CHAPTER 7

DISASTER RECOVERY PLANNING

SharePoint is a great aggregator of information. From semistructured content such as documents and images to unstructured content such as blog entries and discussion threads, information within SharePoint is delivered in one place—and, by the default, is stored in one place. This "single storage" model (actually a collection of databases) makes business-side functionality—for example, collaboration, communication, and data consistency—easier to implement. The challenge, however, is that this puts pressure on IT staff to make SharePoint, now a business-critical application, highly available. Users will want consistent access to content, and they'll want comfort around plans to restore some portions of a site, a complete site, or a collection of sites. The focus of this chapter is leveraging native SharePoint backup/restore capabilities to recover or recreate entire portals or sites.

The feature set contained within SharePoint Backup and Restore constitutes only one component of an overall disaster recovery plan. This chapter provides an overview of the SharePoint Backup and Restore utility. In addition, it details what components of your portal or sites are and are not covered with SharePoint's native backup and restore tools.

When creating a disaster recovery (DR) plan, you need to determine what you are trying to recover from. In other words, think of your disaster recovery plan as "taking out insurance" for your SharePoint environment. There are various levels of protection you might wish to set in place. You may be using a DR plan to create a replica of your portal to recover specific content, or you may wish to develop a plan to create a new environment from scratch (in the event of an actual disaster) that will quickly and effectively replicate the current environment.

For example, on one end of the spectrum, you could make things easy on your operational team and back up your data once every six months or so, but then you run the risk of losing a lot of data if something were to happen (a hard drive crashes, you lose power, and so on). On the other end of the spectrum, you can do a full backup of everything daily to ensure that you always have the latest of everything—but does that make sense for your environment?

To properly capture these types of decisions, we recommend creating a DR operations document.

Creating a Disaster Recovery Operations Document

The following is a framework for a SharePoint disaster recovery document. It is important to note that a DR plan is only effective if it is both complete and accurate.

An effective SharePoint DR plan should contain full documentation on how to recreate an entire SharePoint environment from scratch. This requires a process (and discipline) that is accurate and well-maintained. Every time a SharePoint element (for example, a Web part, xml file, and so on) is altered or added, the "disaster recovery inventory document" must be updated.

Here's an example of information that should be captured:

I. Overview
    a. Explanation of when to use this plan
    b. History of any updates
    c. Permissions required to execute the plan
II. SharePoint Backup/Restore
    a. Step-by-step execution plan for your environment
III. Adding Web Parts
    a. Location of all Web part CAB or install files
    b. Instructions for installation
    c. Location of latest Web.config file
IV. Adding Additional Components (Features, Event Handlers, Workflows, and so on)
    a. Location of all files
    b. Instructions for file movement and/or installation

V. Testing
    a. Instructions on how to test a new portal environment
        i. Smoke test (a quick examination of the environment to inspect stability)
        ii. Validation of Web part execution
        iii. Validation of security model
VI. Miscellaneous
    a. Comments collected from previous restorations

After you've got a document underway, you'll want to start filling in your company-specific recovery steps, including your SharePoint backup and restore steps. To determine your specific steps, you need to decide which SharePoint backup/restore options best suit your needs. Let's take a look at the various SharePoint options.

Backup and Restore Options

There are several backup and restore options in Office SharePoint Server 2007, including Web-based Central Administration backup and restore, command-line backup, and the two-stage Recycle Bin.

There are two out-of-the-box options for backing up a full farm: the stsadm command-line tool (`stsadm -o backup`) and the backup UI in the Central Administration site.

You can also use SQL Server's backup and restore utilities, provided you have a full version of SQL Server. SQL Express does not include a GUI for backup, but you can write a script to automate the backup.

Each option provides a different level of recoverability; we'll discuss the options here, including when to use each.

Backup/Restore Tool in Central Administration

If you're familiar with the Backup/Restore utility in SharePoint Portal Server 2003, one of the first things that you will notice is that the Data Backup and Restore utility is no longer listed as a separate executable.

With MOSS 2007, the Backup and Restore tools are now contained within SharePoint Central Administration. In the Operations tab (see Figure 7.1), there is a section dedicated to Backup and Restore.

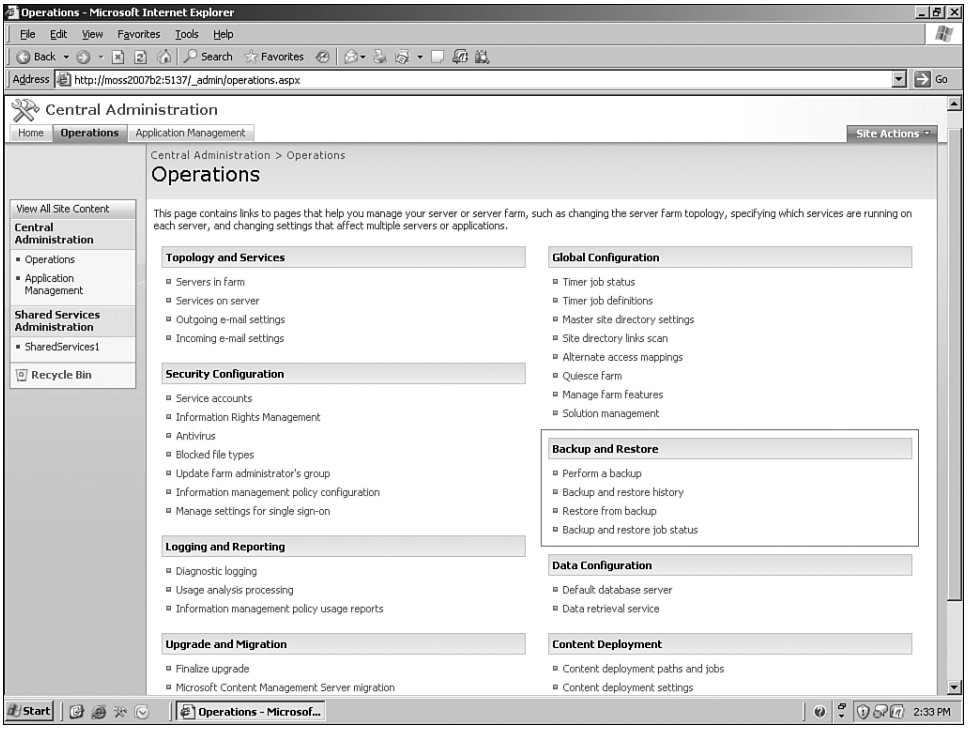

FIGURE 7.1 The Backup and Restore group, located on the Operations page within Central Administration, enables you to perform full and differential backups of your SharePoint farm.

In addition to the relocation of the Restore tools within SharePoint Central Administration, several new features have been added:

- **The ability to select specific farm components for backup.** This includes the selection of an entire farm or specific components such as Windows SharePoint Services Web Application, WSS_Administration, Portal Service, Application Registry Service, Core Services, or User Profile Service.
- **A better interface for managing backups and restores.** The interface is well organized, with clear instructions on expected parameters and intended outcome.
- **The ability to do full or differential backups.** A full backup backs up the selected content with full history. A differential backup backs up all changes to the selected content since the last full backup.

- **More statistics on the backup process.** More information is provided about overall disk space usage, status, and errors.

Using the Backup Utility

One of the great features of the SharePoint Backup tool is the ability to better control what you are backing up. Figure 7.2 shows the interface for selecting which SharePoint components you wish to back up. Each component is associated with a SharePoint database (and ultimately specific SharePoint content) or data collection. It is possible to select an entire farm or individual components for backup.

NOTE To perform a backup, you need to be an administrator on the farm. To run restore, you need to be a Farm Admin and a box admin on the front-end machines.

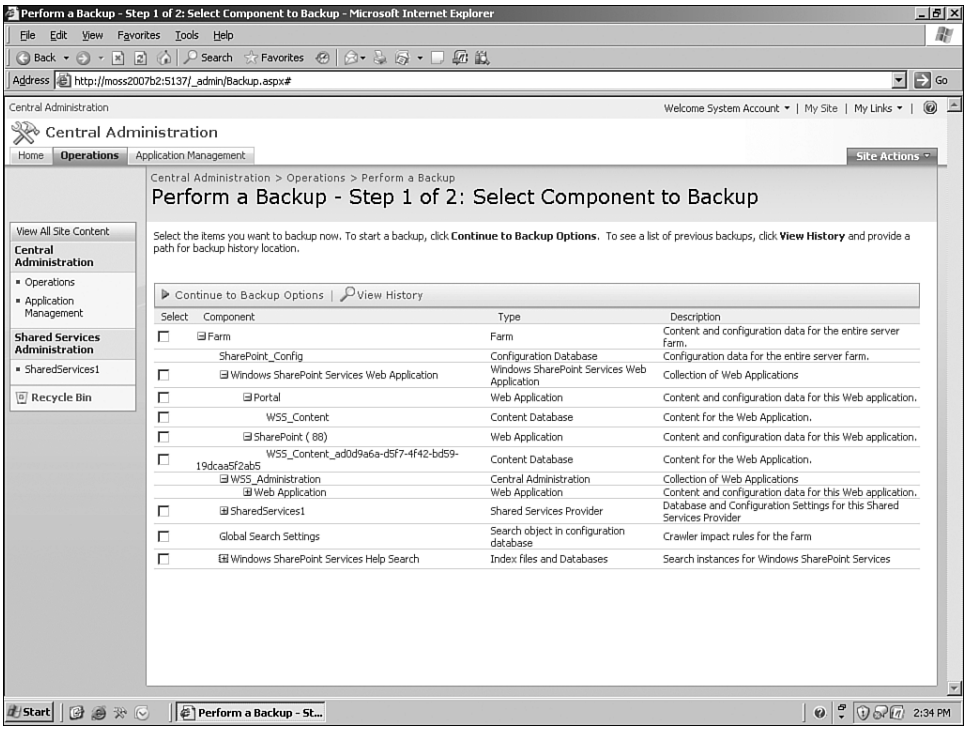

FIGURE 7.2 The backup utility enables you to be selective about which farm components to back up.

Another interesting feature of SharePoint backups is the collection of backup history. SharePoint actually differentiates between full and incremental backups. This is done by examining the backup files on the file system (discussed later in this chapter) and identifying new content.

A full backup backs up the selected content with all the history. Specifically, a full backup backs up the entire database, including all file groups and data files, providing a high degree of data integrity. The downside is that full backups can take a long time for large data stores. We recommend keeping your content databases to a reasonable size (under 100GB) so that backups take a reasonable time.

A differential backup backs up all changes to the selected content since the last backup (either full or differential). This option allows IT administrators to better manage disk space associated with SharePoint backup files. In addition, the backups are faster. The key issue with differential backups is that a restore requires the administrator to restore the last full backup in addition to the differential backups that have taken place.

Given the choice, which should you use? The idea is to use a combination of the two as follows: Start with a full backup of your data. Then perform a daily differential backup of all databases during offline hours. Next, perform a full backup of all databases on a weekly basis. Finally, perform a full restore (to an offline data source such as a mirror server or disk) of your backup set roughly once per month. This lets you validate that your backup procedures are working correctly.

Figure 7.3 shows the Start Backup page. You must specify a location for the SharePoint backup files. The Backup utility accepts only UNC file paths, and permissions on the folder must be sufficient to allow SharePoint Backup (running under the credentials of the logged-in user) to write files to that folder.

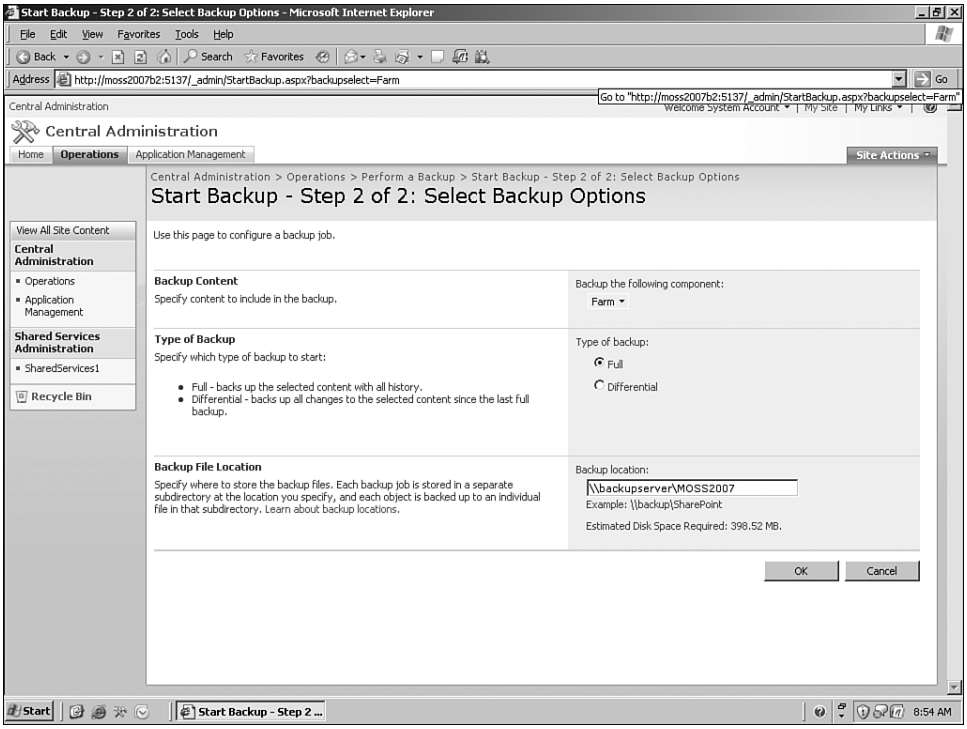

FIGURE 7.3 To start a backup, enter a UNC path to a location where the backup utility should write the files.

Once completed, the Backup tool provides diagnostic details on the backup files created and any errors that may have occurred. As expected, the elapsed time associated with the backup process is proportional to the amount of data being backed up. A standard portal should probably take a few minutes to create all associated files. Figure 7.4 shows a completed backup process. Diagnostic data includes status, elapsed time, file directory path, and associated error messages.

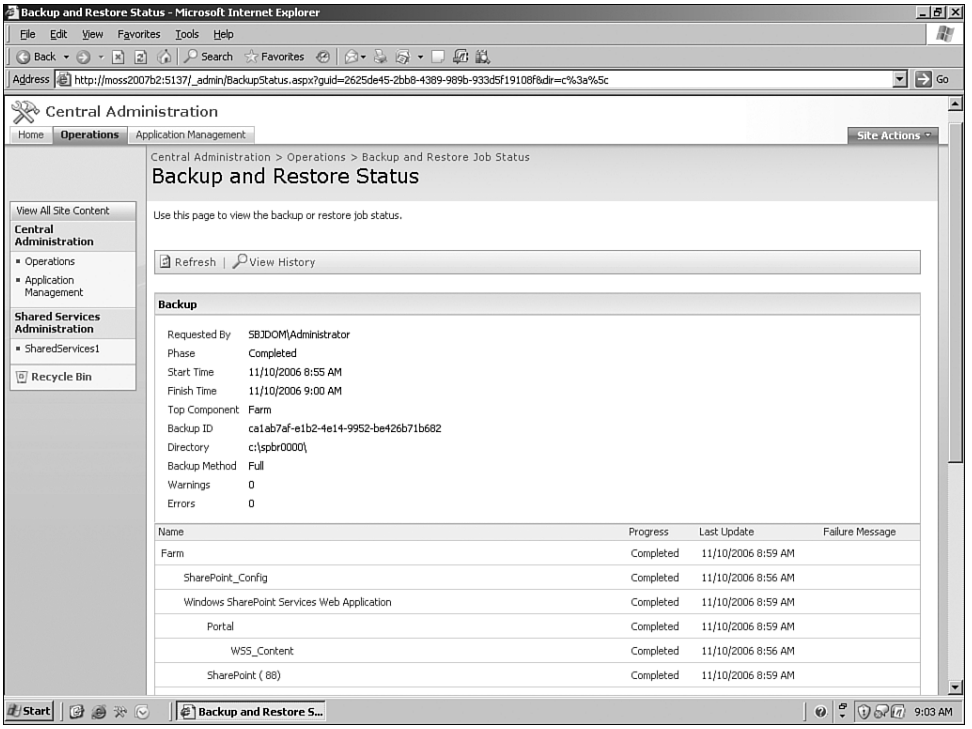

FIGURE 7.4 The status of a running backup or restore (or the result of the backup or restore) is reported in real time.

NOTE It's recommended that you use a remote file share to store your SharePoint backups. Do the following:

1. Make sure the SQL "Setup server account" is using a domain account.
2. On the remote file server, create a folder with a corresponding share.
3. On the file share, grant the following accounts all permission rights (except for "full control"):
   WSS central admin application pool account
   Login account (command line)
   SQL server service account

   Timer Service account. If sptimerv3 is running as a "network service account," add the WSS front-end machine, such as Domain\WSSserver$ (UI).

Examining the Backup Files

When the SharePoint backup completes, the corresponding backup files are placed on the file system in the designated path. If you're familiar with SharePoint Portal Data Backup and Restore, you'll notice that the collection of files is very different. Figure 7.5 shows an example backup file collection. There are two main pieces. One is the spbrtoc.xml (SharePoint Backup Restore Table of Contents) file. The other is the folder that contains all the backup data for that particular backup.

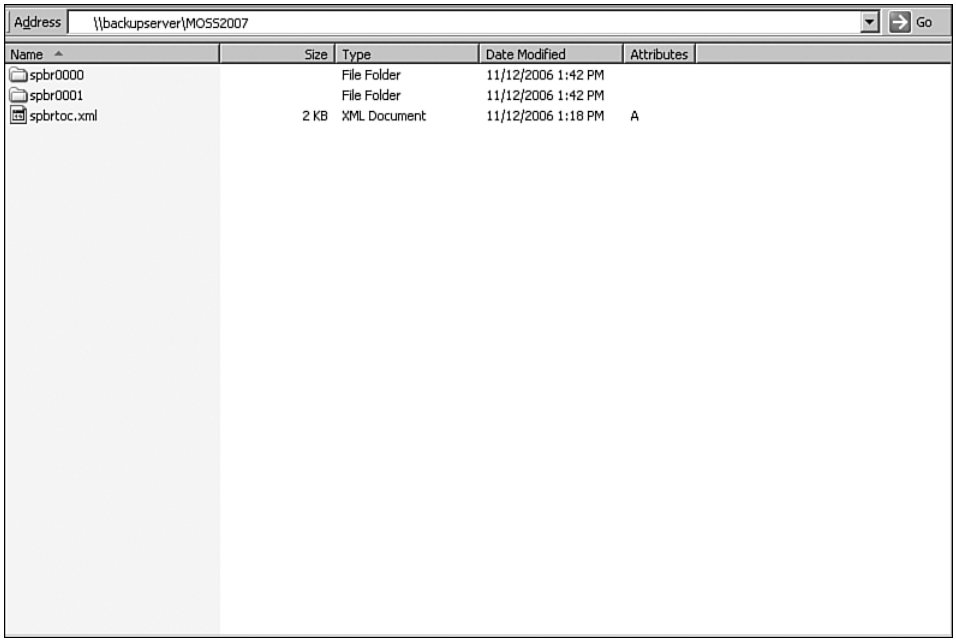

FIGURE 7.5 The SharePoint backups are organized in the spbrtoc.xml file, while each backup instance gets its own sequentially numbered folder.

Let's take a closer look at how SharePoint manages the backup data. First, Figure 7.6 shows the contents of the spbrtoc.xml file. You'll notice that the information maps very closely to the diagnostics shown at the conclusion of the backup process.

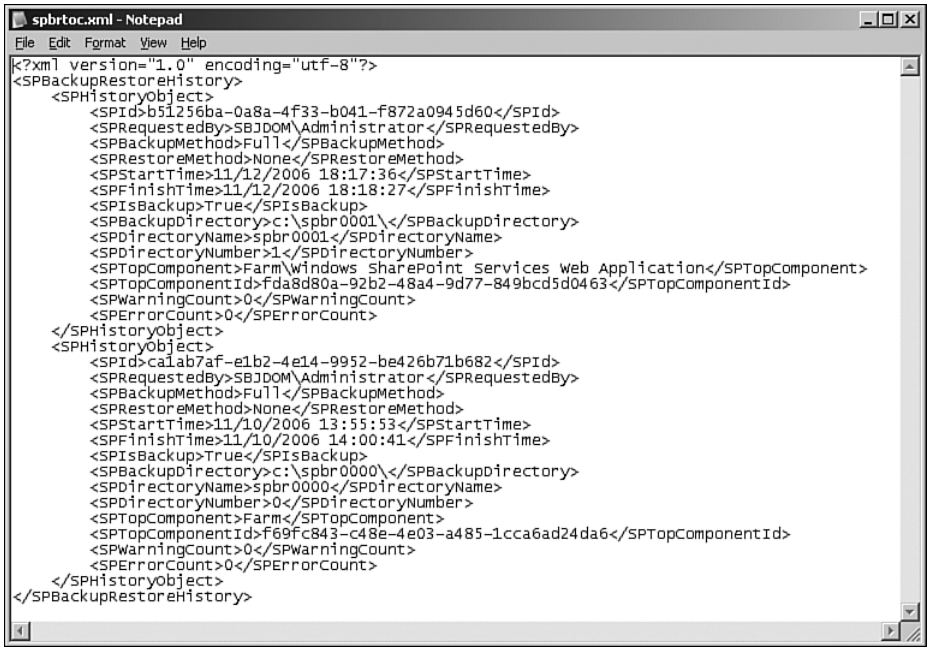

FIGURE 7.6 The spbrtoc.xml file contains information about each backup that has taken place.

More interesting are the actual contents of the backup folder. Figure 7.7 shows the files associated with a full farm backup. Again, for SharePoint Portal Server 2003 users, notice that the backup files no longer map to the specific SQL Server databases. The backup files (file extension .bak) are segmented across a collection of files. A log file, spbackup.log, gives details on the executed backup process. All of this is managed by another xml file, spbackup.xml.

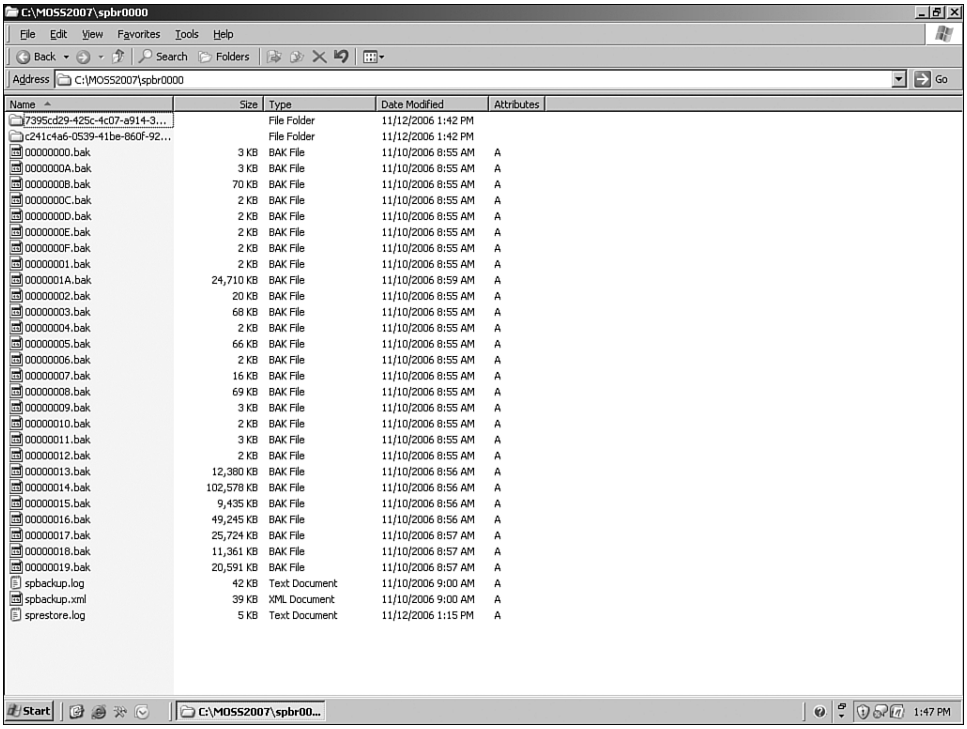

FIGURE 7.7 SharePoint spreads its backup information across a collection of .bak, .xml, and .log files.

The spbackup.xml file contains all the parameters and attributes needed to perform SharePoint backup and restore actions. Figure 7.8 shows a sample xml file. The top section, SPGlobalInformation, contains data on the executed backup. It maps very closely to the data stored in the top-level xml file. The subsequent nodes under SPBackupNode map to specific components selected using the Backup interface. This file provides a road map for the potential restore of SharePoint data. Notice that unlike the manifest file used in the previous version of SharePoint Portal Server, this xml file contains no specific references to portal URLs or database servers. This makes it easier to use these files, unaltered, to restore SharePoint on different servers.

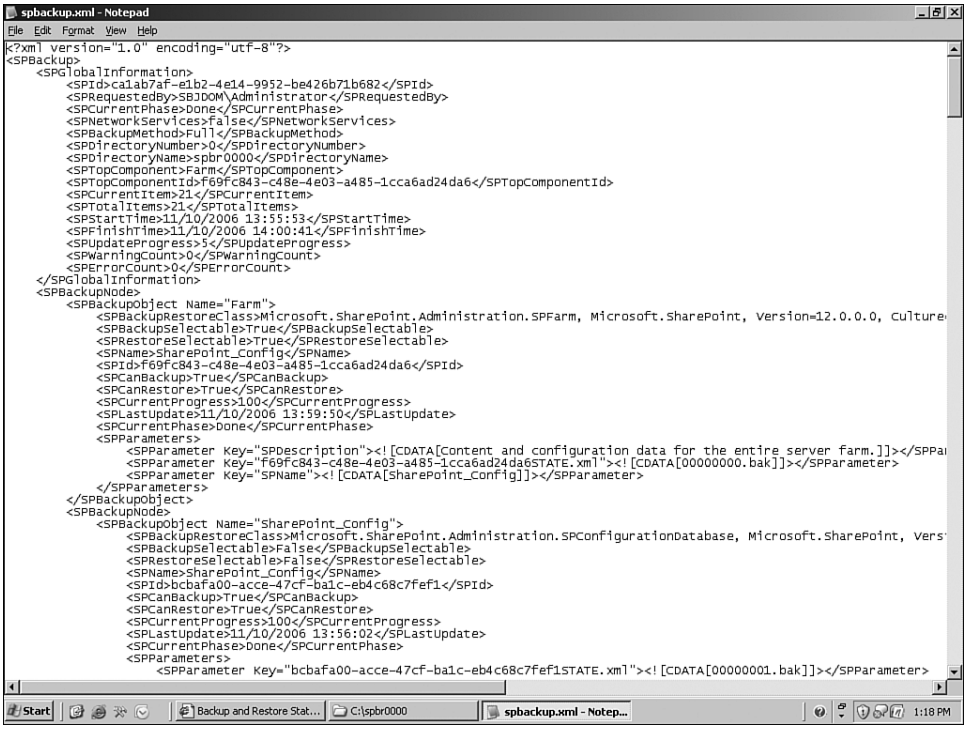

FIGURE 7.8 The spbackup.xml file contains the parameters and attributes needed to perform a restore.

> **WARNING** Do not modify the spbackup.xml file. Doing so can corrupt your backup and/or your restored farm in an unrecoverable manner.

Using the Restore Utility

Before delving into the restoration process, it is important to note that one underlying assumption is involved with SharePoint restores: the authentication mode (that is, Active Directory or another LDAP source) is the same. This is less critical for restorations in an existing SharePoint environment but may impact the recreation on new servers.

WARNING SharePoint maintains its security model (users, roles, access) in its databases. Therefore, this security model is maintained in the restoration. However, if you restore the portal to a machine that does not have access to the same authentication engine (a specific Active Directory domain, for example) the security rules previously defined are no longer valid. This scenario is most commonly seen in the restoration of a SharePoint environment onto a development server. It is important to ensure that the restoration environment has access to the same authentication engine as the backup environment.

As previously mentioned, SharePoint maintains version history associated with backup activity. This offers two immediate benefits: more flexibility for the IT staff in terms of controlling what components of SharePoint to restore, and better management of disk storage space in terms of the amount of space used. Figure 7.9 shows a sample Backup and Restore History screen.

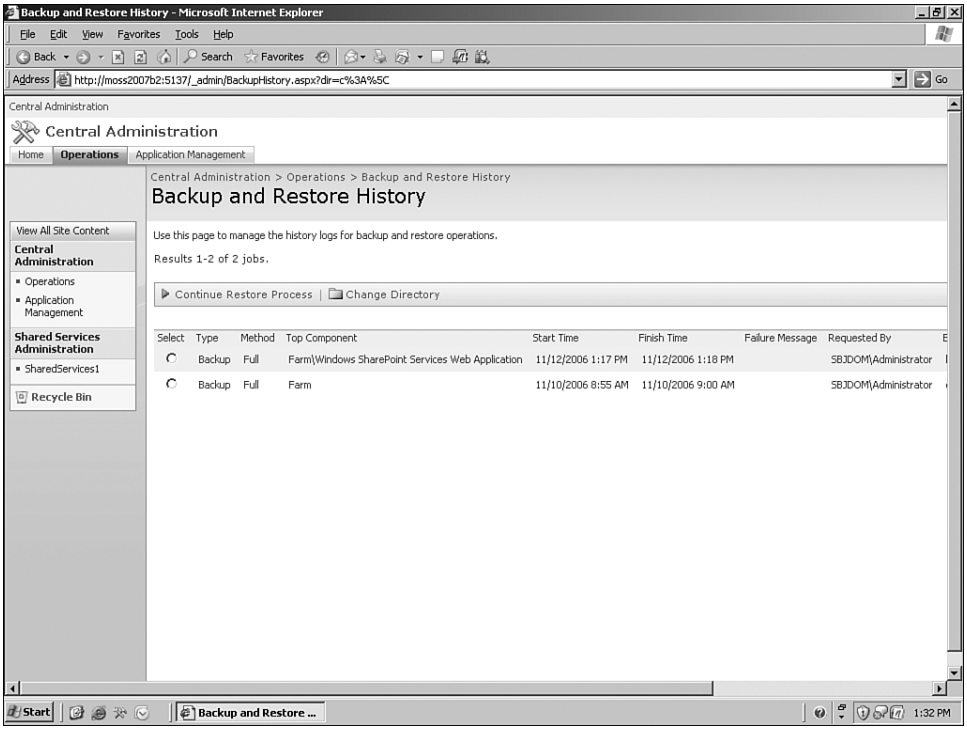

FIGURE 7.9 Central Administration provides a Backup and Restore History screen, which shows the contents of the history logs.

NOTE The information contained in the xml files previously discussed is shown on the interface to clearly identify the type of backups registered and the associated attributes. SharePoint manages a complete collection of historical files associated with backups. This feature allows on-demand restoration of potentially corrupt or disabled components (a requirement for any plan for high availability).

As mentioned previously, to successfully execute a SharePoint restore, the user must have Administrator privileges within SharePoint and have access to the files on the file system.

The SharePoint restoration process is very straightforward and consists of two steps. The first, shown in Figure 7.10, is to select the location of the SharePoint backup files. The second, shown in Figure 7.11, is the selection of a specific SharePoint backup from the collection in history. In Step 3, you are asked which components you wish to restore (see Figure 7.12). You can change some of the configuration details in Step 4 (see Figure 7.13). Once a backup collection has been selected, the restoration starts the moment Start Restore Process is clicked. The timing of the restoration is directly related to the elapsed time during the backup process. Expect a typical full farm restore to take several minutes. Once complete, the restoration will have updated the appropriate SharePoint components with the specific content selected.

NOTE What's the difference between "New" and "Overwrite" on the Restore Page?

Use "new" when migrating to a different farm or restoring such that you want to refer to a new machine or new database. Use "overwrite" when you are restoring on the machines and databases that the original farm backup refers to. "Overwrite" is used for the catastrophic restore scenario and does not give you the option to use a different machine or database name.

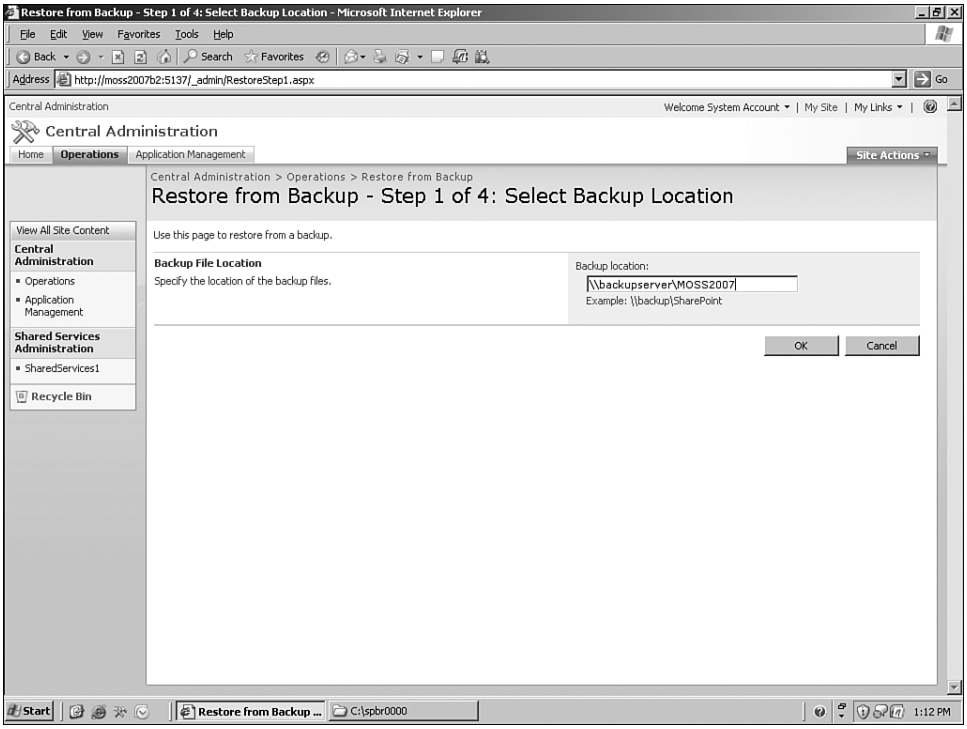

FIGURE 7.10 Restore Step 1

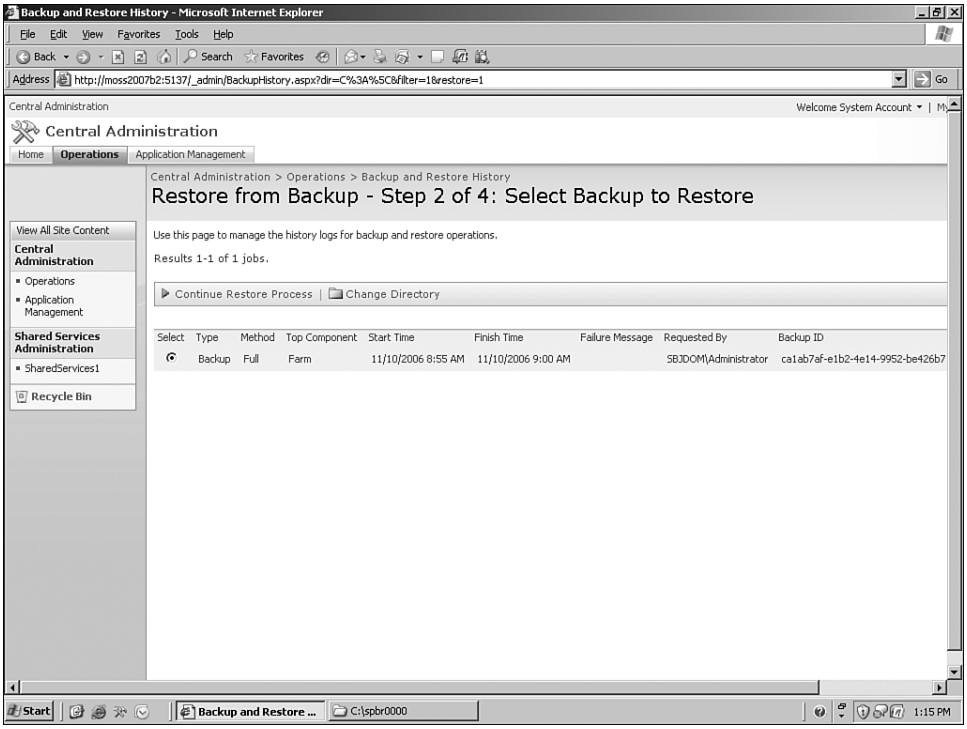

FIGURE 7.11 Restore Step 2

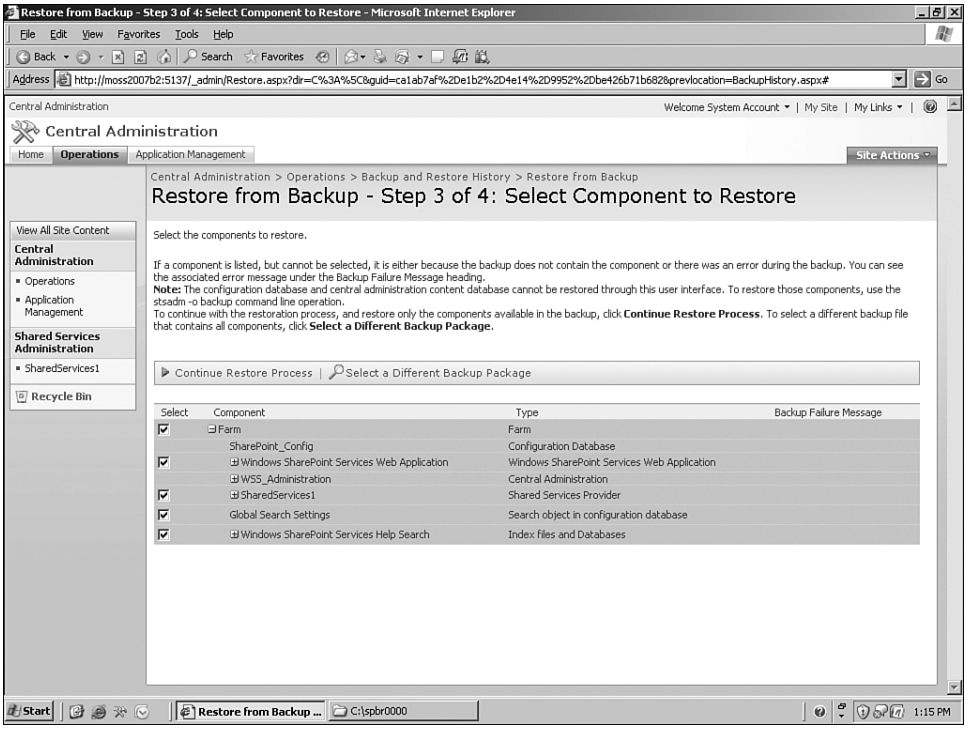

FIGURE 7.12 Restore Step 3

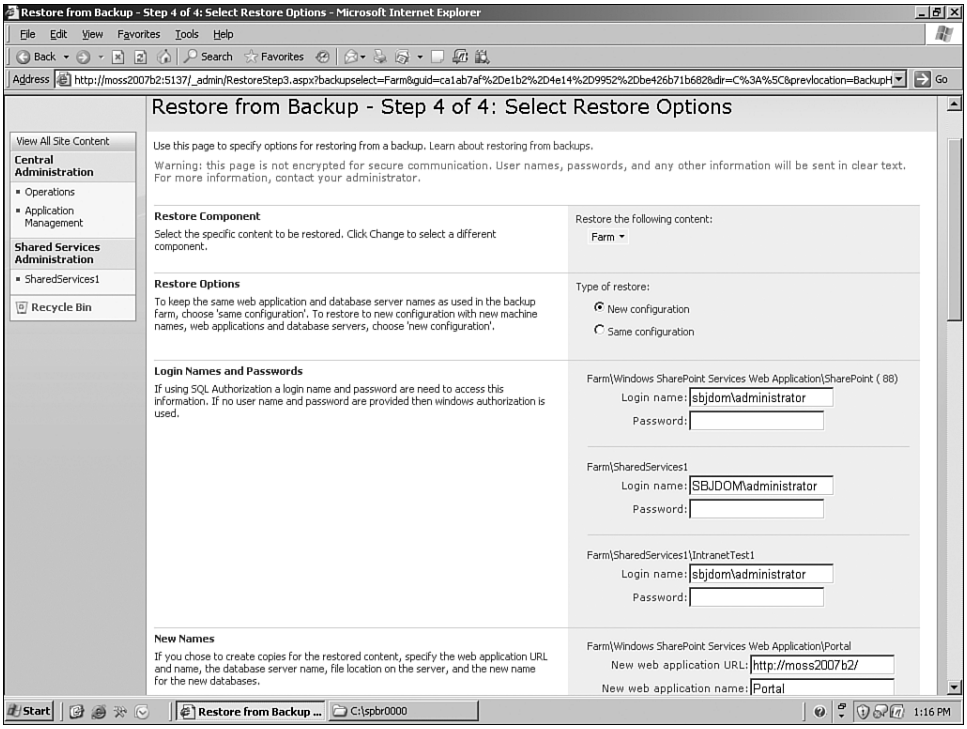

FIGURE 7.13 Restore Step 4

NOTE If a backup or restore fails, you can get details on why the operation failed in spbackup.log (for backups) or sprestore.log (for a restore) in the backup location. If errors occur during the backup/restore process, you have to delete the failed Backup/Restore Timer Job before you can run the next backup/restore process. You can delete the job from http://<adminsite:port>/_admin/ServiceJobDefinitions.aspx.

Scheduling a SharePoint Backup

One of the things you'll notice on the Backup and Restore pages is that there is no tool for scheduling backups. Much like SharePoint Portal Server 2003, there is no scheduling component in Office SharePoint Server 2007. This presents a problem for IT staff interested in ensuring that SharePoint backups are regularly obtained. As in the previous version, the best alternative is to use a simple batch file that executes the SharePoint backup from the command line. This batch file can then be scheduled using the native Windows Task Scheduler. We'll discuss the command-line backup options in the next section.

Command-Line Backup Tools

The stsadm.exe utility is probably familiar to users of WSS 2.0. It enables SharePoint administrators to back up site collections using the command line. This makes it easy to restore a site collection (or a single site) if necessary.

stsadm.exe still exists in WSS 3.0 and has been enhanced for Office SharePoint Server 2007. You can still use stsadm to back up a site collection as follows:

```
stsadm.exe -o backup
  -url <url>
  -filename <filename>
  [-overwrite]
```

For example, if I want to back up my site collection that exists at http://myserver/sites/, I would issue the following command:

```
stsadm –o backup –url http://myserver/sites -filename
c:\mybackups\
```

In addition, the stsadm.exe utility now lets you do a full SharePoint back up as you would with the Central Administration page (the command-line help text calls this the "catastrophic backup"). To issue a full or differential backup using the command line rather than the Web UI, simply use the following format:

```
stsadm.exe -o backup
  -directory <UNC path>
  -backupmethod <full | differential>
  [-item <created path from tree>]
  [-percentage <integer between 1 and 100>]
  [-backupthreads <integer between 1 and 10>]
  [-showtree]
  [-quiet]
```

For example, to back up my entire SharePoint farm, I could issue the following command:

```
  stsadm -o backup -directory \\backups\sharepoint
-backupmethod full
```

This would perform a full backup on my SharePoint farm and write to the Backup and Restore History on the Central Administration page. Then I could use either the command line or the Central Administration UI to restore from this backup. Backups done via the Web UI or the command line are indistinguishable.

Using the stsadm utility is very useful for regular backups because you can use the Windows Task Scheduler to create a recurring backup job.

Two-Stage Recycle Bin

Needing to recover a single item is a more commonplace situation than having to recover from a full-fledged disaster. SharePoint now provides an "undelete" feature to allow end users to recover accidentally deleted files, documents, list items, lists, and document libraries without running a content-database-level backup and restore. This saves the SharePoint Administrator(s) time and hassle because they can easily recover files for end users without having to initiate a full-fledged backup and restore process. In fact, in most cases, users simply recover things themselves.

When a user empties the Recycle Bin, the deleted items move to a second-level Recycle Bin, which can easily be recovered by the administrator, provided the items have not been purged.

The global settings for the Recycle Bin are part of the Web Application General Settings. These settings are accessed through the Central Administration Application Management (see Figure 7.14). The Recycle Bin settings are at the bottom of the general settings page (see Figure 7.15).

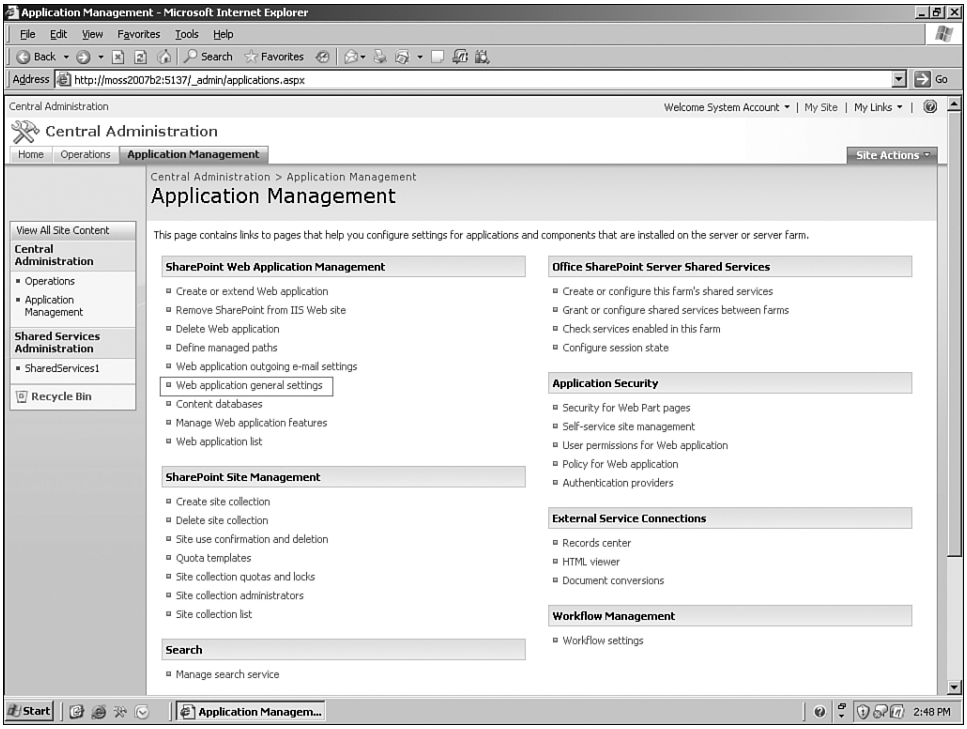

FIGURE 7.14 The global settings for the Recycle Bin are part of the Web Application General Settings.

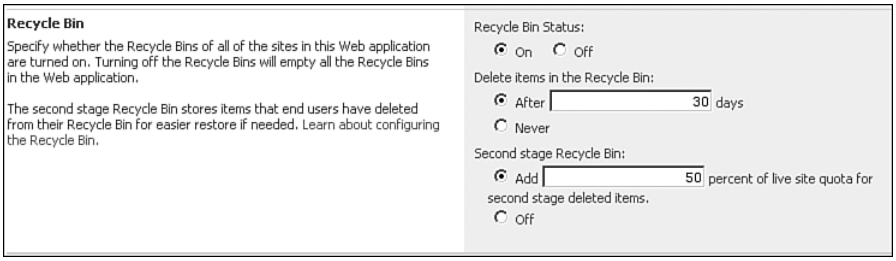

FIGURE 7.15 The global Recycle Bin settings enable you to turn the feature on and off and to set the retention timeframe for items.

The Recycle Bin is a Web application setting, which means that it can only be enabled or disabled for all of the site collections served by the Web application. If you turn it on, it's available on *all sites in all site collections* for that Web application.

We recommend that you configure the Recycle Bin to a size that is a percentage of the overall site quota and set an "auto-clean" schedule (the default is 30 days) for permanent file removal that fits your business needs.

The first level of the Recycle Bin is the user-level Recycle Bin (see Figure 7.16), which is accessible by site users. It provides a site-level view of deleted content and contains all items deleted from a particular site.

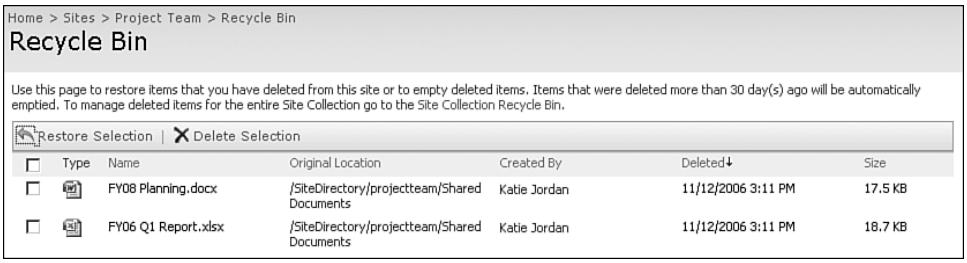

FIGURE 7.16 The Recycle Bin enables an end user to restore deleted items.

NOTE The Recycle Bin works by capturing delete events. If items go missing due to errors, data corruption, or other problems, they will not be recoverable via the Recycle Bin. This is why a full backup process must exist. They Recycle Bin is a convenience item for users who accidentally delete a file or other item.

NOTE The first-level Recycle Bin counts toward the site's maximum quota.

The second level of the Recycle Bin is the administrative Recycle Bin (see Figure 7.17), which is accessible by site collection administrators. It provides a site collection-level view of deleted content and contains all items deleted from a particular site collection. In effect, SharePoint administrators are no longer responsible for maintaining replica environments for item-level restores. In addition, inadvertent site deletions can be managed through the use of custom event handlers that automatically back up a site prior to deletion. Both offer significant support time reductions.

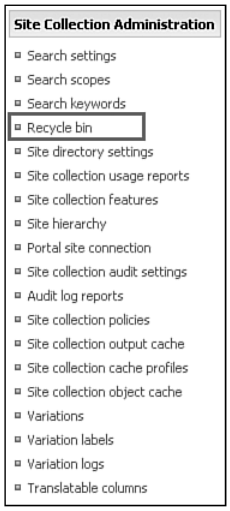

FIGURE 7.17 Site collection-level Recycle Bin settings are available for top-level sites.

SQL Server Backup

Microsoft SQL Server Backup and Restore is typically used by large organizations because they already have SQL Server Tools or offsite data centers. It's also because the person/group responsible for the databases is a DBA, rather than the administrator of Office SharePoint Server. If you are in a large organization where this situation is likely, we recommend this option. We'll leave the steps to back up SQL Server to the DBA.

SQL Enterprise Manager can schedule backup tasks, which enables the DBA to automate the backup process. We recommend that the DBA(s) responsible for the SharePoint databases get proper training on the structure of the SharePoint databases.

It is important to note, however, that only the configuration and content databases get backed up. The next section describes other important items you need to back up.

What's Not Covered by a SharePoint Backup

As powerful as the SharePoint Backup tool appears, it does not contain all the elements necessary to recreate your SharePoint environment. While SharePoint stores all of its content in SQL Server (documents, images, text, security, site metadata, and so on), there is a collection of files on the file system that are not in the database, and therefore they do not get properly captured in a backup.

The following items play a pivotal role in the generation of SharePoint pages but are not covered in the SharePoint backup:

- Third-party or custom Web parts
- SharePoint site definitions and XML files
- SharePoint .aspx template pages
- SharePoint script files

The first step in a successful recovery process is the restoration of the environment using the Backup/Restore tool. The subsequent steps involve bringing in any elements that were not captured in the SharePoint backup. The biggest piece of this is the inclusion of nonnative SharePoint Web parts. Whether purchased from a vendor, acquired online, or custom built, Web parts must be registered in a specific way in order for SharePoint to consider them "safe." While this is not a chapter on deploying SharePoint Web parts, let's quickly touch on the two main requirements for a successful Web part deployment.

First, the associated DLLs need to be placed on the file system, either in the underling BIN directory of the virtual server or in the Global Assembly Cache (GAC). Second, the Web part must be placed in the list of Safe Controls. This is done in the SharePoint Web.config file. If you examine this file, you'll notice a collection of Web part registrations under the SafeControls node. All Web parts, even native SharePoint Web parts, must be registered here. The challenge from a DR perspective is that this file, Web.config, and the associated Web part DLLs are not captured in a SharePoint backup.

If these steps are not executed in a restoration process, the SharePoint pages that contain the respective Web parts will not properly generate (they may not generate at all, redirecting you to a generic error page). Therefore, it is important to take inventory of all nonnative Web parts used and to store the associated CAB or install files for future restoration. You

may even do this in the SharePoint document library to ensure that they do get captured in the SharePoint backup process.

In addition to Web parts, other system files might be altered through standard or advanced SharePoint customization. These files include underlying xml, aspx, and script files. All SharePoint file system files reside in the following directory:

```
C:\program files\common files\microsoft shared\Web server
extensions\12
```

It is important to note any changes made to files in this directory and to take appropriate measures to document the alterations in your DR plan. Table 7.1 shows an example of a section that you should include in your disaster recovery plan so that you can track these changes.

Table 7.1 A Change Log Is Important for Customizations Made to SharePoint Because There's No One Place for Custom SharePoint Artifacts

Date	Description	Location	Made By	Approved By
4/20	Updated portal.js to accommodate customization	\12\TEMPLATE\LAYOUTS\ portal.js	mcardarelli	sjamison
5/18	Added logo.jpg file	\12\TEMPLATE\IMAGES\ logo.jpg	mcardarelli	sjamison

Outside the bounds of the SharePoint-related files necessary for full portal restoration, but equally important, is the need to have a plan in place to ensure that any replica environment stays consistent with the software, patches, and third-party Web parts deployed in the original environment. A disaster recovery plan should always denote the current state of the SharePoint servers and should be updated as changes are made.

NOTE Because the MOSS 2007 DR tools are somewhat anemic, many enterprise customers opt to purchase a third-party backup and restore tool. For example, DocAve 4.1 from AvePoint enables SharePoint administrators to monitor multiple SharePoint 2003 and 2007 farms across the globe, perform item/subsite/site-level data backup, full-fidelity restore, within or cross-site content management, seamless data archiving, and site-specific "one-switch" disaster recovery from a single Web UI.

Key Points

This chapter has provided some key tips regarding your SharePoint disaster recovery plan. In summary:

- SharePoint Disaster Recovery requires a well-maintained, documented plan.
- IT Administrators now have greater control over what components within SharePoint can be restored.
- As your SharePoint environment matures, it will become increasingly complicated to restore SharePoint and all its connected elements.