those who are using CMS and CMS 2002 (who now have to learn some completely new techniques).

So, what are the 2007 products and how do they differ?

> *Microsoft considers WSS 3.0 to be an "application" rather than a product, because it doesn't cost anything. I'm going to use "products" for all different SharePoint 2007 packages because I found when writing this chapter that using "applications" instead was very confusing. The main reason for this confusion is that an IIS Web site that used to be called a "virtual server" in the v2 SharePoint products is now called a "Web application" in the v3 products.*

If you have been to a Microsoft SharePoint 2007 presentation, chances are that at least some of what you have been shown is only available in the most expensive SharePoint 2007 product, which is the *Microsoft Office SharePoint Server 2007 Enterprise Edition* (*MOSS 2007 Enterprise Edition*). Most of what you will have been shown is also in the *MOSS 2007 Standard Edition* (a natural upgrade path for both CMS 2002 and SPS 2003 users — although many will find the additional features of the Enterprise Edition irresistible).

Very little of what you have been shown in the Microsoft presentation will be WSS 3.0, although that is a natural path for WSS users, and it provides many functions users have requested without changing in any way the "size" of the product. If anything, there is now more of a gap in total functionality between WSS 3.0 and MOSS 2007 Standard Edition than there was between WSS (2.0) and SPS 2003 because content management functionality has been added to MOSS 2007 that wasn't present in SPS 2003.

Figure 1-2 is probably what you have been shown. Note that the WSS 3.0 "product" includes the inner circle ("Platform Services") used by all the pieces in the outer ring and also includes the "Collaboration" part in the outer ring. All the other parts of the outer ring are in the various versions of MOSS 2007 only. It's a good idea to try to remember that there is both a WSS 3.0 "technology" (inner circle) and a WSS 3.0 "product" ("Collaboration" plus inner ring) because people tend to say WSS 3.0 without indicating which of the two they mean.

However, there are also a couple of other SharePoint products that are irrelevant in the big picture, and are mainly there to give Microsoft a couple of different price points, so I won't do more than just mention them here. These are "Microsoft Office SharePoint Server 2007 for Search" (which is a subset of the main MOSS 2007 product that provides the MOSS 2007 Search scope without the need to pay for CALs for users) and "Microsoft Office SharePoint Server 2007 for Internet" (which is, in effect, the standard MOSS 2007 product already licensed for use on the Internet).

The rest of this chapter concentrates entirely on the small "free" product — Windows SharePoint Services 3.0. Although usually written off by Microsoft speakers (if they mention it at all) as being for departments (by which they mean a handful of people), even this product can scale up and can be used as the basis of a useful Internet/intranet site. However, the main purpose in using it here is to give you a feel for the basic features that are present in all the SharePoint 2007 products without complicating things with a lot of additional features. In essence, you'll be learning to fly on a small plane, rather than on a Jumbo Jet.

Because the aim of this chapter is to get you off the ground as quickly as possible, I'll be using the very simplest installation of WSS 3.0, which installs on a single server and which uses the default "embedded" database system that used to be called "embedded SQL Server 2005 Express (Windows)," but that now (to our relief) is called simply the "Windows Internal Database." Despite the change of name, it's

still a version of SQL Server 2005 Express, but one that, unlike the normal versions, does not have a 4GB limit in database size. However, it does have an additional restriction that it can only be used for WSS 3.0 (and, to be exact, for other "Windows components") databases and also has additional limits on being attached to management tools.

It is possible to upgrade from WSS 2.0 to WSS 3.0. There are three methods, all of which take the full installation (that is, the complete site structure) of WSS 2.0 and upgrade it completely to WSS 3.0. Be grateful that you won't (as new SharePoint users) be doing this, because there are various small differences between sites upgraded in this way and sites in completely new WSS 3.0 installations. (Look at Shane Young's Chapter 14 if you are faced with an upgrade.)

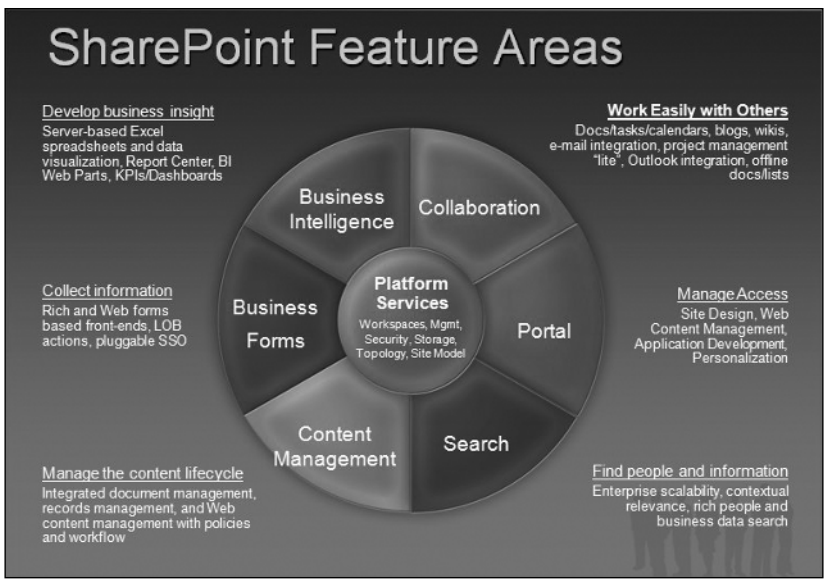

Figure 1-2: A standard view of Microsoft SharePoint 2007

(Diagram: Copyright Microsoft — used by permission)

Installing WSS 3.0

To install WSS 3.0 without problems, you'll need a fresh copy of Windows Server 2003. Whether or not you use the R2 version doesn't matter, but don't make it a Domain Controller (DC). Any version will do, except for Web Edition, so I'd suggest using the Standard Edition. Make sure you have *at least* 512MB memory.

MOSS 2007 will only install if there is a minimum of 1GB or more of memory, and there is very clearly a good reason for that. WSS 3.0 will work faster with more memory, too, but unlike MOSS 2007, it will at least install and work with 512MB. For the small test installation needed for this chapter, that will be enough.

There is a good Microsoft link with complete details of how to install WSS 3.0 on a single server at `http://technet2.microsoft.com/Office/en-us/library/6181fe5b-90ca-40cf-aade-abd59cf3c9071033.mspx?pf=true`*, but the simple instructions in this chapter are all that you should need.*

The first step is to install IIS. The easiest way to do that is to create the Applications Server role in the "Manage Your Server" window (Figure 1-3), which is the first thing you see after you install Windows Server 2003. Select Add or Remove a Role ➪ Next ➪ Custom Configuration. Then, select Applications Server ➪ ASP.NET. (*Do not select FPSE.*)

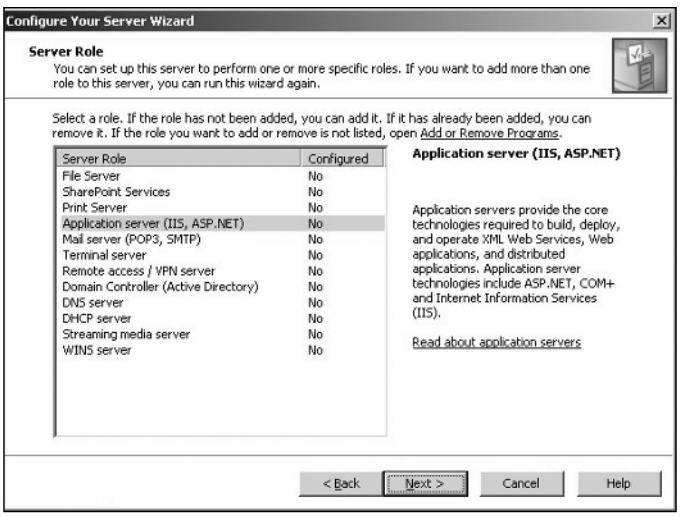

Figure 1-3: Specifying IIS as the Application Server

You'll now see the screen shown in Figure 1-4.

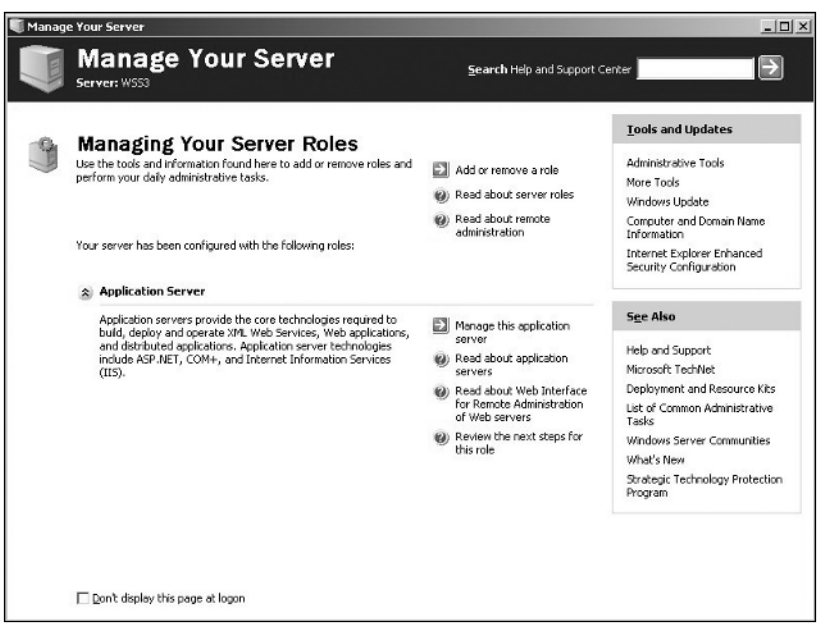

Figure 1-4: Manage Your Server screen

The second thing to do is to install .NET Framework 3.0 from the following link (Figure 1-5):

```
http://go.microsoft.com/fwlink/?LinkID=72322&clcid=0x409
```

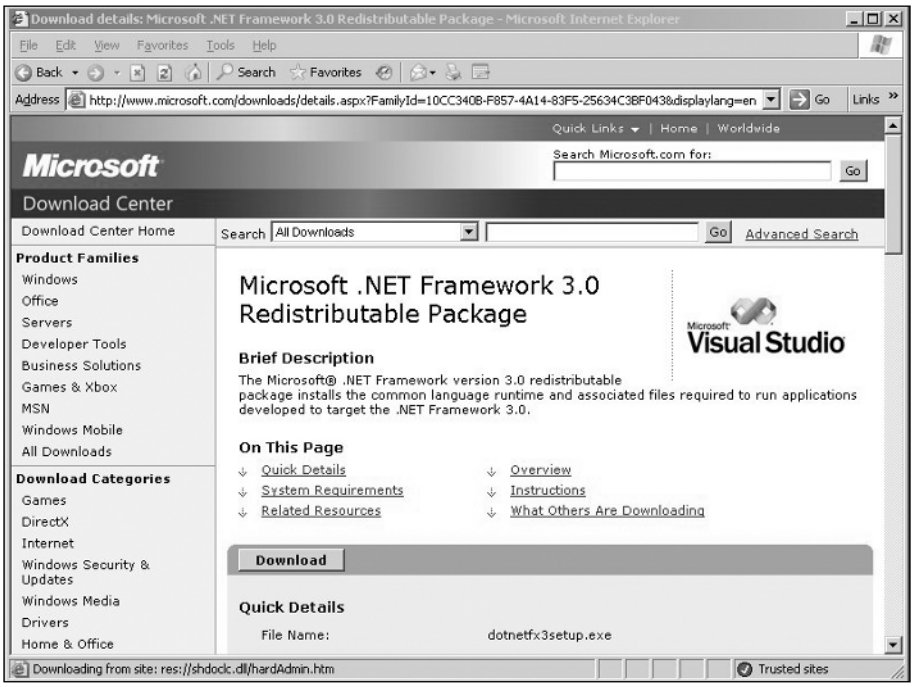

Figure 1-5: Installing .NET Framework 3.0

When this is completed, you must restart before moving on to the final part of this process. This third thing is probably unnecessary, because usually this is the default setting at this point. Just in case, though, go to IIS and ensure that ASP.NET 2.0 is set to Allowed, as shown in Figure 1-6. Just to be 100 percent sure, also go to `C:\Windows\Microsoft.Net\Framework\V2.0.50727` and run `aspnet_regiis -i` to ensure that ASP.NET 2.0 is registered correctly with IIS.

Now you're ready for the WSS installation, which is in two parts. What is called "installation," in fact, only copies the necessary files. The key to this process is the configuration wizard that follows. The wizard takes you through ten steps and performs a lot of work, so don't worry if any step seems to take a while.

WSS 3.0 is available as a free download from the following URL:

```
http://www.microsoft.com/downloads/details.aspx?FamilyID=d51730b5-48fc-4ca2-b454-8d
c2caf93951&DisplayLang=en
```

Run this download. Select the Basic Installation. Then, at the end, select Close and let the configuration wizard do its work.

At the end of all this, you will be asked (R2) to supply a UserName and Password (use the Server Administrator). After a while, you should see a ready-made WSS 3.0 Web site, as shown in Figure 1-7.

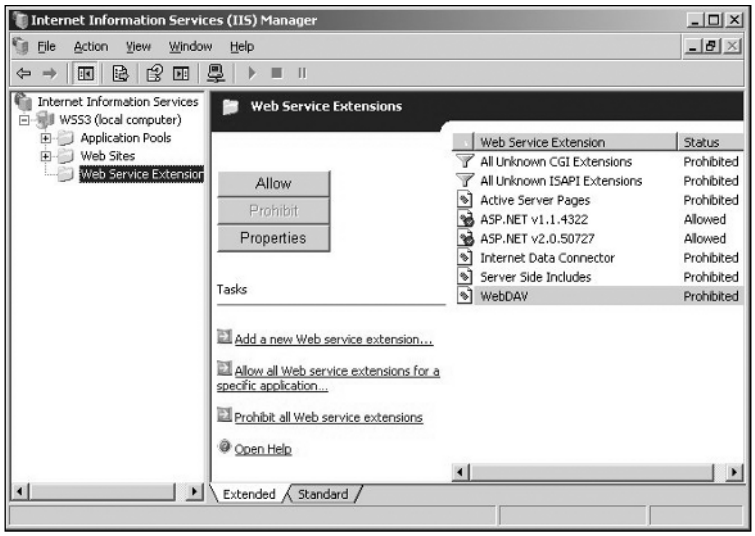

Figure 1-6: Ensuring that ASP.NET is set to Allowed

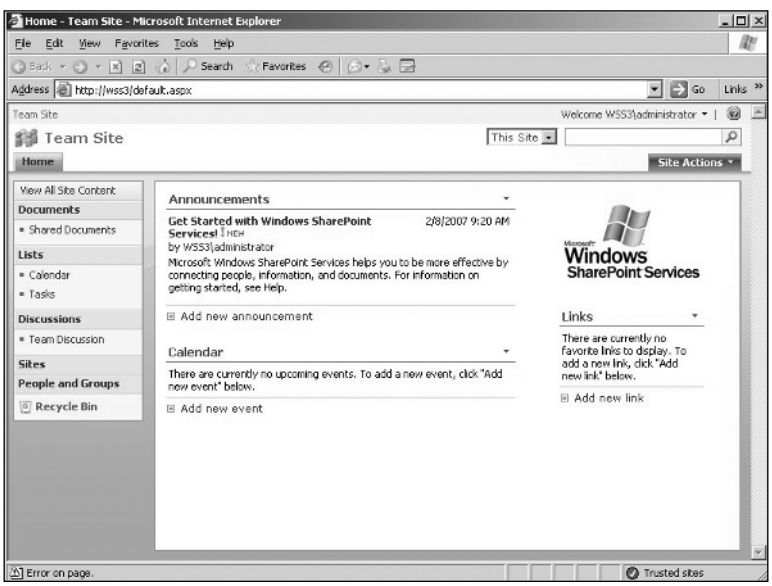

Figure 1-7: WSS 3.0 Web site

My experiences with this installation on a clean machine and performed as described have always been positive. If you do have problems with the configuration wizard phase, first check the Microsoft article referenced earlier to see if it helps you work out what you have missed doing (if anything). If that doesn't help, post a message to the `microsoft.public.sharepoint.setup_and_administration` *newsgroup (at* `msnews.microsoft.com`*) saying that you were doing a WSS 3.0 installation and got the following error at step X of the configuration wizard stage.*

Licensing

Although this copy of WSS 3.0 is being used for test purposes, I should at least mention the licensing requirements. As I mentioned earlier in this chapter, WSS 3.0 is "free." This means that WSS 3.0 can be downloaded for no charge and can be used for no charge. However, to use it, you will need a server running some version of Windows Server 2003 (not the Web Edition) and normal licensing of that Server will apply. (An enterprise license, or a server license plus user CALs, is the usual level.) In addition, if that server is available on the Internet, you will need an "Internet Connection License" (again server-based).

> *The Internet Connection License is connected with the server operating system (Windows 2003) and has nothing specifically to do with WSS 3.0. The situation is different with MOSS 2007, where you need a MOSS-specific Internet license that is much more costly.*

So, WSS 3.0 is free, but in the single-server installation using the built-in (free) database, the server it is running on must be licensed. If a full database system such as SQL Server 2005 is used to store the WSS 3.0 data in (as opposed to the free "embedded" database system), that must be licensed, too.

> *This information is provided as a guideline only. On licensing issues, always contact the local Microsoft office to be sure.*

The Components of the Standard Web Site

Before looking at what is visible on the Home page, let's start off with a few basics.

Libraries and Lists

Information that users add to a site is stored in either *libraries* or *lists*. There are several different types of both provided out-of-the-box, and it is possible to create your own by using a *custom list*. Libraries are actually just a special type of lists. The main distinguishing feature of libraries is that they are used for storing files.

Documents in document libraries were stored in the file system in STS (v1), but since WSS (v2), they have been stored in the database as blobs/images, depending on which terminology you want to use.

I've previously mentioned Web Parts in passing as pieces of code that can be added to Web pages. Each list or library that is available for use in a site has its equivalent Web Part added to the gallery. For example, what you see on the default page for the site just created by the installation of WSS 3.0 called "Announcements" is, in fact, the (equivalent) Web Part of the Announcements list that has been by default added to the page. Click the word "Announcements" and you can see the Announcements list in all its glory.

Another thing that occasionally leads to confusion is the fact that what you have directly after install is not what you must have. If you don't want an Announcements Web Part on that top page, you can remove it. If you want to position it elsewhere on the page, you can move it. But that's less important than the fact that, even though the installation created only one list (plus a Web Part) of each Type (called with two exceptions after the list type — so Announcements, Calendar, and so on), you can create any number of additional lists of the same type, provided you give them different names. Each of these additional lists will then have its own additional corresponding Web Parts.

For example, if you want more than one document library (which is, in my experience, very likely), you can create a second document library called, say, "Final Offers." So, now you will have two document libraries listed on the left pane of the Web page (Shared Documents and Final Offers), and two matching Web Parts. You'll have noted that "Shared Documents" is one of the exceptions to the "rule" that the list created by the installation routine always has the same name as the type of list.

> *All lists and libraries can be shared, including the new Final Offers library. In fact, at this stage, the only difference between the Shared Documents library and the Final Offers library is the name, because both are empty, both have exactly the same look, and both can be accessed in the same way. It is unfortunate that Microsoft chose this name (Shared Documents) because some people assume that this is the only document library that can be shared. It isn't.*

Later in this chapter, you look at how to create a Final Offers library and what you have there. But first, just look at that first page shown in Figure 1-8, in which you'll notice that I already created a Final Offers library from the standard Document Library template. The Final Offers already appears on the left side of the page.

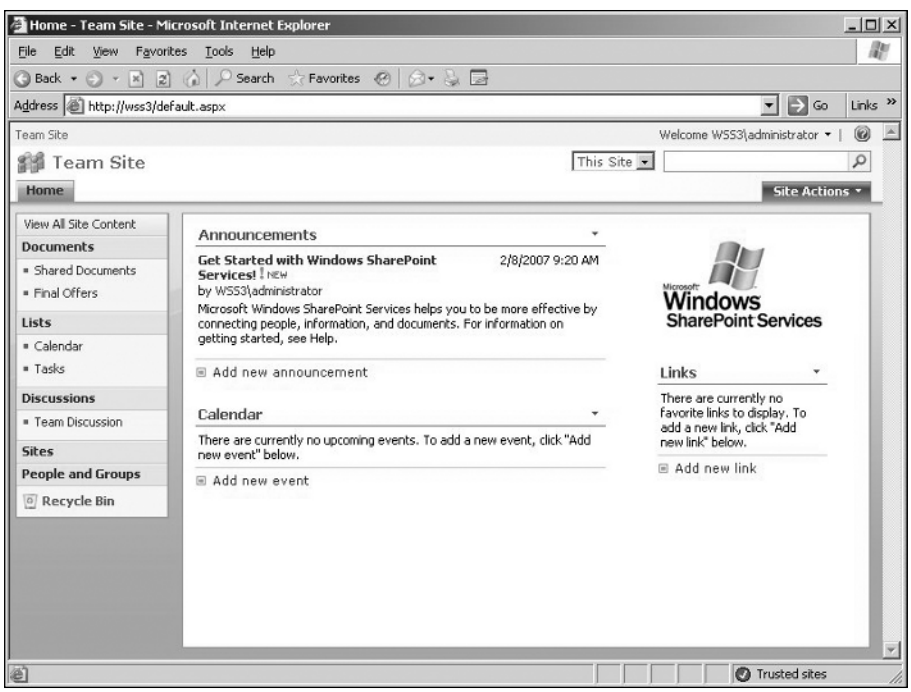

Figure 1-8: Final Offers library

What you see on the left is the "Quick Launch" section. It contains links to five lists — Shared Documents, Final Offers, Calendar, Tasks, and Team Discussion. Notice that the two document libraries are under the heading Documents; Calendar and Tasks are under the heading Lists; and Team Discussions is under the heading Discussions. They are all lists of one kind or another. This is just a way to make this section easier to read.

Web Parts

The main body of the page consists of four Web Parts in two columns. Three of these are the equivalent Web Part for a list (Announcements, Calendar, Links), and the fourth is a Web Part called a *site image Web Part* (which, at the moment, is pointing to the internal address of the WSS logo image).

> *This main body of the page is actually divided into zones. The setup of these "Web zones" is included in the template for the Home page (*`default.aspx`*) with one typical setup being a horizontal zone across the top, with two vertical zones under that, and under them, a final horizontal zone at the bottom. Web Parts can be dragged into any of the zones. However, if any zone has no content added to it, the user looking at the Web page will not be aware of those zones at all. For example, if you only have Web Parts in one vertical zone, the user will only see a single column of data taking up the entire main body of the page. The initial zone arrangement can be amended by accessing the site and page with SharePoint Designer 2007 (SPD 2007).*

Of the three lists that the first three Web Parts represent, only one (Calendar) is also listed in the left-hand column. When creating a list, you can always choose whether it is to be listed in the Quick Launch pane. You decide (at any time) which Web Parts will be used in the main body and where they will be located there (within the current Web zone setup).

If you want all the lists that the installation routine created to be accessible from this top page, then you should have a total of seven Lists (Shared Documents, Calendar, Tasks, Team Discussions, Announcements, Links, and the Final Offers list that was added). Check this by clicking View All Site Content at the top of the left-hand column, and the screen shown in Figure 1-9 appears. (At this stage, your site will still not have the Final Offers list, of course.)

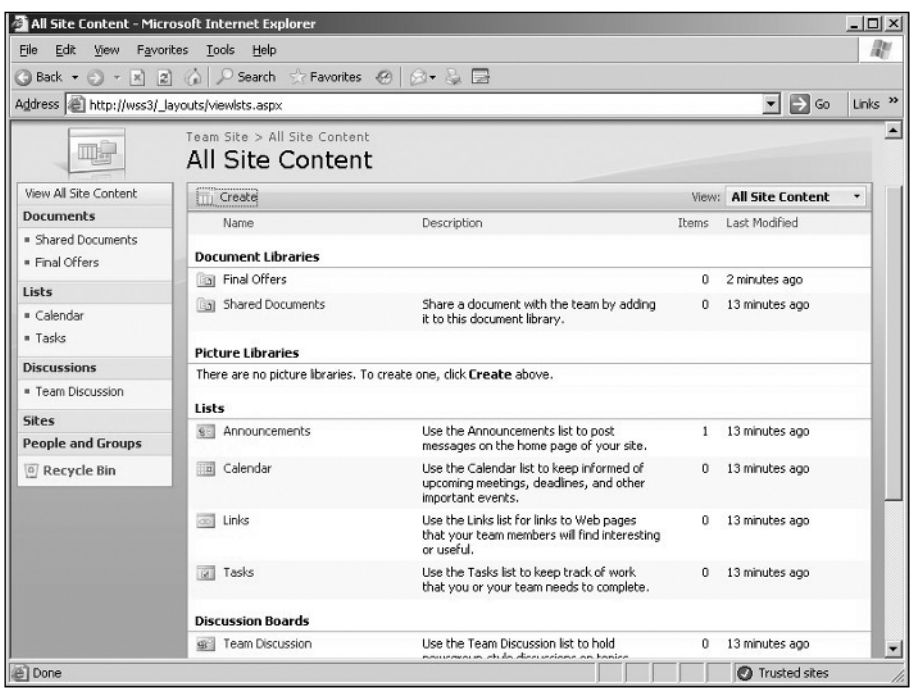

Figure 1-9: Result of clicking View All Site Content

Yes, they are all there (and nothing else).

Within the sections here, the names of the lists are in alphabetical order — so, Final Offers comes before Shared Documents. In the Quick Launch section of the top page, Shared Documents comes before Final Offers. This is because newly added lists are always added to Quick Launch (within the appropriate section) below the existing lists. Typically, the next step after adding a list is to go to go back to the top page and choose Site Actions ➪ Site Settings. You can select (in the "Look and Feel" column) Quick Launch, where one of the options is Change Order, as shown in Figure 1-10. Note, too, that though I tend to use alphabetic sorts in the Quick Launch section, this isn't necessary — you can have whatever order you want, and can vary this from section to section.

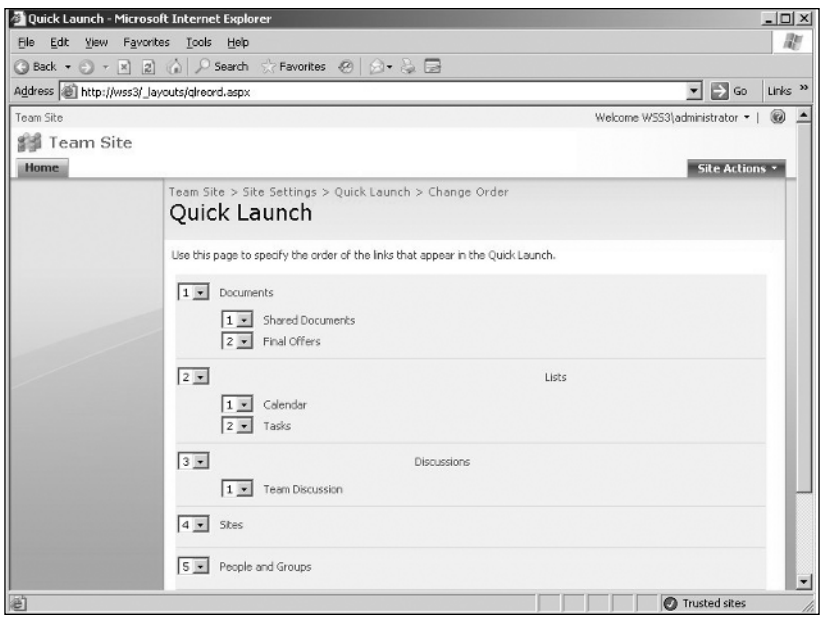

Figure 1-10: Changing the order in Quick Launch

If you don't want a Quick Launch section at all, you need to select Tree View in the same "Look and Feel" column. I mention that here because it isn't exactly an obvious choice for removing Quick Launch!

Options Within Lists

Before you can look further at the Web Parts on that top page, you'll need to first of all have a basic knowledge of what options you have for lists. I'll look at a document library, but the principles apply to other types of lists, too. To do this, you'll need to create a Final Offers list, too.

Select Site Actions ➪ Create and Document Library (top left). Just type **Final Offers** in the Name field and leave all the other options as they are.

Now, you should have the same site I've been showing you already. Follow the earlier instructions to resort the Quick Launch so that Final Offers is listed before Shared Documents.

Click Final Offers and you'll see nothing much. So, first upload a few documents — preferably of different types such as a Word document, a text document, and a PDF document. If you have Office 2003 Pro or Office 2007 installed on your client, you'll be able to use the "Upload Multiple Documents" feature. If you don't (or if you are using a browser such as Firefox that doesn't support ActiveX), you won't see this at all, and you'll be forced to upload documents one at a time if you are using the standard Upload Document page.

> *There are other ways to get bulk data into document libraries, but in this overview, I'll stick to using the WSS 3.0 User Interface. There's a useful list of alternatives for doing bulk uploads in the "Tutorials" section of the WSSFAQ part of the WSS FAQ site (*www.wssfaq.com*) These still apply for the WSS 3.0 product.*

For a few documents, the list you see is fine, but what if you have a few hundred documents? That's where WSS 3.0 offers two main alternatives to either just have long Web pages or a number of Web pages of, say, 100 documents at a time (which, not by coincidence, is the default value). These two alternatives are Views and Folders.

Before you move on, add a few more documents to the document library so that you have between 10 and 20 documents there. With six documents or less, you won't be able to see the point of some of the discussion that follows. You should now see a screen similar to Figure 1-11.

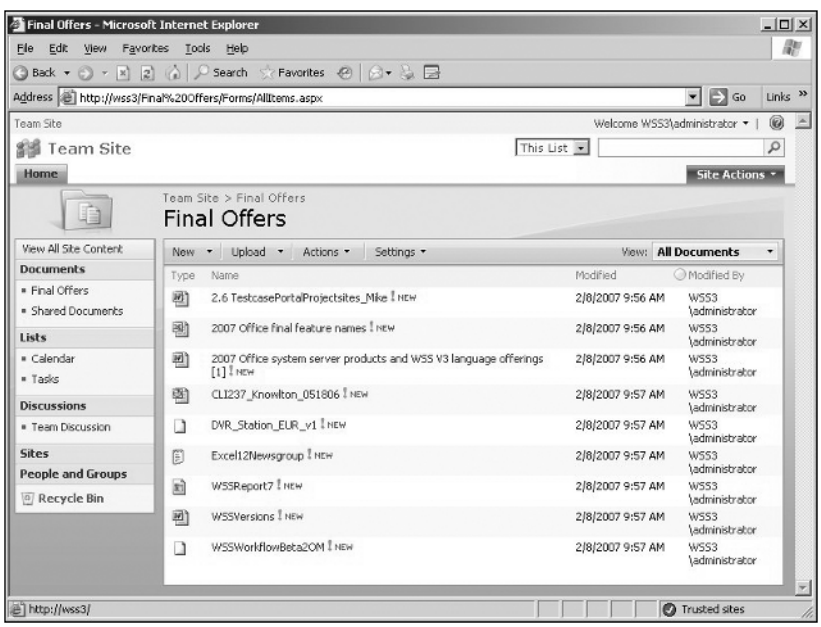

Figure 1-11: Result of adding documents to the document library

Views

A *view* is a selection of documents based on specified criteria. At the moment, the only view is the All Documents view (look at the right-hand edge of the horizontal menu line that appears at the top of the main pane in the Web page). This view, which was created by the installation routine, shows a list of all

the documents with a maximum of 100 documents (default value) per Web page. If there are more than 100, there will be a Next button so that the user can access the next 100.

Just because the name of the view is All Documents, it doesn't necessarily mean that it lists all documents. You could change the definition, but in normal cases, it's a good idea to leave this view as it is and create new ones if you want fewer than all the documents listed.

You could create a new view now, but if you did so at the moment, all that you could do would be create a view that selected on the existing information you have about those documents (name, date added, who by, and so on), and that won't be too useful to you. So, first add a column to the document library and then use that column for selection purposes.

Select Settings (in the horizontal menu line that appears at the top of the main pane in the Web page) and Create Column. Give it the name "Document Source" and specify that the field is of Type Choice. Fill in four values (HP, IBM, Microsoft, Other) and make the default value Microsoft (use copy/paste to get the name right!).

Site Columns

If there is already a suitable site column for your needs, it's advisable to use this (Settings ➪ Document Library Settings ➪ Add from existing site columns) instead of creating your own new column. If there isn't (which is the case here), first create a site column and then use that site column in your document library. This is especially advisable in the case of a field of type Choice because you must only write the list of alternatives once, and the list is then available wherever that site column is used. What's more, when you need to add, say, Adobe and Sun to the list, you add those names once, and they are immediately available to all document libraries in which you are using the (site) column. (This repeated use is quite likely in this particular case and less likely in others.) This would not be the case if you had specified a new (normal) column for each document library; then you'd have to add Adobe and Sun to the column in each document library that used it.

To create a new site column, go to Site Actions ➪ Site Settings and under Galleries click "site columns." Check the existing site columns just in case. There's a site column called Category that looks promising (it was the name I used for views in WSS 2.0), but you see that it's a "single line of text" type, so you reject it. (Experience shows that allowing users to write the name will give you "HP," "H-P," and "Hewlett-Packard" at the very least.) Instead, you select Create, and do exactly as previously described for a normal column.

Give it a name "Document Source" and specify that the field is of Type Choice. Fill in the four values (HP, IBM, Microsoft, Other) and make the default value Microsoft. I usually select "Allow Fill-in choices" even if this means checking the document libraries occasionally to see if a non-standard name has been used. The other alternative is to not allow fill-in choices and hope the users will contact you with a request for an addition to the list of choices (rather than them using the wrong one).

By the way, try to avoid using a name for a normal column that is the name of a site column. This is possible, but can only lead to confusion. Similarly, try to avoid using the same name for a new site column as for an existing site column. Once people know that Category is a single line of text column, there's no point in you creating a site column called Category that isn't.

This is the quick-and-dirty method, which adds a column only for this one document library. If you want to use the same column in another document library using this method, you must add it again and re-specify those four values.

However, there is another option (new to WSS 3.0) that allows you more flexibility and, over the course of working with the site, probably time savings for the administrator. This is the use of *site columns*, which are a set of ready-made names and definitions that can be used (when adding a column) throughout a site.

Creating a View

After all that preparation you are now ready to create a view.

Go again to Settings and this time use "Create a View." Select Standard View and call the view "HP." Unselect "Document Source" from the columns section. Sort by "Name" and, finally, in the Filter section, specify that Document Source is equal to HP. Click OK.

Repeat these steps for IBM, Microsoft, and Other.

These are views that you are going to use when you access the document library page. To make this sensible, add some more documents and then, if your client allows you to do so (Office 2003 upward and Internet Explorer 5.5 upward are required), open Final Offers in Datasheet Mode (Actions) and add random HP/IBM and so on values (by using copy/paste). If you don't have Office 2003 or Office 2007, use the drop-down at the right of each name and select Edit Properties. If you are using Firefox, add an Edit icon to each row of the Final Offers library and use that to edit the item by adding a value to the Document Source field. (Make sure you are using the All Documents view when you do this, as shown in Figure 1-12.)

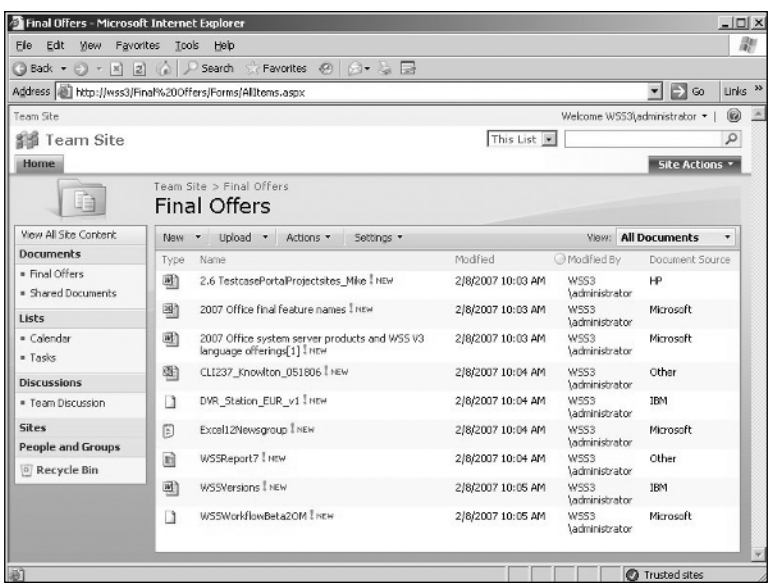

Figure 1-12: All Documents view

Now that you have five sensible views for use in the Final Offers library, you create one additional view that will be used for the Web Part of the Final Offers library.

Again, create a view, only this time call it WebPartView. This time, remove Type, Modified By, and Document Source from the list of Columns. Sort according to Modified (*descending*) and define a second sort by Name. Change the Item Limit from the default 100 to 6. Click OK.

> *When you open the list/library, possible views will be listed in a drop-down at the top-right of the screen with the name shown for the view being used for the display.*

Amending the Home Page

Now, go back to Home to start investigating what you can do with Web Parts on such a page.

In Site Actions, select Edit Page. Have a look at what you can see by clicking the various edit buttons (for Announcements, Calendar, and so on), but don't change anything.

The first thing you're going to do is to add the Web Part of the Final Offers library to the page.

Click the "Add a Web Part" link in the left pane and you'll see a long list of the Web Parts that are available to you. Select Final Offers and Add.

What you'll now see is a very large Final Offers section (with maybe 10 to 20 documents) at the top of the center column that dominates the page. Avoiding this domination is the reason you created a special view called WebPartView. Select "edit" at the top-right of the Final Offers block and then Modify Shared Web Part (Figure 1-13).

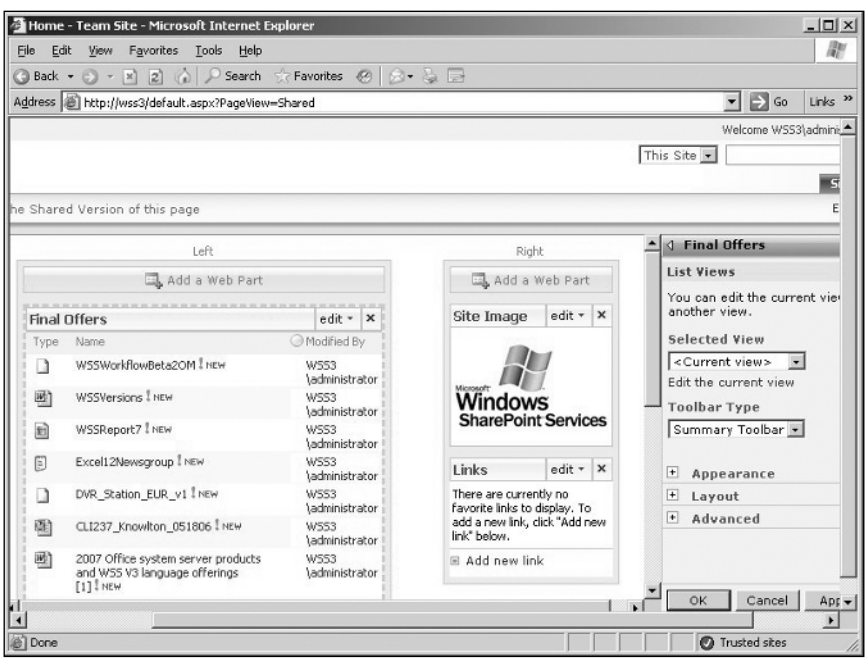

Figure 1-13: Editing the view

The present Selected View is <Current view>. It's a pity, but that's what it always shows here. In this case, the Current view is the default view that was available when the installation routine added this Web Part to the Home page. Here it's the All Documents view without the Modified field.

Yes, this is odd. Logically, it ought to be identical to the All Documents view, as was the case with WSS (2.0).

Use the drop-down under the heading Selected View, select WebPartView, and click OK (or Apply). You'll notice that the Final Offers Web Part no longer dominates the Home page — there are fewer fields and only six documents listed.

If you think that you are never going to want to change the view that will be used for this Web Part on this page (and never need it anywhere else), you can delete the WebPartView. The Current view has already been set for this Web Part and the Web Part retains the settings of the view even if the view is deleted. But do this only if you are completely sure and the listing of WebPartView in the drop-down list offends you; otherwise, you will need to re-create the view if you want a different version of it for the Web Part on this page.

Now is perhaps the time to note that you can add the Final Offers Web Part to the Home page twice (or more times) if you want. Both Web Parts are, however, still a representation of the same document library. In other words, if you use the "Add new document" feature of one of those Web Parts to add a document, that document is, in fact, added to the *document library* (not the Web Part) and, thus, will appear in *both* Web Parts, even though you only added it to one.

This is a simplification. Whether the added document will be listed in both Web Parts depends on the views being used in the two Web Parts. But, assuming that there is no reason for it not appearing (for example, that the document isn't filtered out in the view being used in one of the Web Parts), then it will appear in both. What is certain is that it is present in the document library from which both Web Parts get their information.

Typically, there is no point in having two Web Parts on a page that represent the same list/library, but there is one case where this is useful, and I'll cover that soon when explaining what Web Part connections are.

But first, let's get back to lists.

The main alternative to using views to divide information into useful chunks is to use folders.

Folders

Just as in a file system, you can create folders within a document library and you can create subfolders below them, and so on. The main problem with using folders is that they encourage bad habits.

One example is the temptation to copy your existing file system in bulk to a document library when you first install WSS 3.0 or MOSS 2007. Some Microsoft people (typically not those from the SharePoint team) have read up a little on SharePoint and have noticed that the recommended limits for the number of documents you can store in a single document library are 2,000 documents per folder, and 2,000 folders per document library. They then make the mistake of thinking that this means that a SharePoint system is a solution to the age-old problem of how to move files away from the file system — simply copy the file structure of your entire common disk to SharePoint. *Well, don't do it!*

The problems include the fact that there is an overhead of up to 80 percent in the disk space required to store a file as an image in the database compared to storing it in the file system. There's also the fact (compounded by that size increase) that backup processes for SharePoint systems are by no means as simple as those for files in file systems, and they take longer (much longer). Finally, four million documents per document library (if you use folders) is by no means a figure to aim at, and is much more of an "our system can handle XXX" boasting point of little value in the real world.

The real-world recommended process is that you use SharePoint 2007 document libraries for new information, rather than as a thoughtless dumping ground for old information. By all means, transfer key information that is still of considerable value from your file system to a SharePoint 2007 document library. But do this only after a major study that both evaluates the data you already have and specifies a *set of* document libraries to handle it. You'll need to do such an evaluation in any case when creating a set of document libraries for storing your new information, and your old information will give you good guidelines on which document libraries (and sub-divisions of document libraries) you will need.

> *Anyone who wants to read a semi-official Microsoft take on this (from within the SharePoint team) should have a look here:*
>
> `http://blogs.msdn.com/sharepoint/archive/2007/01/02/is-the-file-server-dead.aspx`.

As for the choice of using folders or using views, my own preference is for a number of document libraries to cover completely different topic areas to use views rather than folders to cover the different pieces of data in sections (and sub-sections) within each topic area.

Using again the HP, IBM, Microsoft example, it could be that you have a document library for Quotes and a second document library for Technical Specifications (and later a third for Contracts, perhaps). Though you could keep the relevant information in each document library in HP, IBM, Microsoft folders, views have the advantage that they can be used when specifying the look of a Web Part on a page. So, for example, you could have a Quotes page where there are three Web Parts displaying information from the Quotes Document Library, with one part showing only HP quotes, one IBM quotes, and one Microsoft quotes. You can't do that with folders. There are also other slightly annoying "features" of the way folders are implemented that I'll leave for you to discover.

On the other hand, it is possible to specify that only certain people can access a folder, which isn't possible for a view. So, perhaps it's time for a quick overview of security.

Security

Security in SharePoint is composed of two things: who can access what, and with what rights they can access.

A user can be given rights (through Create User) to access a site/sub-site/document library or list/folder or item. Typically, the user is given rights to access a site or sub-site, and this automatically means that the user has the right to access everything in the site. But in special circumstances (for example, a document library contains personnel information), access rights to a document library/list and to a folder or even an item can be restricted if needed. You cannot give special access rights to a view. Anyone who can access the document library or list can access any view.

Specifying access rights for items in a list/library, even though it is now possible in WSS 3.0 and MOSS 2007, is a pain, and should only be done if really necessary. If the items can be grouped in any way at all, put each group in its own document library/list or folder, and specify access rights for the group of items.

Specifying access rights for folders would perhaps be necessary in similar circumstances (such as for document libraries), but where your design of the site was based on a small number of document libraries plus folders, rather than on a large(r) number of document libraries (with views).

When you create a sub-site, you can specify that it inherits the access rights of the site below which it is being created. This is the default value when creating a sub-site. The value can be changed at a later date, but it is better to think out whether special access rights are needed when doing the design of the SharePoint site structure.

Permission Level

"Permission Level" is the new name for what used to be called a Site Group. You are likely to still see both names being used, but I will use the new name throughout this chapter because it is at least no more confusing than the earlier name.

Each user when being "created" (that is, given rights to access a site and so on) is assigned to one or more (typically only one) *permission levels*. This is a somewhat confusing name for what is just a collection of rights (for example, the right to add information, the right to delete, and so on). There is a long list of rights and the four ready-made permission levels (Reader, Contributor, Site Designer, and Administrator) that Microsoft provides out-of-the-box. Each contains a sub-group of selected rights (logically) as follows:

- A Contributor has the rights of a Reader and more
- A Site Designer has the rights of a Contributor and more
- An Administrator has all rights

This ever-increasing number of rights in the default permission levels is the reason why typically only one permission level is selected when a user is created (that is, a user is given access rights to site/sub-site/list, and so on).

These built-in permission levels are useful in most cases, but a typical request is, "How do I give a user Add rights, but not the right to Delete?"

One method would be to amend the rights specified for the Contributor permission level by removing the cross for Delete. This is, however, not a good idea, because it means that everyone who is a member of the Contributor permission level can from now on only add items and not delete them. So, the people who are allowed to delete items will need to become members of the Site Designer permission level, which will give them too many rights.

Instead, you would create a new permission level with a suitable name — AddOnlyContributor wouldn't be catchy, but would describe the functionality well — and give it the same rights that the Contributor permission level has, minus the Delete right.

It's wise to create a new permission level slightly different from that of the standard Contributor permission level, even if you never intend to use the standard list of rights of the default Contributor permission level. This is because people coming later to administrate the site who have experience from other SharePoint sites will automatically assume that the Contributor permission level has the standard, default rights assigned to it.

In time, as you build up a set of new permission levels, it is probable that there will no longer be the strict hierarchy of permission levels there was at the beginning. This is the reason why, when giving a user access rights to a site (via "Create User") or specifying that a user has rights to access, say, a list, there is the option of selecting one or more rights.

For example, suppose that you wanted only Fred, Jim, and Bert to be able to access a document library. The seemingly easiest way would be to access the document library, select "Modify Settings and Columns" and "Change Permissions for this Document Library," then remove all the existing people/groups with access rights to the library. Then, one by one add Fred, Jim, and Bert. A smarter way, however (and one allowing more flexibility in the future), would be to create a new permission level especially for access to this document library. Fred, Jim, and Bert would, as usual, be created at the site level, but, in addition to being assigned to the normal Contributor permission level, they would also be assigned to this special permission level.

In this case, when changing permissions for the document library, and after all other users have been removed, this special permission level would be specified in the list of what is allowed to access the library. The benefits naturally are that when Wanda also needs to be able to access the document library (and there are possibly other places this group of users needs restricted access to), she only needs (at Site ➪ Create User level) to be made a member of the additional special permission level (called, perhaps, PersonnelContributors).

All of these explanations talk about users. Typically, in a domain environment, users are members of an Active Directory (AD) group and AD groups (rather than individual users) would be "created" using the Create User function. Thus, these AD groups could be used in the previously described document library restricted access case. However, the permission level approach used here for restricted access to a document library can still be valid, even when using only AD groups, because perhaps more than one AD group needs to be given rights. As with many things in working with SharePoint systems, there are several options, and it might be just that you need to ask someone else to create a new AD group, but you can create a new permission level yourself and, thus, using the permission level method is a faster way for you to solve the problem.

Anonymous Access

One thing that hasn't been mentioned so far is the possibility of having anonymous access to a site.

Typically, this would be an information site open to the Internet where there is no need to keep track of exactly who is accessing the site or, indeed, of who is adding to the site, because usually such sites don't allow anyone other than a few key people to add items to the site. Another of the uses I have come across and recommend is having the Home page specified as anonymous access, while having the site structure below it to be restricted access. This Home page would serve as an advertising page that encourages people to want to be given access rights to the real sites.

When specifying anonymous access for a site, it is important to note that this is a two-step process. First, anonymous access must be made possible by specifying this in IIS. Then (and only then) you can specify anonymous access for a particular site.

Another thing I've not mentioned is the scenario where the user is not a member of a domain (or, indeed, in the test case here, if the server is in a workgroup, not a domain). Then you could use local (on the WSS server) users.

> *To specify that a local user has access rights to the WSS sites, that local user must first be created (Start ➪ Programs ➪ Administration Tools ➪ Computer Management/ Local Users and Groups ➪ Users ➪ right-click + "New User" and so on) and given a permanent password that cannot be changed. This local user is then assigned (in WSS 3.0) access rights to a WSS site via Create User — that is, in the case here, two steps rather than the one (Create User) step required when adding an* AD *user or group.*
>
> *(Typically I use local users called Reader password Reader, Contributor password Contributor, and Designer password Designer on my test sites.)*

Specifying users local to the server should only be done when you have a single front-end. If you have load-balanced front-ends, there will be potential problems because of the need to keep those front-ends identical. In other words, those local users must be added to all front-end servers. The main problem with local users in a single-server situation is that the password must be permanent because the users don't have the rights to the server to allow them to change it, and this is a major security problem.

There is one last important thing about access rights. In the v2 version of WSS, if you could access a site/page and so on, you saw all the links on that page, even if they led to pages that you have no rights to access. If you then clicked those links, you were asked three times for your name and password (which was to give you the opportunity of using a different name with better rights) before being given an error message. In the v3 products, you see only what you can access.

You still might come across the "three times and you are out" syndrome, however, because you might have the rights to *access* a document library (that is, read), but not the rights to save anything to it. In this case, again you will be asked three times for "better" authorization, and then rejected.

The easiest way to see this "lesser links for lesser rights" working is to log in as a different user and then access Site Settings, first as Administrator and then as a Contributor. Your options will have shrunk.

Finally, here are three common access questions and answers:

Q: *Why isn't Create User working? I've specified a valid domain and username.*

A: In order for Create User to work in this case, it must be possible to access a Domain Controller (DC) of the domain in question. Lack of access could mean that the DC is down at the time of access. But a more normal case is that there is no trust relationship between the domain containing the WSS server and the domain of the user that is not being found.

Q: *I've given "All Domain Users" rights to access my site, but they still need to log on. How do I avoid this?*

A: In the client, specify for the security zone in which the WSS site is located (in Internet Explorer, go to Tools ➪ Internet Options ➪ Security ➪ Custom Level and then to the final User Authorization part). Go to the Logon section and select "Automatic Logon with current username and password."

(This will work, provided the user is logging on to a client with the same domain\username that has been given access rights to the site.)

Q: *After making this change, my users can access the site without the need for logon, but they are requested to log on when opening a Word document. How do I avoid this?*

A: Ensure that the WSS site is in the Trusted Sites zone in your Internet Explorer. If this doesn't work, add the WSS site to the Intranet zone.

The Home Page — Again

Now that the security aspect has been briefly covered, let's go back to the Home page for a few tips on the Web Parts you see there.

The Announcements Web Part is almost always something that you should use as a means of giving the latest information about the site to your users. Use it sparingly so that people will notice it when it's important that they do so.

One thing to watch out for is the standard view (Current view) that is used in the Web Part. This view of the Announcements Web Part is a special view that should not be changed. If you do change it by mistake, recover by adding a new copy of the Announcements Web Part to the Home page and deleting the original one.

The Links list is an odd one. Although there are separate entry fields for URL and Description when you create a new Links item, you'll notice that when you are in a view (such as the default All Links view) and select the URL field, what will be visible in the list will be the Description (which, when clicked, goes to the URL). If you don't have a description, you will see the URL itself, but only then.

Because URL is the only field available to you in a view (and this is shown as a description text linked to the URL), a view can only be sorted by Description (what is visible) and not by the URL itself. If you know that, for some reason, you must sort according to the actual URL (for example, you need to keep track of the numbers of Knowledge Base articles), then create a new text field called `URLTxt` and copy the URL to that field as well as to the URL field when you create a new item. Similarly, if you see the need to have the text of the Description field available as a pure text field, then add a second new field called `Desc` and add the text of the Description to that field, as well as to the Description field, when creating a new item.

Both these actions will give you more flexibility at the cost of additional manual actions every time you add a new Links list item.

Whereas you will usually have only one Announcements list (and Web Part), you can have as many Links lists as you like — all, of course, with more specific names than "Links," and, of course, none of them need to include the word "Links" in the name.

There are restrictions in the kinds of URL that you can specify in this field and, most importantly, that "test this link" will accept links that the upload function will not accept. So, `http://` *links are fine, provided they are absolute URLs. WSS 3.0 will not accept relative URLs, even though the "test this link" function will, and WSS 3.0 also requires that the whole path is not longer than 260 characters, which again "test this link" has no problems with. Similarly, "*`file:///`*" will work in WSS 3.0, but things like* `ftp://` *will not, even though "test this link" has no problems with them.*

Looking further at the Home page, you'll notice the WSS logo taking up rather a lot of space in the right-hand column. You'll find that it's wise to get rid of this because this space could be used for much more important things (such as useful data). I just close the Web Part so that it's no longer visible, but still available in case I want to resurrect it (even if only to give it a new image). One thing that you must constantly be aware of is that when putting images on your Web pages via this kind of Web Part, make quite sure that all the people accessing the Web page have the rights to see the image. One way to ensure this is to have an image library in a site that has anonymous access, and upload all your site images to that.

Moving on, you'll see that the Calendar is both linked to in the Quick Launch section and is a Web Part in the main section of the page. It's a matter of opinion whether this duplication is a good idea. I tend to have everything listed in Quick Launch, whether or not it appears in the main body. One reason is that if you use the space in the main body of the page, these are important enough for the duplication not to be a problem. Another reason is that I can, at any time, re-adjust what's visible in the main body of the page without the need to fiddle around with adding something to Quick Launch (which requires opening the list and specifying "add to Quick Launch").

However, the main reason for adding everything to the Quick Launch section as well as having them available on the page itself is that the Quick Launch section in WSS 3.0 sites now is repeated on pages lower in the structure (that is, even on pages that don't have that particular Web Part showing on the page), whereas in WSS 2.0 it was only available in the Home page of a site (`default.aspx`). Just make sure the length of the Quick Launch list is not excessive to avoid it dominating all the subsequent pages.

The Calendar itself is of no use for an information-providing Web site of the usual kind because it provides links to documents or Web pages, or even to text information in lists, and dates are of no relevance. Thus, often you can just close the Web Part and leave some free space for something else.

But there will be circumstances where having a Calendar in your face every time you access the Home page will be very useful. Think, for example, of the internal Web site of a company. It would be very useful if there were a calendar showing where all the salesmen were (and were going to be) at any one time. At the very least, it would avoid the disaster scenario of two sales representatives appearing on successive days at the site of a potential customer, or of two support people needing to re-boot the same set of servers at slightly different times on the same day.

> *When thinking about the Calendar Web Part, it's easy to start thinking about using an Outlook calendar on a Web site. This is addressed later in this chapter in a discussion of the integration between WSS 3.0 and the various components of Office 2007 (such as Outlook 2007). Unfortunately, what you'll see there is that in Outlook 2007 you will have the Outlook calendar plus one or more SharePoint calendars.*

One link that is only included in the Quick Launch section and is not in the main body of the page is *General Discussions*. Like Shared Documents, this is misleading. This is an example of creating a list of type Discussions and giving it the name "General Discussions." It is not the only place you can have discussions among yourselves. You can quite easily set up a number of Discussion lists (which, typically, you would give rather more meaningful and specific names than "General Discussions" such as "Outlook Questions," perhaps, if you are going to use Discussions as a way to list and answer user questions).

One reason why you might well have a large number of Discussion lists could be that SharePoint Discussion lists are not of major league/premier division (pick your own sporting metaphor) quality. If you imagine that a Discussion list is like Lotus Notes with many years of development behind it, think again. Discussion lists are a simple way of providing threads of questions, answers, and comments that

are little changed in the essentials compared to when they were part of STS back in 2001. So, use them by all means (and I used them in STS, despite their failings for all the questions involved in a major migration exercise), but be aware of their weaknesses and try to use many different Discussion lists to keep each set of discussions relatively small and focused.

There is one final tip on lists and libraries for non-U.S.-based people. The U.S. uses MM/DD/YYYY for the date, whereas most of the rest of the world uses DD.MM.YYYY. It's not easy to find where to change this if you have an installation that is showing you MM/DD/YYYY because you'll be looking for something to do with "date" or "date format" and failing to find it. In fact, the only way, it seems, to change this is to choose (in Site Settings) a Regional Setting (Figure 1-14) that is for a country that has DD.MM.YYYY as standard. This means that you must be careful, because changing the Regional Setting will cause other things to change, too — the 24-hour clock might be standard and you might have a different sort order (both of which can be corrected on the Regional Settings page) and maybe (and much worse) Regional settings mean that you have a decimal comma, not decimal point, which will throw off all your formulae. However, for example, choosing "English(UK)" will probably be a safe way to get a date format change without anything nasty happening.

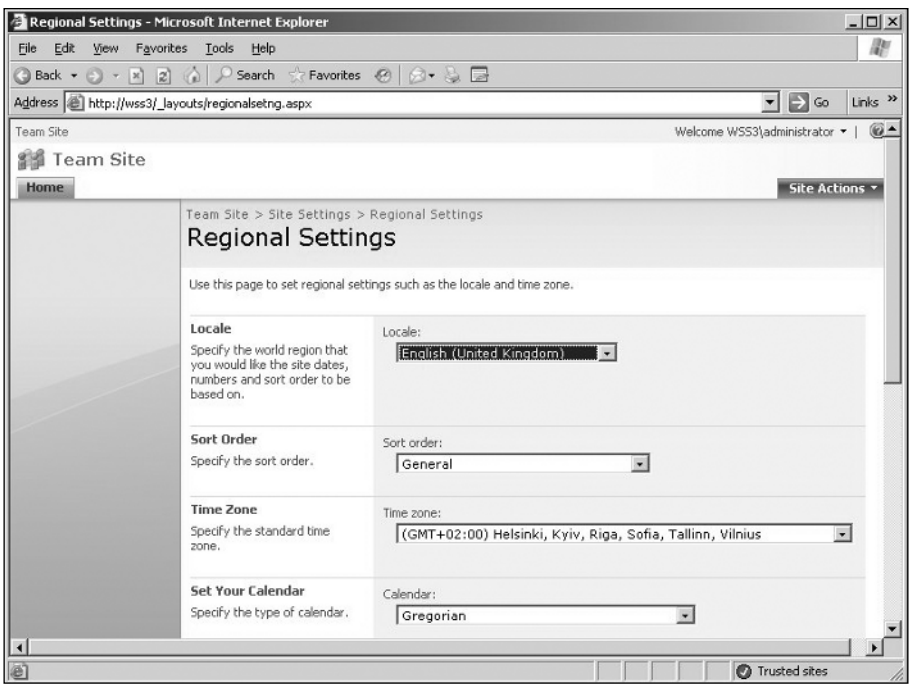

Figure 1-14: Regional Settings screen

Most locations allow you to choose between a 12- and a 24-hour clock, but some don't. This is a "design decision" no doubt, but one that seems overkill to me — I hate Big Brother restricting my options in this way.

That's enough about the lists and libraries. I'll now look briefly at Web Part connections before closing this Home page section by having a look at a few nice improvements compared to earlier versions.

Web Part Connections

Provided two Web Parts on a page have a common field (which may or may not have the same name, but ought to be the same kind of field), they can be linked together. The procedure that uses the drop-down for a Web Part when you specify that you want to Edit the Page is straightforward enough, so I won't go into the details of it, but instead discuss a couple of cases where you might use this.

Let's say you had a picture library containing images of all your business contacts. This would be just a standard picture library with an extra text field for Full Name. You put the Web Part of this picture library on the page so that it shows just one image. You then have a Custom List with company name, department, and fullname. You put a Web Part of the Custom List on the page that consists only of the Full Names from that company. You connect that Web Part via Web Part connections to the picture library using the two Full Name/fullname fields to connect. Then, when the user clicks fullname in the Web Part of the Custom List, the image in the Web Part of the picture library will change.

That's the simple one-to-one version of Web Part connections, but you can chain them, too. In this case, you have three Web Parts of the *same* Custom List on the page, along with a single Web Part of the picture library.

Custom List Web Part 1 (WP1) shows only the companies. Custom List Web Part 2 (WP2) shows only the departments. Custom List Web Part 3 (WP3) shows only the full names. You can connect WP1 to WP2 (using the Company field), WP2 to WP3 (using the Departments field), and, finally, as mentioned previously, WP3 to the Web Part of the picture library (using the Full Name field).

> *Note that this is an example of the rare case where using the Web Part of the same list more than once on a page can be beneficial.*

Those are the principles of Web Part connections. You need to experiment with when and how they will be useful to you. They can, of course, be used on any pages where there are Web Parts, not just on the Home page.

A Few Nice Improvements

The way to see the first improvement to the Home page is to open the Final Offers library. You'll see it in the section above the main body of the screen — Home ➪ Final Offers. This is known as a *breadcrumb* and it simply means that wherever you are in the structure of the site, you can jump back to anywhere on the path to it. It's no big deal in this case because there is a Home link available anyway in the main title bar, but in a deep structure, it's a real time-saver. (By the way, earlier versions saw you madly clicking Back, Back, Back, on the browser.)

Another small thing is that, whereas in the version 2 products it was necessary to amend the Links on the Title bar in Front Page 2003, this is now a function that is under Site Settings. So, you can add, amend, and remove links while still in the browser.

But the really big change has probably escaped you newcomers to SharePoint because it's something that obviously has to be there. This is the Recycle Bin at the bottom of the Quick Launch section. In the v2 SharePoint products, if a user deleted a document, it was gone for good — unless that user was the boss, in which case recovery of that single document meant the administrator using a backup of the entire site and using it to create a copy of that entire site on a spare server. This was the only way to recover that single document (if you were lucky and the backup included that document).

So, the WSS 3.0/MOSS 2007 addition of a Recycle Bin is a major plus. This is a Recycle Bin only for data stored in the WSS database and is of two levels. There is one level the user sees and one that I suggest you keep the user unaware of — namely a second-level Recycle Bin (for Administrators only).

There's nothing weird about the concept. Documents deleted by the user move to the Recycle Bin until they are either deleted by the user, or the time limit for such documents runs out (the default is 30 days). From that point on, the user cannot individually recover the deleted documents, and will timidly approach the administrator. In my mind, the administrator should reply that he or she will see what can be done, then go for lunch, and, upon returning, go to the second-level Recycle Bin that only the administrator can see. With one click, the administrator could recover the document.

This second-level Recycle Bin retains documents for a percentage of the Quota Limit, which is set by default to 50 percent (which seems rather high to me). Reducing this and not letting the users know about the second-level Recycle Bin seems a good option to me — experience shows that if users do know about the second-level Recycle Bin, and know, too, that it contains almost everything they've ever deleted, they will be (shall we say) "less responsible."

This is perhaps a good time to use the Recycle Bin as an example of the places that only the administrator can get to.

The user sees his or her own Recycle Bin, as just mentioned, but isn't aware (unless told) that there is a second-level Recycle Bin. This is because the administration of that second-level Recycle Bin is under Site Settings, and as you have learned, for a user, that menu is restricted to things he or she can work with. So, the second-level Recycle Bin link in Site Settings is only visible to a person who is an administrator. All that the administrator can see is a long list of documents that have been deleted; there are no administration settings that can be amended there.

This is fairly normal. If you want to work with an administrator function, go to Site Settings; but if you want to specify the defaults, go to SharePoint v3 Central Administration (also called Central Admin, and discussed in more detail in Chapter 3). In the v3 products, you can either go to the Server (Start ➪ Administration Tools ➪ SharePoint v3 Central Administration) or, in the browser, go to your site but use the port that was randomly generated for the Central Administration site when you installed WSS 3.0 (in other words, you access `http://servername:nnnnn` where *nnnnn* is the port name).

Both methods of getting there give you the same Web page and, in the case of amending the Recycle Bin settings, you must go to Application Management and then to Web Application General Settings, where you'll find what appear from the somewhat confusing different default types (number of days / percentage of quota) for the first and second-level Recycle Bins. There is also the option for both of having the Recycle Bin off. Somehow, having been so relieved that v3 finally has a Recycle Bin, I can't see me turning it off in a hurry.

> *Although the rough split just mentioned (Central Administration for specifying settings; Site Settings for routine administration) applies a lot of the time, there are no guarantees. So, the best advice I can give you is to first look in the place that seems most likely to be correct for what you want to do as an administrator. But, if you don't find it there fairly quickly, try the other location. Failing that, search the WSS FAQ to see if it's listed there. Failing that, post a message to a SharePoint newsgroup. Some of these settings are well hidden and not in an obvious location (just think of removing Quick Launch by selecting Tree View that was mentioned earlier).*

Next are the requirements for clients and the effect on what you can do in a site, depending on what software you have installed on a client and which browser you are using there.

Interaction with Browsers and Clients

The browser story is simply told. Browsers that support ActiveX have potentially full functionality, whereas those that don't have ActiveX (such as Firefox or Apple's Safari) don't. (Note that from this point forward, "Firefox" will be used to refer to all such non-ActiveX browsers.) This lack of functionality isn't usually important for a user, but is important for an administrator. So, administrators (even if they regularly use Firefox) should also have a copy of a recent version (6.01 upward) of Internet Explorer (IE) handy.

Firefox seems to be faster in use than IE, and is certainly faster when initially loading the site because it does not load any ActiveX components (which are always the first step for a new IE session). The Web designer must be slightly careful and check that his or her amendments look equally good in both main browsers, but this is merely cosmetic. The main problem is with things such as a drop-down (on or after the filename in a view) followed by a right-click, which works in IE but not in Firefox. In some cases, drop-downs work, but Firefox shows fewer items, as demonstrated in Figures 1-15 and 1-16.

Figure 1-15: Firefox 2.0.0.1

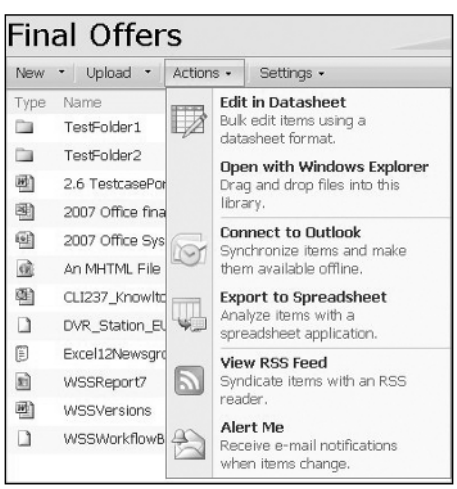

Figure 1-16: IE 7.0

Similarly, multiple uploads of documents can be very useful for the administrator, and this, too, is only available in IE (and also requires a particular Office version — see the discussion later in this chapter).

In the case of a lack of a drop-down, the administrator must provide users (of Firefox and so on) with an alternative to the drop-down for editing an item in a list. In other words, the administrator must provide an Edit icon in the view. In the second case, the user of Firefox is forced (if he or she uses the Web User Interface) to upload one document at a time. Similar to the drop-down on the filename itself (rather than as on the area after the filename) is the way to use "Save As" to save a document to the user's hard disk. In this case, a Firefox user must open the file (read-only) and then, in the application, do "Save As" to save to the hard disk.

The interaction between a client and a SharePoint Web site (apart from the differences caused by a browser) is very much governed by which Office version the client is running.

The basic principle here is that for each version level of SharePoint, there is a corresponding version of Office. So, STS was released as a part of the extended Office XP package, WSS (2.0) was released at the same time as Office 2003, and WSS 3.0 was released at the same time as Office 2007. Because of this, new functionality in the new version of SharePoint is only available if the client is using the corresponding new version of Office.

So, if the client has Office XP, functionality is available that was available with STS, but no more; if Office 2003, functionality is available that was available with WSS 2.0, but no more. (Office 2000 has very limited functionality.)

> *That is the basic principle, but as you'll see when reading the "WSS 3.0 and Office 2007 Working Together" section later in this chapter, in a few cases, functionality that was available with WSS 2.0/Office 2003 is no longer available with WSS 3.0/Office 2007.*

There is a very good Microsoft paper for WSS 2.0 that describes in detail the differences between the use of Office 2000, Office XP, and Office 2003 with WSS 2.0. You can use this paper for details of what functionality you will have with WSS 3.0 if one of these Office versions is installed on your client (and not Office 2007). The paper is called "Good, Better, Best: Windows SharePoint Services Integration with Microsoft Office" and is available here:

```
http://www.microsoft.com/office/sharepoint/prodinfo/officeintegration.mspx.
```

> *There is a newer version of this document now titled "Microsoft SharePoint Products and Technologies Document: Microsoft Office Programs and SharePoint Products and Technologies Integration—Fair, Good, Better, Best" available at:*

```
http://www.microsoft.com/downloads/details.aspx?familyid=e0d05a69-f67b-4d37-961e-2d
b3c4065cb9&displaylang=en&tm
```

One thing to note is that, whereas Office 2003 (and 2007) has support for SharePoint set by default, this is not the default for Office XP. This may well need to be set by amending the user's copy of Office XP (Control Panel ➪ Add or Remove Programs ➪ Select Office XP and specify Change) and going to the Office Shared Features section and selecting SharePoint support there. (In Office 2007, this is in the Office Tools section at the end, and is "Windows SharePoint Services support.")

> *Another thing to watch out for is that not all versions of the same Office release have the same functionality when used with SharePoint sites. This particularly applies to Office 2003 Standard and Business Editions, which do not, for example, support Datasheet view (unlike Office 2003 Pro Edition), although they do support multiple uploads (if IE is used).*

The usual problems for users of back versions of Office lie in two areas. First, they will not (unless they have at least Office XP) be able to create a new document in the browser by using "New Document," and they will not be able to use the "Edit in Microsoft Word" (if a Word document) function and then save the amended document back to the site (this again requires at least Office XP). This is the same problem faced by people who are storing documents from any application other than the main Office products (Word, Excel, PowerPoint) because this "New Document" and "Edit In" problem applies to all non-Office products — even Visio files are only covered with Visio 2003 (or 2007) installed on the client.

The main problem, however, lies in the area of synchronization between SharePoint and Office, which is examined next. There have been considerable improvements in synchronization in SharePoint v3/Office 2007 compared to the limited first attempt at this with SharePoint v2/Office 2003, and all these improvements are lost if the user base is not using Office 2007.

The next section provides an overview of the main ways in which this improved functionality can be beneficial.

WSS 3.0 and Office 2007 Working Together

There are two main ways in which WSS 3.0 and Office 2007 work together.

The first main set of functions is the ability of the user (when using Excel, PowerPoint, and Word) to specify a Document Workspace as the place where a document (that the user is saving) is stored. The other variant of this is with Outlook, where there are two functions where a SharePoint site is used.

First, when a meeting is created, it is possible to specify that a Meeting Workspace is to be created for that meeting (and the people invited to that meeting are given rights to access the workspace). The second is that, when a message is sent with an attachment, it is possible to specify that the attachment is stored in a SharePoint site (so that people always get the latest version of the document when opening email and are told automatically about later versions).

These basic functions were changed little between Office 2003 (where they were introduced along with WSS 2.0) and Office 2007. What improvements have been made for Office 2007 are beyond the scope of this chapter, and for now, the key thing to remember is that both the Document Workspace and the Meeting Workspace are only sites created using a particular site template. The same site templates are also available when creating a normal site or sub-site. The difference here is that those sites would be part of a site structure, whereas Document and Meeting Workspaces are single independent sites. There are often more of them, and they are usually more difficult to administer (in the sense that they are created more or less randomly and often no one bothers to delete them when their use is over) than sites within a site structure. However, there are ways to automatically delete them after a time.

The second main way in which WSS 3.0 and Office 2007 work together is to have lists that are synchronized (or copied) between the two. Two-way synchronization is an area in which functionality has been greatly improved in connection with Office 2007, whereas Office 2003 with WSS 2.0 offered only one-way transfers.

Access 2007

One product that didn't have synchronization before is Access 2007.

There is no built-in function in the SharePoint v3 products for providing quality reports of the contents of SharePoint lists. All you can do is to create a suitable view and hope it will fit into standard letter (or A4) format when it's printed. Thus, one of the ideas of the 2007 products is to provide a couple of alternative methods of creating reports from SharePoint v3 lists. One is to use SQL Server 2005 Reporting Services to extract data direct from the SharePoint databases. The "lighter" version is to synchronize a list between Access 2007 and WSS 3.0, and then use the Access reporting functions to create a report on that SharePoint list.

This latter method is intended to replace the previous usage of transferring data to Excel and manually producing reports there. In Access 2007, it works as follows.

Note that Excel 2007 no longer provides this functionality.

Access 2007 is opened and in the External Data menu bar, "SharePoint List." A WSS 3.0 site is specified in the format `http://servername/sitename` and "link to the data source by creating a linked table." This will provide a list of lists available in that WSS 3.0 site, and you can select one or many by using checkboxes. In this case, let's take a document library with some documents in it. Let's do this by opening the Final Offers document library, and then, using Upload from the menu bar, selecting "Upload Multiple Documents."

Because this is the first mention in this chapter of using the Multiple Documents function, note that there is a limit for the maximum size of an upload of a batch of documents (which also applies as the limit for a single document). This default limit is now 250MB compared to the 20MB it was in WSS 2.0 (where there was also a separate limit for a single document). This limit can be amended if necessary in SharePoint v3 Central Administration ➪ Application Management ➪ Web Application General Settings (although, compared to WSS 2.0 where 20MB was far too little, 250MB should be enough in most environments).

Select the document library (in Access 2007) and click Synchronize (top-right section of the menu bar). Then, double-click the document library and you will see (if this is a newly created standard document library) five columns: ID, Checked Out, Name, File Size, and Title. Checked Out typically contains no information and Title typically only sometimes contains information.

Select Create in the menu bar and click Report. Now, select the ID text and delete it; repeat for Checked Out. Now, select Title and drag it to between Name and File Size, and also widen it. Play with the widths of all the columns until they use the full width of the printed page. Finally, sort the list by File Size by selecting "Group & Sort" in the menu bar. Then select "Add a Sort" (it appears below the list), select File Size, and then amend the settings to "from largest to small." Click Done.

Now, you can use Save and it will know that you want to save a report. Give it a suitable name such as "DocLib_SortByFileSize" and click OK. Now, both the document library and the report are saved in the same section in the left-hand column in Access 2007.

This is, of course, a trivial example. However, the idea is for you to see what Access 2007 offers for report creation that WSS 3.0 itself doesn't. Note that there could be several ready-made reports available in the same section as a linked list, and that these automatically receive the latest data from the WSS 3.0 site via synchronization. Note too that "Report Design" (instead of "Report") gives you a more powerful way of creating your report, including creating reports made by linking two or more lists.

Excel 2007

It's logical to move on to Excel 2007 at this point. As stated in the Access 2007 discussion, using Access is Microsoft's solution to what was earlier done with Excel 2003. To ensure that users don't just ignore this and work with Excel 2003 anyway, Microsoft has taken the drastic step of actually reducing functionality between Excel 2007 and WSS 3.0 compared with what was available with Excel 2003 with WSS 2.0.

In Excel 2003, it was possible to connect to (read) and update (write) lists that were located on SharePoint sites. In Excel 2007, this is no longer possible.

Instead, Microsoft has tried to avoid problems for those people who already used this in WSS 2.0 by allowing existing files from Excel 2003 to load. Microsoft has also continued to support update functionality (in connection with WSS 2.0) when files are opened in Excel 2007. However, once these files are saved to the new Excel 12 file formats, even that is no longer possible, and lists that are linked to SharePoint will be converted to read-only tables.

It is still possible to "Publish" an Excel table to WSS 3.0 (by using File ➪ Document Management ➪ Server) just as it was in Excel 2003 with WSS 2.0, but this is only a one-time write to SharePoint. It is equivalent to the "Create List from Excel" function in WSS 2.0 with which you could move the contents of an Excel file to become a WSS 2.0 list (or what is now the "Create/Import Spreadsheet" function in WSS 3.0).

Outlook 2007

When talking about Office 2007 and WSS 3.0, interaction with Outlook 2007 is perhaps the area in which the greatest amount of work has been done, and unlike the situation with Excel 2007, this has not been done at the cost of losing existing functionality. You can still send messages in Outlook 2007 where you specify that your attachments should be stored in a WSS site, and this still means that readers of your email message will see the latest version of that document when they access the message and not the version that was current when the message was sent. (Earlier readers will be informed that an attachment they have already accessed has been amended.) You can also still create a Meeting Workspace when creating a message, informing a group of people about an upcoming meeting, and these people will still be automatically granted access to the small site created for the meeting (and be informed by email of how to access it). This site is a good way to have a Web location for a single small project (one typical reason for inviting people to a meeting).

But, back to the newer features, interaction occurs with calendars and contacts. In Outlook 2007, you can administer SharePoint documents, and SharePoint tasks are tracked in the same way as Outlook tasks.

When considering the interaction between SharePoint calendars and contacts and Outlook 2007, one key thing to be aware of is that these are always listed *in addition to* the standard Outlook 2007 calendars and contacts. In other words, in Outlook 2007, you will see a list of both your Outlook calendar *and* one or more SharePoint calendars. The interaction available is that you can administer your SharePoint calendars within Outlook 2007, and you can show your SharePoint calendar(s) alongside or overlaying your Outlook calendar (just as you can do with your Outlook calendar and the Outlook calendar of a colleague). You do not, however, combine the two calendars in a permanent way to become a single calendar.

SharePoint contacts, too, are listed completely separately from your Outlook contacts, and here there is no equivalent to the joint calendar view. Instead, "interaction" between the two consists of the ability to copy and paste contacts between them. Tasks are better. They are tracked in the same way as Outlook

tasks and appear on the To-Do bar, for example. Just as in the previous two cases, Tasks deriving from SharePoint can also be administered in Outlook 2007. Here, too, a task can be created in Outlook 2007, and then saved to a SharePoint site, but still with the option of adding the Task to the To-Do bar.

The ability to administer SharePoint documents in Outlook 2007 started out for me as something of a "so what?" bullet point. I didn't see the point of using another system instead of a SharePoint site itself to search for and open/save documents that are located in SharePoint (which are what Microsoft highlights as being better in Outlook 2007 than before) or even — to use an example that Microsoft gives — to have the files from a SharePoint library listed in Outlook 2007 so that you can refer to them there rather than by opening the SharePoint site in a browser. I assumed this was only useful to the user whose life revolves around Outlook, and whose SharePoint needs are limited and occasional. But then I starting investigating this function and found that there were a couple of very good reasons for having the contents of a document library visible in the Outlook 2007 User Interface.

> *To set this up, by the way, open the document library in WSS 3.0 (or MOSS 2007) and click the Actions menu item. If you haven't done this before, you will see a "Connect to Client" option and when you select this, you will be prompted to set up a connection to Outlook (the alternative is that "Connect to Outlook" is listed). Don't be surprised if this process takes quite a while (and perhaps demands re-authorization from you), but in the end, you will have an entry in Outlook 2007 in the "SharePoint Lists" section toward the end and below all the Mailbox entries.*

When you have the SharePoint document library files accessible in Outlook 2007, one thing you can do (that you can't do in SharePoint itself) is to take the files offline to work with them. Then, when you get back, you can update those documents in the SharePoint document library (again via Outlook 2007). For some people, this would be a very valid reason for having a synchronized copy of a SharePoint document library in Outlook 2007.

However, the feature I liked the most was the ability to have a three-pane screen with the Mailbox at the left, the document library in the center, and a pane in the right of the screen that shows the *contents* of the documents (Figure 1-17). This is provided that Outlook 2007 provides Viewer support for them, which it does for Excel 2007, PowerPoint 2007, Word 2007, Visio 2007, and `jpg` (and other image formats) and text files. (As far as I could see, files saved under older versions of the Office applications also were viewable.) Being able to quickly browse through PowerPoint slides in this manner was, for me, a really nice-to-have. (I needed access to my work Outlook; hence, the slightly different site name in Figure 1-17.)

Be aware that both PowerPoint 2007 and Word 2007 can store their documents in a so-called Document Workspace (or one small site meant for discussions and so on about a single document) that works for documents in the same way that a Meeting Workspace works for Meetings (and small projects).

Ways to See What's New

Now that I have gone through most of the basics of what can be in a WSS site, it's time to move on to what a normal user's needs are.

Most users don't contribute to Web sites. They just want to quickly and easily access information that is important for them. There are several ways for an administrator to ensure that this goal of the user is attainable.

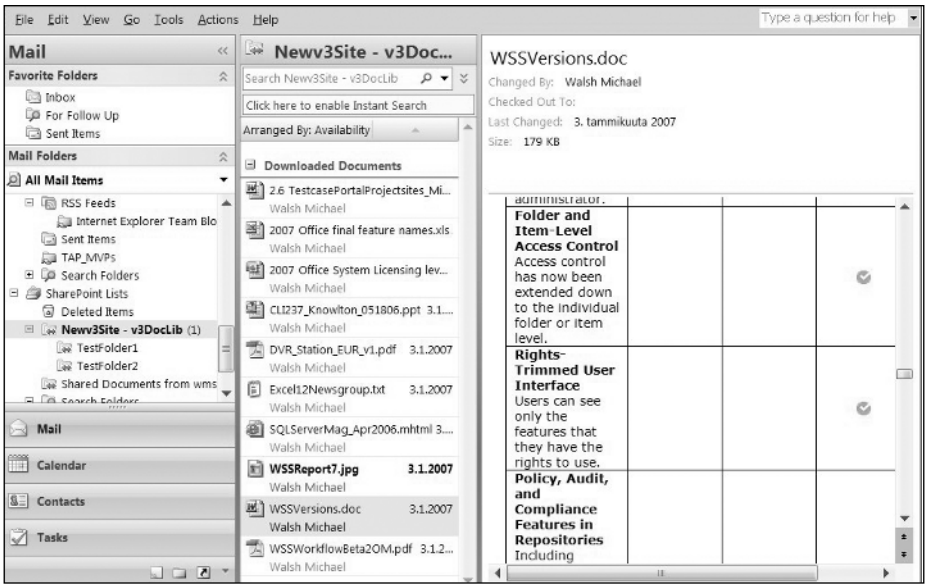

Figure 1-17: Three-pane screen

As I've mentioned already, the first thing is to have pages with Web Parts showing the latest few documents or items. If this is done according to user needs (is he or she is interested in the last few new documents or the last few amended documents — a single Web Part can only show one or the other), then all the user needs to do is to have the URL to that Web page (or those Web pages) in IE Favorites (or similar for another browser). The user can then access the page regularly to see what's happening — for example, to see if a Project Meeting report has been added to the site yet.

This is nice and simple, but a bit haphazard. What happens if the six latest items are listed, and there has been a flurry of action since the last visit? In fact, what happens if seven items have been added since then? It will be there in the list, but not visible in the Web Part in the page. So, you need some other options, which may include the following:

- Alerts
- RSS feeds
- Search functions

Alerts

Alerts are one such option. In the SharePoint 2007 products, it is possible to specify an alert on all levels (site, sub-site, list, folder, item, and even view) and, if anything changes, the user receives an email that can be immediate or accumulated *per alert* on a daily basis.

The ability to be able to create alerts is dependent on the administrator for the SharePoint installation having defined that alerts are supported. However, typically this just means specifying an Exchange Server to handle the emails (or, indeed, any other SMTP mail server).

Even so, there are a couple of snags with alerts. For the administrator, any additional feature adds to the amount of questions he or she will get from users and, with emails being involved, the number of questions increases a lot. For example, there will be questions about the form of the email, with the user wanting more or different information on it. There will be questions about the frequency of alerts, and why isn't a user is getting an alert when that user has amended an item, or why is the user getting an alert to tell him or her that he or she has created an alert (all of which happen). All of these are at the level of "small and irritating," rather than particularly difficult for the administrator to deal with. However, they still provide a reason for an administrator with too much on his or her plate to not implement alerts.

For the user, there are, perhaps surprisingly, downsides as well. If the user sets alerts at too high of a level (the site, say), then the user is alerted to everything that moves on the site. So, there is typically a very long daily message, most of which doesn't interest the user. However, if the user sets alerts at too low of a level (say, the folder level), then there will be a large number of email messages (one per folder) daily. In other words, just as when signing up for free newsletters, care should be taken by users when signing up for alerts.

RSS Feeds

For the user who doesn't like adding to the number of emails received, the SharePoint 2007 products offer something that wasn't in the v2 products — *RSS feeds*. Because you can safely assume that a user is already using email, my feeling is that only users who are already using RSS feeds for other things will bother with adding RSS feeds from WSS 3.0 lists, and that they would be well-advised to consider using both alerts and RSS feeds, depending, in part, on the urgency of the information.

> *I'm assuming here that it is never a good idea to have an alert email sent at once, and so you can divide your warnings of changes into daily batched alerts, and use (immediate) RSS feeds for the really key lists only.*

So, alerts and RSS feeds have their uses, but what about users who don't want to either continually follow developments in a site or to receive alert emails or read RSS feeds? They can rely on the Search functions that WSS 3.0 (and MOSS 2007) includes.

Search Functions

Search in the 2007 SharePoint products is based on the same search technology that Microsoft uses for its MSN site — in other words, it's a Google-level search technology (or at least is close to that). Unlike the situation with the v2 products, the search technology used is the same for both WSS 3.0 and MOSS 2007.

The differences only come in the scope of the search, with WSS 3.0 (like WSS 2.0) being restricted to search data stored in its own databases and, in fact, also being restricted to searching in a single site in an installation rather than over multiple sites in the same SharePoint installation as MOSS 2007 can do. (MOSS 2007 can index and, thus, search file shares, public folders, Web sites, SharePoint sites, databases, structured and unstructured data sources, and so on.)

What can be searched when using the product as delivered is not by any means everything most people will want to search. This is because you can only search what has been indexed, and what can be indexed is dependent on what file formats the indexing system can recognize. As with most Microsoft products, this means the main Office products and the most general neutral file formats such as `.txt`.

This takes us to the obvious missing format that most people would want to include in their searches — the `.pdf` file format used by Adobe Acrobat files. Just as with all other missing formats, the way to make

sure that the `.pdf` file format is indexed is to install an *iFilter* for `.pdf` files on the server doing the indexing. Microsoft's strategy is to provide iFilters only for their own main products (and for those very widespread neutral formats), and to rely on the software makers to write software that uses their own file formats to provide iFilters for those file formats. Mostly, that type of iFilter is provided free by those vendors because they want the files produced by their software to be searchable in such applications as the SharePoint software, whereas filters for less common (but still neutral) formats are often provided at a cost by small, third-party vendors.

Luckily, an iFilter for a 32-bit SharePoint v2 system will still usually work for a 32-bit SharePoint v3 system, so the iFilter that Adobe has long provided for Acrobat files will still work with WSS 3.0 and MOSS 2007. (Less lucky is the fact that this won't work for a 64-bit system, and Adobe has, as of this writing, not provided a 64-bit version.)

Another thing that is strangely lacking in v3 of SharePoint as it was in v2 is an icon for `.pdf` *files. You may have noticed that in a document library view, the line normally starts with an icon, and that this is a Word icon, or an Excel icon, and so on. All file types that are not recognized are given the same Windows icon. This is, however, easily fixed. There's a* `pdf` *icon available on the Adobe site, and this can be slotted in to a file on* `C:` *to provide even Acrobat files listed in WSS 3.0 with their own icons. (Add the icon to* `C:\Program Files\Common Files\Microsoft Shared\Web server extensions\12\TEMPLATE\IMAGES`*, and then add an entry for* `pdf` *to* `C:\Program Files\Common Files\Microsoft Shared\Web server extensions\12\TEMPLATE\XML\DOCICON.XML` *and after saving do* `IISRESTART`*.)*

Additional Areas of Interest

Finally, here are a few words on a few additional areas of interest.

Customizing a Site

There are roughly three different ways you can customize a site. You can customize it using the user interface, you can customize it with SPD 2007, or you can customize it with Visual Studio 2005.

- *User Interface changes* — Done in a browser while accessing a WSS site, these affect the look of the site in standard ways. For example, you can apply a new Site theme (Site Settings ➪ Site theme) so that at least the colors used in your site are different from the standard that most people will use when using the product as delivered.
- *SPD 2007* — This is typically used to make changes to the code that Microsoft supplies with the SharePoint product. So, to take a trivial example, a heading that Microsoft provides could be amended or removed; other headings could be made larger; or the color changed, or both. A more complicated (and far-reaching) example would be to amend the master page of the installation.

Changes to the 2003 SharePoint products were made using FrontPage 2003. SPD 2007 is really the renamed FrontPage 2007 — renamed because of the much greater amount of functionality it contains and also because of the new concentration on this being a tool only for SharePoint sites. You should not use FrontPage 2003 to amend WSS 3.0 or MOSS 2007 sites and, although you can use SPD 2007 to amend WSS 2.0 and SPS 2003 sites (backward compatibility), I would advise against it, because there will be many functions in SPD 2007 that will not work with the 2003 SharePoint products. The use of SPD 2007 with those products might prove to be confusing.

- *Visual Studio 2005* — This is typically used to add functionality to a SharePoint implementation. For example, this could be the creation of a new Web Part offering functionality that the sold product doesn't offer in its Web Parts (or perhaps, that is included in MOSS 2007, but you want to use the same functionality with a WSS 3.0 installation, so you need to write your own). It could be new code that (at the extreme) provides a new customer interface to a SharePoint installation. Typically, such new code would use the many functions that are described in detail in the WSS 3.0 and MOSS 2007 Software Developer Kits (SDKs), which can be downloaded from the Microsoft site.

Virus Checking

One common question is to ask is whether normal server-based virus checkers are enough for SharePoint systems. The answer is that they are not — you will need a special SharePoint virus checker installed on the front-end servers (in the case of a farm), or on the single server (when that's all you have).

After the 2003 SharePoint products were released, it took a while for the main virus checker vendors to provide SharePoint versions of their products, and it looks as we are in the same situation today with the 2007 SharePoint products. The only exception that I know of, as of this writing, is the Forefront product from Microsoft itself ("Forefront Security for SharePoint 2007"). This is an update for WSS 3.0 and MOSS 2007 of a product for the 2003 SharePoint products that was previously sold by Sybaris and, along with other Sybaris products, was acquired by Microsoft when it bought the company. Like that earlier product, it continues to use most of the many third-party scanners that were the strength of the Sybaris software.

Backup/Restore

Here, too, there are special needs for SharePoint products, although typically any backup scenario will also include a standard SQL Server backup of the configuration and (especially) content databases.

This area is covered in Chapter 3 of this book, and is also covered in a chapter of the *Microsoft Office SharePoint Server 2007 Administrator's Companion* by Bill English (Redmond, WA: Microsoft Press, 2007).

Workflow

Workflow wasn't available in the v2 products, and was a major customer request for the v3 products. Chapters 8 and 9 discuss workflow, so for now, this is just a note for you that workflow is based on the Windows Workflow Foundation (WWF) that is installed as part of .NET Framework 3.0.

WSS 3.0 includes only one built-in workflow; otherwise, with WSS 3.0, you create sequential workflows by using SPD 2007. In addition, you can create more complicated workflows (with loops) in Visual Studio 2005.

Wikis and Blogs

These are new functions in the 2007 SharePoint products and, because of this, still only offer basic functionality.

To create a site for a wiki (or a blog), you follow normal procedures to create a site, but when given the choice of which site template to use for the site, you select the wiki (or blog) template. This site is just

like any other site created in this way in that it is a sub-site to the site in which it is created, and may (or may not) inherit the permissions of the parent site.

The standard wiki site, when created, includes a link to a Web page, "How to Use This Wiki Site," which gives the basic information that you'll need. Going into details of this and blog sites (which are created in a similar way) goes beyond the scope of this chapter.

For More Information

Apart from the Microsoft SharePoint Web pages starting at `www.microsoft.com/sharepoint`, there are various private Web pages full of SharePoint information. The two I will mention here are my own WSS FAQ site (`www.wssfaq.com`) and Ian Morrish's WSS DEMO site (`www.wssdemo.com`).

As far as other books are concerned, the most complete list of SharePoint 2007 books can be found here:

```
http://wss.asaris.de/sites/walsh/Lists/WSSv3%20FAQ/V%20Books.aspx
```

There are books for all needs, but most administrators will probably find that one book they will definitely need to have will be the *Microsoft Office SharePoint Server 2007 Administrator's Companion* from Microsoft Press, although, if past experience is any judge, several of the chapters from this book may well (at a later date) appear in `.pdf` form on the Microsoft Web site.

Questions can be asked in the Microsoft public newsgroups at `msnews.microsoft.com`. There are four newsgroups for different aspects of using the products, and they are for all SharePoint products and versions, so make sure you say which exact product you are using when asking a question there.

- `microsoft.public.sharepoint.setup_and_administration` is for Installation and Administration questions.
- `microsoft.public.sharepoint.design_and_customization` is for questions on amending the code supplied with the product, typically (if WSS 3.0 or MOSS 2007) with SPD 2007.
- `microsoft.public.sharepoint.development_and_programming` is for questions on creating new code such as writing Web Parts or writing code that uses Web services, and this typically uses Visual Studio 2005.
- `microsoft.public.sharepoint.general` is for questions that don't fit in any of the previously mentioned newsgroups.

Microsoft forums are also available. These are listed here, along with their RSS feeds, which is the way I prefer to read them.

- *SharePoint General Q&A and Discussion* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1200&Mode=0&SiteID=1`
- *SharePoint Setup, Upgrade, Administration, and Operation* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1201&Mode=0&SiteID=1`
- *SharePoint Design and Customization* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1202&Mode=0&SiteID=1`

- *SharePoint Development and Programming* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1203&Mode=0&SiteID=1`
- *SharePoint Business Data Catalog* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1204&Mode=0&SiteID=1`
- *SharePoint Business Intelligence* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1205&Mode=0&SiteID=1`
- *SharePoint InfoPath Forms Services* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1206&Mode=0&SiteID=1`
- *SharePoint Workflow* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1207&Mode=0&SiteID=1`
- *SharePoint Excel Services* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1208&Mode=0&SiteID=1`
- *SharePoint Search* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1209&Mode=0&SiteID=1`
- *SharePoint Knowledge Network* — `http://forums.microsoft.com/MSDN/rss.aspx?ForumID=1210&Mode=0&SiteID=1`

As of this writing, these RSS feeds only gave links to the first items in each thread, and there were no feeds for replies. Microsoft has promised to fix this by the time this book is published, but check this when you first use these RSS feeds just to be sure.

If you don't like the style of these Microsoft newsgroups or forums, then Dustin Miller's "SharePoint University" has its own SharePoint v3 forums and can be found here:

`http://www.sharepointu.com/forums/default.asp.`

Finally, Microsoft's Product Services Support (PSS) team has as always actively been creating Knowledge Base (KB) articles warning of problems and giving fixes or workarounds. The Knowledge Base can be searched at `http://support.microsoft.com/search?adv=1`, but again, I find the best way to follow developments is to use the relevant RSS feeds, which are the following:

- *Windows SharePoint Services 3.0 KB Articles:*

  `http://support.microsoft.com/common/rss.aspx?rssid=12200&ln=en-us&msid=232c70a3bb2693499aef676972853630`

- *Microsoft Office SharePoint Server 2007 KB Articles:*

  `http://support.microsoft.com/common/rss.aspx?rssid=11373&ln=en-us&msid=232c70a3bb2693499aef676972853630`

- *SPD 2007 KB Articles:*

  `http://support.microsoft.com/common/rss.aspx?rssid=11677&ln=en-us&msid=232c70a3bb2693499aef676972853630`

Even if you are only using WSS 3.0, you will still need the MOSS 2007 feed because the categorization is often random. Similarly, many of the supposedly SPD 2007 feeds will be general Office 2007 feeds and will not apply to SPD 2007 specifically.

If you want to avoid these problems and also see KB articles that are relevant to WSS 3.0 and MOSS 2007, but that have been allocated to, say, Word 2007 or other Office products, check the KB Articles section of the WSS FAQ site where I try to keep track of all relevant KB articles (for WSS 2.0 and for WSS 3.0 and MOSS 2007). The direct address is as follows:

`http://wss.asaris.de/sites/walsh/Lists/KB%20Articles/V3%20Sorted%20By%20Date.aspx`

As with all SharePoint lists, you can set an alert on it. (Note, though, that this is still at the moment a v2 site, and so alerts can only be set at the list level, so you get alerts for WSS 2.0 KB articles as well as for the v3 SharePoint products.)

Summary

This chapter provided brief overview of the main SharePoint concepts by helping you to install a copy of WSS 3.0 and taking you briefly through the various items that you will see on the Web page when you create a site based on the standard "Team Site" template.

This chapter has provided a feel for how the SharePoint products have developed over the years, and examined how the functionality provided has evolved with particular notice being given to the additional features that are included in WSS 3.0 compared to WSS 2.0. As part of this, the chapter also discussed the interaction with Office 2007 products.

The chapter concluded with tips on where to get further information, and where to ask further questions on the SharePoint v3 products.