Part I

Overview of Excel Services

1

Introduction to Excel Services

Microsoft Office Excel has existed on the desktop for more than 20 years. It is by far the most broadly used spreadsheet tool available. And in Microsoft Office 2007, Excel is significantly enhanced. In Figure 1-1, you can see that quite a bit has changed in the last 20 years.

Today, spreadsheets are used more broadly than ever. Spreadsheets are often business- and mission-critical, and calculations range from trivial to those that require hours of compute time. There is a whole new set of needs for sharing spreadsheets broadly, incorporating them in web-based applications, and meeting compliance and regulation demands. In Office 2007, Microsoft has introduced a new product to address these and other needs: Microsoft Office SharePoint Server. Excel Services is a part of Office SharePoint Server that extends Excel to the server, enabling server-based workbook calculations, browser-based access to workbooks, and a web services interface to workbooks.

As with any new technology, there is a lot to learn, and many questions to answer. This book will help you get acquainted with Excel Services. The book is intended for people learning about, deploying, and using Excel Services. After reading the book, and going through the exercises, you will have a firm understanding of the benefits that Excel Services offers end users, how administrators can set up and manage Excel Services, and how developers can leverage Excel Services programmatically.

This chapter introduces Excel Services. It first explains what problems Excel Services was designed to address. It then discusses what Excel Services is at a high level, and the key scenarios that are targeted with the initial release. Finally, it attempts to preempt any misconceptions by discussing what Excel Services is not.

The remaining chapters of this book guide you through installing, setting up, and using Excel Services.

> *This book does not cover what is new in Excel 2007, nor does it cover Microsoft Office SharePoint Server broadly. The focus is on Excel Services functionality.*

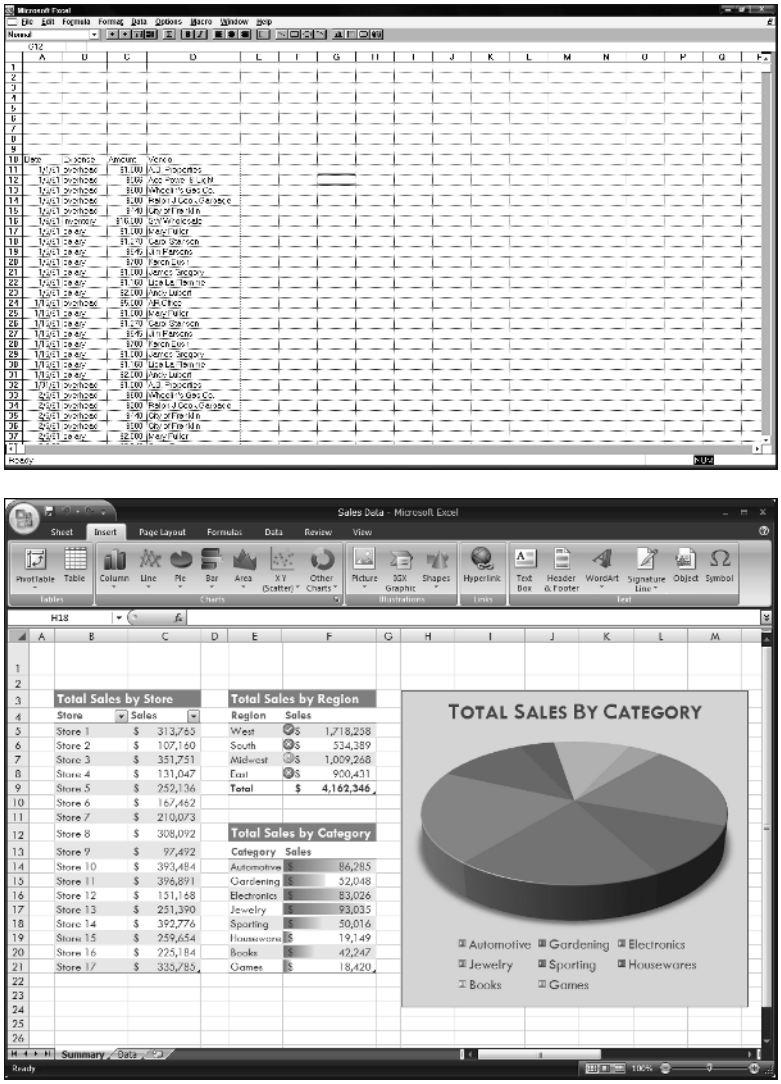

Figure 1-1

Who Hasn't Heard of Excel?

Excel is by far the most popular spreadsheet tool available. It is used by millions of end users each day. It is used to solve a very wide range of problems — solutions that span the gamut from simple list-keeping to mission-critical trading solutions driving multimillion dollar trades on Wall Street. Everyone who has an M.B.A. knows how to use it (and probably was already introduced to it somewhere back

in middle-school). Every financial department relies on it. Managers in business in every industry and at every level of the organization make decisions based on the numbers presented by and calculated in Excel.

Whether it is being used for forecasting sales, analyzing a return on investment, developing a budget, tracking an investment portfolio, or calculating commission models, it is safe to say that Excel is at the heart of the business world. Excel formulas are the lingua franca, and much of the logic that drives businesses worldwide is captured in Excel models.

Given how critical Excel is for business, it is not surprising that there are ever-increasing needs for better management, distribution, and incorporation of spreadsheets in larger applications. The next section takes a look at these needs in more detail.

Why Excel on the Server?

Spreadsheets are used everywhere, and as they become more important to an organization, one or more of the following needs arise:

- There is a need to distribute the spreadsheet broadly.
- There is a need to control and manage the distribution and life cycle of the spreadsheet. (This is especially true in this age of compliance and regulations.)
- The spreadsheet needs to be viewed in the context of additional data as part of web applications and business intelligence (BI) dashboards.
- The results of the spreadsheet calculation need to be incorporated in other applications.
- The spreadsheet calculation takes a long time, preventing users from doing other things on their computers while it is processing.

This list is not all-encompassing. There many other needs that exist for users and organizations working with spreadsheets. This book focuses on the needs addressed by this first version of Excel Services. The "What Excel Services Is Not" section later in this chapter covers some of the things Excel Services does not do in this release.

The Problems with Distributing Spreadsheets

Today, when a spreadsheet author is working on an Excel workbook and wants to share it broadly (either for review or to distribute the final results), the most common way to do it is through e-mail. Sending an e-mail with an attached spreadsheet is simple and straightforward, but it also poses a problem. Every recipient of the e-mail with the attached spreadsheet receives his or her own copy of the spreadsheet. Now, instead of having one copy of the spreadsheet that everyone is looking at, there are as many copies as there were recipients on the To line of the e-mail. If the spreadsheet author makes a change to the workbook, he or she must send out an updated copy. But even then, the author has no way of verifying that everyone is looking at the latest copy sent. And, because each recipient has his or her own copy, they can each make changes to the workbook, making it out of sync with the version that is controlled by the spreadsheet author. Figure 1-2 shows this process.

The same problem holds true if the spreadsheet is placed on a file-share or document-management system. Each user opens a local copy of the spreadsheet, which he or she can then modify, making it out of sync with the sanctioned version.

Another problem that is exists when sharing spreadsheets this way is security. The spreadsheet may contain certain areas that its author does not want to share broadly (for example, the author may be interested in sharing the aggregated summary and not the detailed data on a different sheet in the workbook). In fact, in many cases, the actual model represented in the workbook and the formulas that comprise it are considered proprietary or a trade-secret that the spreadsheet author would like to protect. When all the recipients receive a copy of the entire spreadsheet file, it is practically impossible to guarantee that these elements will remain secured.

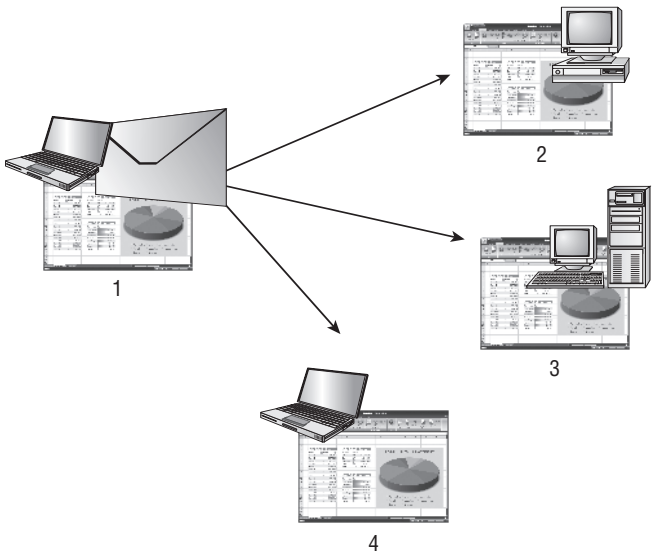

Figure 1-2

Controlling Spreadsheet Distribution

Distributing a workbook broadly may require additional control and management. The author may need to have the workbook reviewed and approved prior to it being made available broadly. Notifications may need to be sent out to all those tracking updates to the workbook. And after the spreadsheet has been distributed, there may be a need for an audit log to track who saw the spreadsheet, which version they saw, and who changed it. These types of requirements may be part of internal procedures, or mandated by external industry and government regulations. In addition, the workbook life cycle is not over when it is distributed. There may be records-management or document-retention policies that need to be enforced. For example, automatic backups and snapshots may need to be retained for archival purposes.

These needs usually span more than just workbooks. Documents (such as legal documents, human resource records, and product specifications), presentations, and e-mails all need to be controlled and managed. But there is no doubt that these requirements are especially relevant for workbooks that may include official financial reports, organizational budgets, trade records, and so forth.

Workbooks stored on local machines, copied from file share systems, and distributed as e-mail attachments are hard to track, control, and manage. This creates the potential for inefficiency and costly mistakes when people are basing their work and decisions on the wrong workbook. In this age of compliance and regulations, the concerns are even more acute.

The needs discussed thus far focused on distributing workbooks. The next two sections focus on incorporating workbooks, the models they represent, and the results of those models in web-based applications, dashboards, and custom solutions.

Incorporating Spreadsheets in Dashboards

The Excel client is perhaps the most used tool for data analysis and reporting. Whether the data originates in Excel or in some back-end system, users find a way to get it into Excel so that they can manipulate it, analyze it, and format the data and their findings for printing and sharing more broadly.

But when you look at what are considered the BI solutions within companies, Excel is usually not formally part of them. In fact, it is often explicitly called out as not being part of the sanctioned data analysis and reporting solution. Why? Here are two primary reasons.

BI solutions are essentially used for decision support, and it is critical that the numbers viewed in them are 100 percent trusted and secure. As mentioned previously, distributing workbooks as e-mail attachments may result in errors due to people looking at the wrong version or changing data they are not supposed to. When you look at typical BI solutions, they usually include a portal element that enables web-based report and analysis distribution. Distributing reports and analysis this way provides easy broad access, as well as the security and control necessary to ensure that the consumers of a report are looking at the correct report, can only access the data they are privileged to access, and cannot change (by accident or on purpose) the sanctioned data and results.

In addition, BI solutions often include dashboards, which enable the visual display of critical data in summary form. For example, a management dashboard may provide a complete view on the operations of a project, department, or company A customer dashboard may provide a sales representative with all the information and data available about a customer. The key to dashboards is the ability to combine different views on data, different data, and different content types all on the same page to give you a complete view for tracking information and making decisions. Although you can embed different data and objects in Excel workbooks, and build dashboards directly in the workbook, most BI dashboards are web-based. This allows for the greatest flexibility in incorporating different content types.

Leveraging Spreadsheet Models in Custom Solutions

The business logic that drives many applications is first defined in Excel workbooks. Excel workbooks are used to model such things as sales commissions, pricing and discounts models, and stock trades. Excel affords rapid development of these models in the hands of users who understand the business

logic. The workbooks then serve as a specification for the developers who recode the logic in a formal programming language, and incorporate it in the target applications. This process is error-prone and costly. The developer needs to mimic the Excel model exactly and guarantee the same results. The cost of recoding and maintaining the code with every change to the workbook are high.

Excel can be called programmatically, but this solution is often not robust enough. This is especially true when the application calling the model must scale to support many such calculations with high availability. Excel is designed as a client application and, as such, there are limitations that affect these scenarios. There is, therefore, a need to support leveraging the business logic defined in Excel workbooks programmatically as part of scalable applications.

Offloading Spreadsheet Calculations to Servers

Some workbooks take a long time to calculate. Although many workbooks calculate instantaneously, there are also workbooks that take minutes and even hours to calculate. In some cases, calculating a single instance of the workbook may be very quick, but you have to repeat the calculation many times with different inputs. A great example for this is calculating risk using a Monte Carlo simulation. Though each calculation of the workbook may be short, it could require thousands of iterations with different input to return the needed results.

There is no doubt that the need to distribute workbooks is applicable to a broader set of users than dealing with workbooks that take a significant amount of time to calculate. But the cost this issue creates is significant. When these long-running calculations are taking place, you're blocked from using your computer for any additional work. When the calculation takes minutes or even hours, the ramifications are obvious.

The following section introduces you to Excel Services and how it can help address these needs.

What Is Excel Services?

Excel Services is a new server-side technology that is shipping in Microsoft Office 2007. It is not a product onto itself, but rather an integrated set of features that is part of the new Microsoft Office SharePoint Server 2007 (more on this product in the next section).

At its core, Excel Services enables you to calculate workbooks authored in the Excel client on a server, and distribute the updated results through either a browser-based interface or a programmatic interface in the form of web services. Excel Services extends the Excel client to the server, and, together with the rest of Office SharePoint Server functionality, provides an answer to the problems and needs described earlier.

With Excel Services, you can broadly distribute workbooks in a secure, controlled, and managed way. You can guarantee that only the correct version of the workbooks you author are accessed; that the details and formulas in the workbook are hidden and secure; and that the workbook has gone through the appropriate approval processes and is managed by the records-management policies of the organization. You can incorporate workbooks as integral parts of BI dashboards and portals, and offload workbook calculations from the client to free up desktop machines. And you can use workbook models and their results directly

inside applications. All of this is fully integrated with the Excel client, leveraging all the capabilities and knowledge already out there.

Excel Services is designed to be a server product that supports many concurrent users, workbooks, and requests. It leverages server hardware and advanced configurations, from single-box deployments to multiple-tier and multiple-box scaled-out farms. It can work inside of the firewall as part of a point solution for a department, or as an enterprise-wide solution. It can also be deployed as part of an extranet topology to support scenarios that extend beyond the organization to external partners and customers through the Internet.

Next, you will take a step back and take a quick look at the rest of the functionality in Office SharePoint Server.

Microsoft Office SharePoint Server

Before going any deeper into Excel Services, you need to first zoom out and take a look at the broader product of which it is part: Microsoft Office SharePoint Server 2007. This, too, is a new product from Microsoft. In a nutshell, Office SharePoint Server is Microsoft's portal, document management, and business and collaboration application server all in one. The primary interface is through the web browser, as shown in Figure 1-3, but Office SharePoint Server functionality can also be accessed directly from the client.

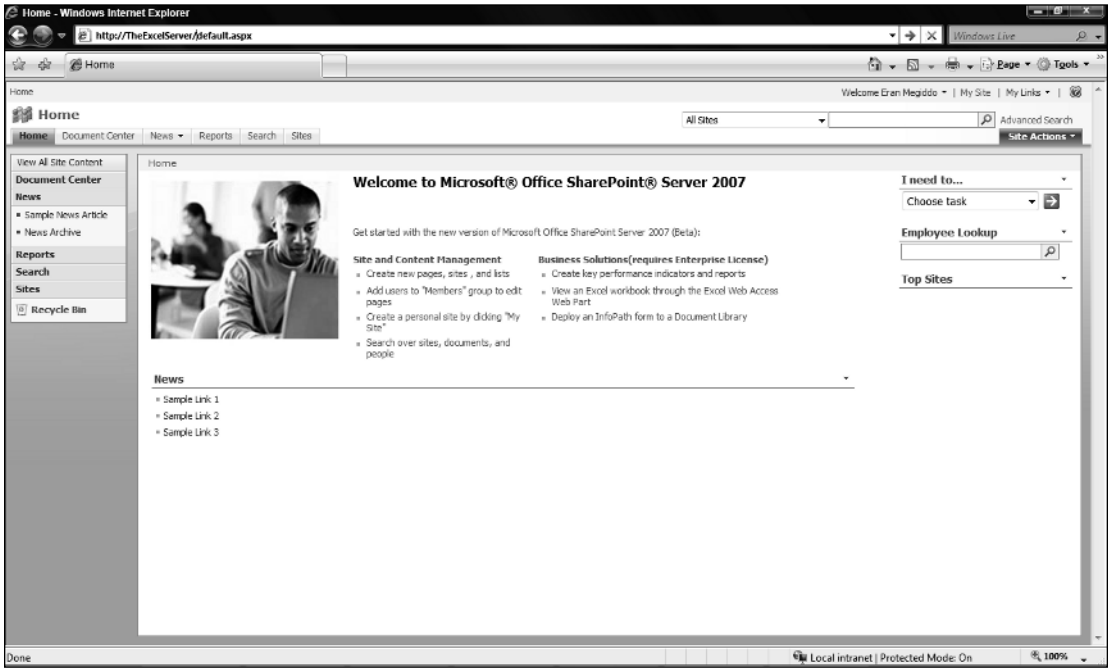

Figure 1-3

Office SharePoint Server provides the core storage, security, and administration framework. Beyond these, it includes a core set of six integrated capabilities:

- *Collaboration* — Document, task, and calendar sharing, and support for blogs and wikis
- *Portal* — Enterprise portal templates, personalization, news and content aggregation, and social networking.
- *Content Management* — Core document management (for example, check-in, check-out, and versioning), records management, and web content management
- *Search* — Enterprise-level search, including contextual relevance and rich people and data searching
- *Business process and forms* — Web-based forms and line-of-business application integration
- *Business Intelligence* — Dashboards, key performance indicator lists, report libraries, and web-based spreadsheets through Excel Services

As you can see, Office SharePoint Server is an extremely rich product. Each one of the six areas could be the subject of an entire book unto itself. This book does not attempt to cover all the functionality provided by Office SharePoint Server, neither in breadth nor in depth. The focus is Excel Services.

With that said, Excel Services is an integral part of Office SharePoint Server, and, as such, the solutions to the needs described earlier rely on many of the other capabilities of the product. This book will freely cross the line between features that are specific to Excel Services and features that are part of the Office SharePoint Server as a whole. The distinction is of no significance, because one set of features does not exist without the other.

As the name suggests, the heart of Office SharePoint Server is what used to be known as Microsoft SharePoint Portal Server. This is Microsoft's portal offering. Like SharePoint Portal, Office SharePoint Server is based on and requires Windows SharePoint Services. Windows SharePoint Services provide the base storage repository, document (and workbook) management functionality, as well as the end-user and administrator web user interface. Windows SharePoint Services also provides the base programmability and extensibility model for developing custom web-based applications, which in turn is based on ASP.NET 2.0.

In Office 2007, both Windows SharePoint Services and the technology for SharePoint Portal Services are new versions with new functionality. In addition to these core technologies, Office SharePoint Server integrates what used to be the Content Management Server for web content management, the new Forms Server functionality for web-based forms and forms management, and Excel Services. Together, these provide the platform and capabilities described. The following sections cover a few of the core capabilities and key Office SharePoint Server concepts in more detail.

Security

Office SharePoint Server provides two fundamental building blocks for security: authentication and authorization. A user or application looking to calculate and view a workbook with Excel Services is first authenticated through Office SharePoint Server. The authentication is based on ASP.NET 2.0 authentication, and a number of methods are supported. These include Windows-based authentication, forms-based authentication, and anonymous access. The authentication model is pluggable, and third-party authentication solutions such as Lightweight Directory Access Protocol (LDAP) directories can be used.

After a user or the calling application is authenticated, authorization is done against the resource being requested. This can be any resource in the SharePoint store. The scenarios and examples in this book cover libraries of workbooks, specific workbooks, and dashboards. Office SharePoint supports specific rights, and users or groups of users can be granted these rights. For example, a user may have the rights to open, view, and edit a workbook in the browser and the client, or the user may have only the right to view the workbook in the browser.

Office SharePoint Server also includes a single sign-on (SSO) store. This provides an additional mechanism by which users are authenticated, and is especially relevant for scenarios that include workbooks connected to external data sources. The SSO store infrastructure is pluggable, and third-party solutions can replace the built-in solution.

Chapter 8 describes the security model, the various authentication mechanisms, and the specific user rights in more detail. Later in this book, you will see how Excel Services uses this infrastructure to load workbooks, query external data sources, and provide results.

Web Part Pages

A key underlying concept for Office SharePoint Server is *Web Part Pages*. Practically all the web pages served by Office SharePoint Server are Web Part Pages. This is an ASP.NET 2.0 technology that enables you to place numerous web parts (equivalent to web controls) on a single web page, and defines how these web parts are laid out, as well as how they can interact with one another. Office SharePoint Server ships with a large selection of out-of-the-box web parts, and developers can create their own and deploy them to the server. Out-of-the-box web parts enable you to display documents in a specific document library, views on a task list, RSS content aggregation, a list of links, and simple Hypertext Markup Language (HTML) or rich text content, among other things. Excel Services has a web part component as well, through which spreadsheets are rendered in the browser.

Web parts can have numerous properties that control and customize them. Many of these settings can be personalized so that different users will receive different behaviors when they view the same page. Using this functionality, it is possible to set up a page that includes a list of tasks, and users logging in will see the tasks assigned to them.

Web parts on the same page can also interact, passing data between them. For example, the author of a Web Part Page can connect a web part that lists workbooks in a document library to an Excel Services web part. A user viewing the page can select from the list of workbooks, and the selected workbook is displayed. Behind the scenes, the workbook URL is passed between the two web parts.

Web parts are the foundation for building composite applications as well as BI dashboards.

Content Storage and Management

Office SharePoint Server provides the default content store for Excel Services scenarios. The core storage container is a *document library*, which, in turn, is associated with a site. Document libraries are conceptually equivalent to file system folders. As shown in Figure 1-4, they can contain virtually any file, including images, Word documents, PowerPoint presentations, Excel workbooks, and web pages. Document libraries can also contain additional document libraries or subfolders.

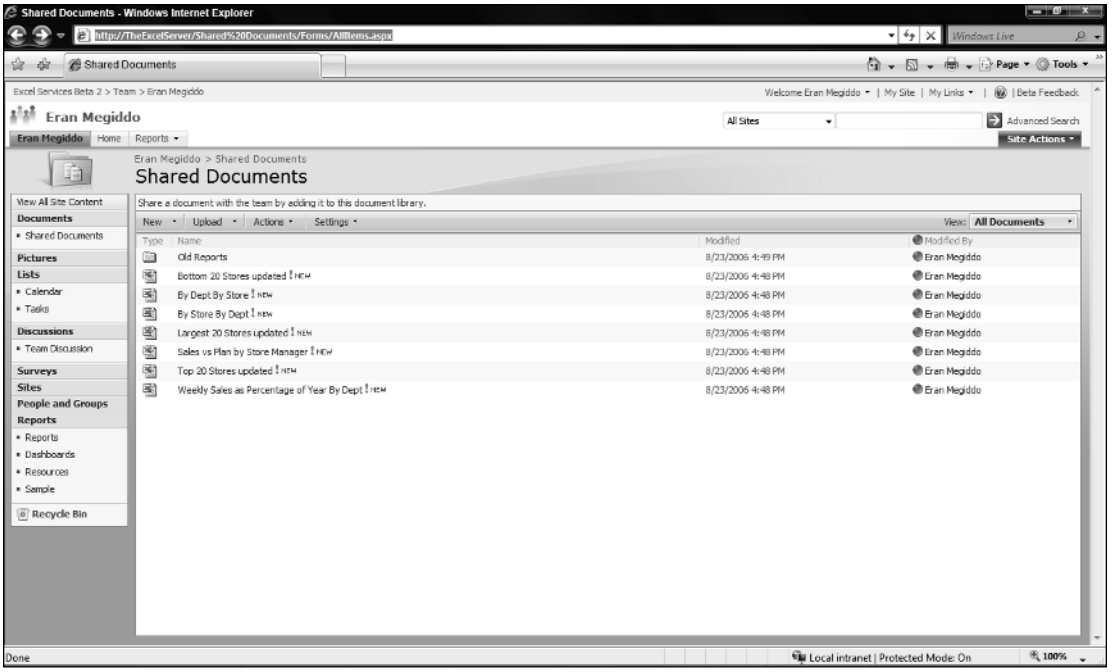

Figure 1-4

The content security and management functionality that is part of Office SharePoint Server accrues to any content stored in document libraries. As described previously in the "Security" section, users can be granted specific rights to a specific document library or a specific spreadsheet (or any other file) in a document library. Files in a document library can be controlled through enforced check-out and check-in procedures. Versions can be automatically kept for each file. These can be major versions and/or minor versions. Document approval settings can govern whether a file is made available broadly prior to being approved by the moderator of the document library. Notifications can be set to alert (through e-mail or RSS feeds) users when a new file is added to a document library, or when an existing file is updated or deleted.

More advanced document management capabilities include the following:

- ❑ Workflow
- ❑ Records management policy
- ❑ Auditing support

Office SharePoint Server's workflow support is built on Windows Workflow services. Workflows can be associated with a document library such that any time a file is added, updated, or deleted, a workflow is triggered. Workflows can also be triggered manually by users or programmatically. Office SharePoint Server includes a number of out-of-the-box workflows, and you can develop additional custom workflows using SharePoint Designer or Visual Studio. Following are the included workflows:

- ❑ Collect Feedback
- ❑ Approval

- Disposition Approval (to support retention policies)
- Collect Signatures
- Translation (for managing human document translation)
- A three-state workflow that assigns items in SharePoint lists between people based on changes to a specific column

In East Asia, there is also a more complex approval workflow to support specific routing options relevant in those markets. Figure 1-5 shows the web-based user interface for setting up workflows.

To further support records management, compliance, and regulation needs, Office SharePoint Server enables you to define policies for such things as document retention. For example, you can define which documents in a certain document library will be automatically archived for a given period of time without the possibility of anyone deleting them. You can specify the date when these documents are to be removed from the archive and deleted. As with workflows, there is a web-based user interface (as shown in Figure 1-6). Users can choose to customize the out-of-the-box policies, or they can be extended programmatically.

The document management functionality is completed by an auditing capability. Practically any event associated with a document can be audited, including documents being created, edited, viewed, and deleted. You can extend this capability programmatically so that custom events can be audited.

Office SharePoint Server also extends content management to include web content management by integrating the functionality of the Content Management Server.

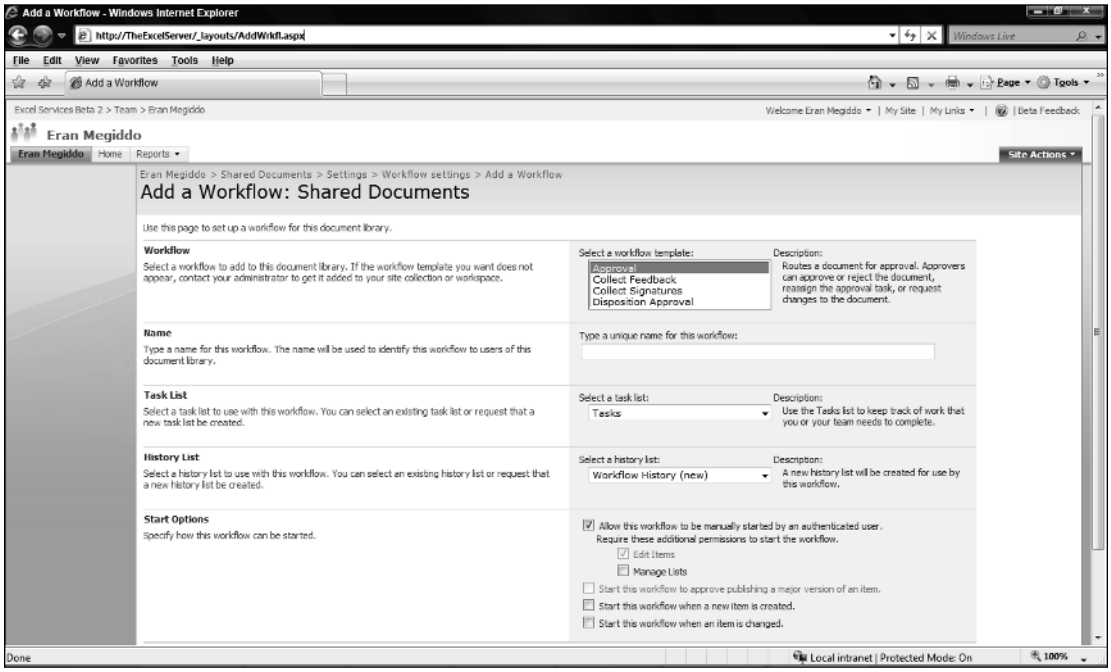

Figure 1-5

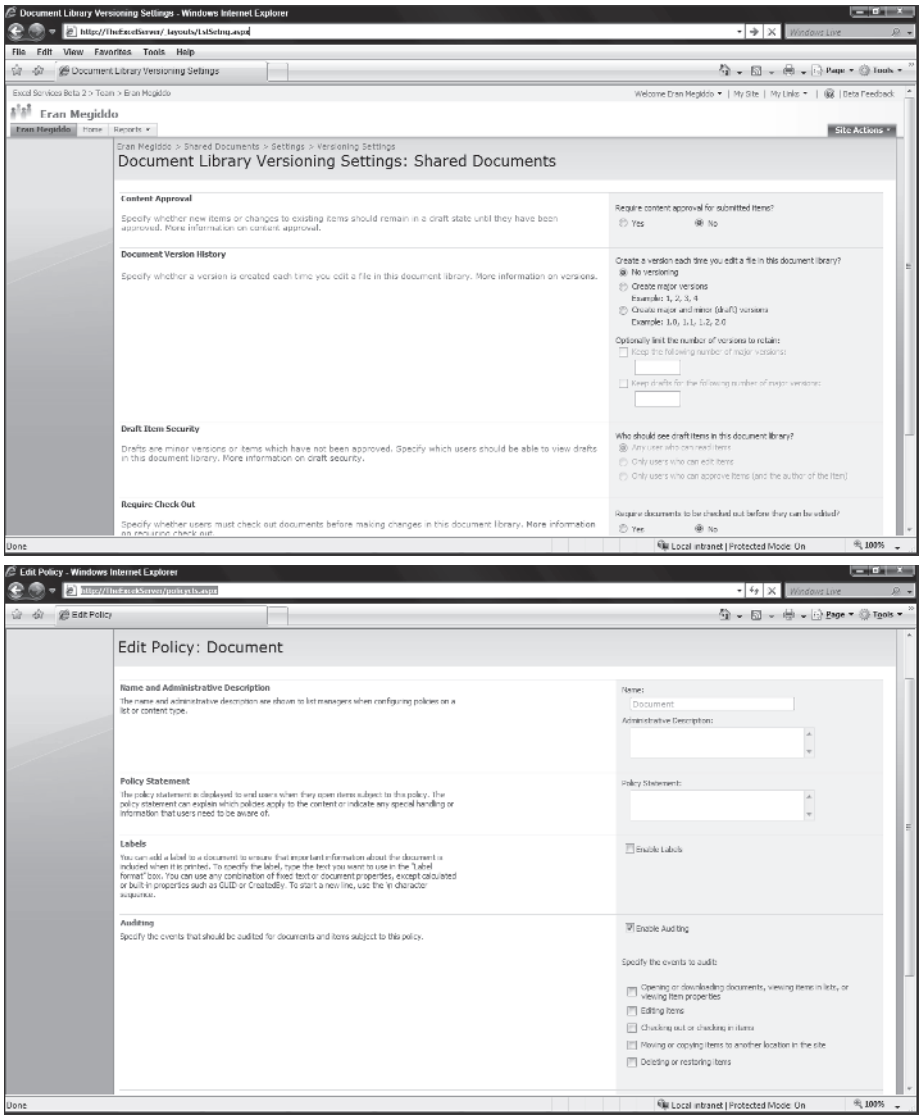

Figure 1-6

Collaboration

Office SharePoint Server provides a hub for storing and managing content, as well as for collaborating on it. Collaboration functionality includes support for discussion groups, shared task and calendar management, wikis, blogs, integration with e-mail systems, and (together with the Live Communications Server) support for presence awareness. The latter enables users to directly interact with one another from within the context of their dashboard, collaboration site, or application, as shown in Figure 1-7.

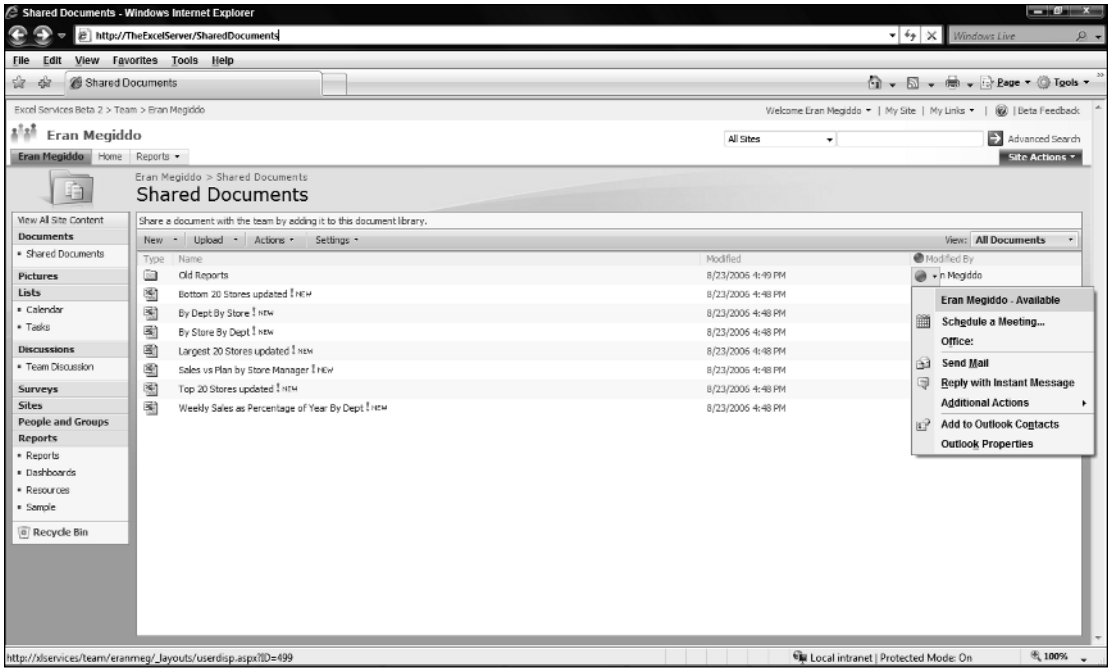

Figure 1-7

Users can collaborate on a document or set of documents. They can collaborate on tasks and entire projects. Entire sites and applications can be hubs for collaboration, as is the case with a blog or wiki. For example, you can set up a fully featured wiki by simply selecting that template when creating a new site. Using the My Site feature, you can also create personal sites to share data and documents. Audience-targeting features enable you to personalize content and target it to specific sets of users.

Search

The document and content stored in Office SharePoint Server can be indexed and subsequently searched. Searching is available from within any site. It can also be incorporated into custom pages and applications. In addition to searching within the content stored in the server, you can connect to external data stored in line-of-business applications and databases. The data is indexed, and you search and view it directly from within Office SharePoint Server. For example, you can search for people, colleagues, and experts in a specific field. The search indexer can also connect to third-party content repositories.

Forms Services

Yet another component of the server is Forms Services, which enables server-side forms authoring and management. Forms Services is the server-side extension of Microsoft Office's InfoPath tool. You can author electronic forms in InfoPath client and publish these forms to the server. The forms can then be filled out through either the InfoPath client or, using Forms Services, in the browser.

Forms can contain different controls (such as data pickers and combo boxes). The forms can be backed by simple input validation or custom code. Forms can display data that comes from external data sources because they push the data entered in the form back to an external data source.

Office SharePoint Server provides a complete forms-management solution, as well, that leverages all the document-management functionality provided.

Business Intelligence

BI functionality is also built into Office SharePoint Server. It supports the creation of BI portals and dashboards in Excel Services by providing browser-based spreadsheets that users can interact with to explore data. In addition, Office SharePoint Server includes Report Center site templates, report libraries, key performance indicator (KPI) lists and parts, Data Connection Libraries (DCLs), and support for building and using dashboards.

Report Centers

Office SharePoint Server includes a template for Report Centers, which are sites focused on disseminating BI information through the portal. As shown in Figure 1-8, the site includes a Report Library (which is a customized document library that is tailored to support reports), a DCL, a report calendar, and a set of default views and pages onto these collections.

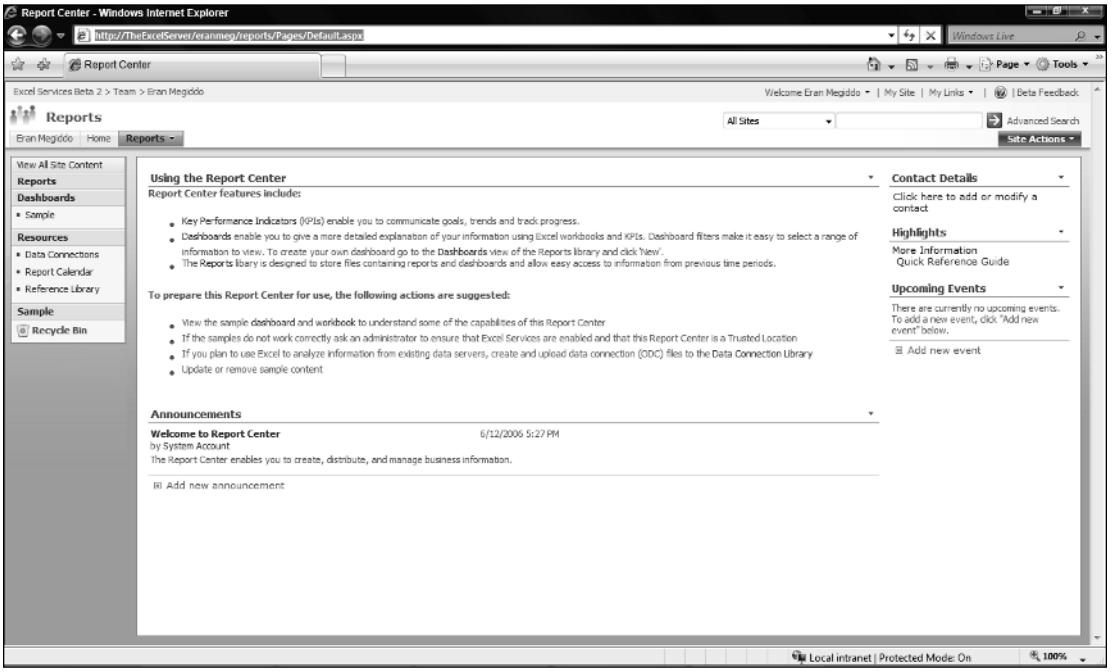

Figure 1-8

Report Libraries

Report Libraries are document libraries tailored to contain reports and dashboards. The primary addition to regular document libraries is the support for retaining report histories for each report in the library. A good example is a monthly inventory report. If you want to keep a copy of your monthly inventory reports (so you can return to any given month and view inventory, for example), you'll need a copy of the report as it was at the end of every month. However, a simple document library would quickly become cluttered if you tried to store all of these reports in it.

Report Libraries enable you to save the history for the report and collapse it behind a single entry point for the inventory report. In the basic views of the Report Library, you see only the latest copy of the inventory report listed. Through the Report Library menu, you can expand and see all the historical copies of this report.

Report Libraries also include a number of default fields and views that are common in reporting scenarios.

Data Connection Libraries

Data Connection Libraries (DCLs) are document libraries extended and tailored to contain data connection definitions. In general, DCLs are geared to hold files that contain connection information for applications such as Excel to connect to external data sources.

Because data sources are stored in a central location on an Office SharePoint Server, end users and administrators can locate and maintain them more easily. For example, if you are in Excel and want to connect to an external data source, you can browse through the list of data connections that have already been defined and saved in a DCL. These connections are represented by descriptive names and text, making them easy to identify. Based on these, you can select the data source that you're interested in and Excel will connect to it. You don't need to know the various technical details of the data connection definition (such as the server name, the database name, and query).

Storing data connection definitions in central DCLs also makes it easier to manage data connections. Because the data definition is stored in one location, you only need to update it in that location.

As an example, consider a data connection file that defines the connection information for a sales database. Many Excel users have authored workbooks that connect to that sales database, all using the data connection information stored in that file. If the administrator of that database needs to move the data to a different machine, he or she only has to update the data connection definition in the one central location stored in the DCL. The next time the Excel workbooks that use this connection are opened, they will automatically update to the new connection information and the new machine name, and the data is refreshed. This saves the administrator from needing to maintain and update every single workbook manually.

Chapter 5 covers data connections and DCLs as they pertain to Excel Services in depth.

Key Performance Indicator Lists and Parts

Key performance indicators (KPIs) provide the basis for tracking key measurements in an organization. In addition to the actual values being tracked, KPIs include goals and trends. Many BI systems enable tracking KPIs on dashboards, as part of lists, in reports, or as part of more complex score carding applications. Office SharePoint Server provides basic functionality for displaying KPIs in dashboards using the KPI web part, or as part of KPI lists, as shown in Figure 1-9.

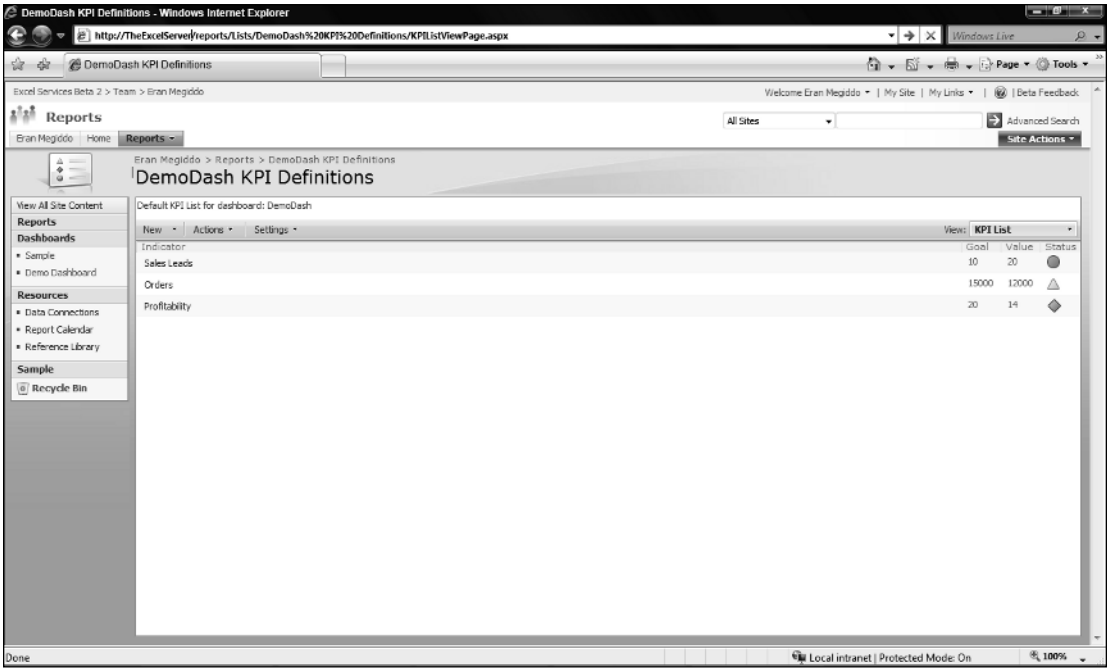

Figure 1-9

KPI lists contain KPIs and can present them visually. The KPIs can be contained in a flat list and be grouped hierarchically. The status, goals, and trends for the KPI can be sourced from values in a SQL Server Analysis Services 2005 cube, in an Excel workbook, or in a SharePoint list, and simply entered into the KPI.

These different KPI types can all exist in the same list, providing a great aggregation and view across multiple data sources.

The KPI web part displays the value and additional information for a single KPI. Users can get to these parts either by clicking on a KPI in the list, or by placing the KPI part directly on a dashboard.

Dashboards

A dashboard is an aggregation of related data and content in the form of reports, tables, and charts. These are presented on the same page, and provide an aggregate, contextualized view. For example,

a regional sales manager can have a dashboard page that displays the latest sales numbers, current customer support escalations, the target sales for the period, and a list of current leads.

Office SharePoint Server supports building such BI dashboards based on the core Web Part Pages capabilities described earlier. In addition, it provides features that target common BI dashboard needs and makes it easy to create such dashboards.

To create a BI dashboard, you use the dashboard page template, which guides you through a wizard-like setup process. For example, you can choose the default parts to include in the dashboard and additional customization options. When you are finished with the setup, you are presented with the dashboard page, and can then continue to assemble it by configuring the content to display in the dashboard.

Another core feature of BI dashboards in Office SharePoint Server is a set of filters that enable you to connect all the content on the dashboard. This is done by filtering the different elements of the dashboard together. For example, the sales dashboard described earlier can include filters for the product line and for the sales region. The sales manager viewing the dashboard can select a region and all the various parts of the dashboard (such as the latest sales numbers or support escalations) will show the data for only that region.

Office SharePoint Server includes several filters out-of-the-box, and more can be added programmatically. The out-of-the-box filters include those that support filtering based on an authored list of values, values that come from a database query, data based filtering, and others. Figure 1-10 shows an example of such a dashboard.

Figure 1-10

Chapter 12 describes how you can build a dashboard with Office SharePoint Server using Excel Services to include workbooks in the dashboard.

Administration

Tying together all the functionality described is a unified setup, deployment, administration, and monitoring system. It starts with one setup that delivers all the functionality described. Managing topologies, administrating server settings, and monitoring server status and health is all unified as well. Chapters 2 and 7 go into far more detail on all these topics as they pertain to Excel Services functionality. For now, it is important to understand that Excel Services administration is an integral part of Office SharePoint Server integration, and not a separate solution. With that said, there are numerous settings that are specific to managing Excel Services functionality, which this book covers in depth as well.

The key to Office SharePoint Server is that all the capabilities are completely integrated into a single product. Though you can use it as a document management system or as a portal, the real strength is building solutions and applications that leverage all this functionality in an integrated way. The same is true for the functionality provided by Excel Services. The needs described previously are best addressed when you leverage Excel Services capabilities with the broad functionality of Office SharePoint Server.

There are many more capabilities and features of Office SharePoint Server that are definitely worth learning about. This book can only provide a brief overview. As the book continues and expands on Excel Services and walks you through setting-up, administrating, and using it, additional Office SharePoint Server concepts are described. Excel Services is the Office SharePoint functionality that supports loading, calculating, and rendering an Excel workbook on the server. The solutions that are detailed in the following section, and described in further detail in Chapters 9 through 13, all focus on Excel Services functionality, but also apply to additional Office SharePoint server features from the sets described.

Now that you have a basic understanding of the core set of needs users have for Excel, what Excel Services is, and how it fits in the bigger Office SharePoint Server product, the next section describes the five key ways that you can use Excel Services.

Five Key Ways to Use Excel Services

This section examines five key uses for Excel Services. These use cases line up with the needs described. They also serve as examples in Chapters 9 through 13, which provide step-by-step guides to using Excel Services for each one of these scenarios.

Distributing Workbooks Through the Browser

With Excel Services, you can calculate a workbook on the server and view the results in the browser. This is the foundation for broad and secure distribution of workbooks. You use the Excel client to author a workbook, or you can create or update the workbook programmatically. Then the consumers of the workbook only need a browser to access, view, and explore it.

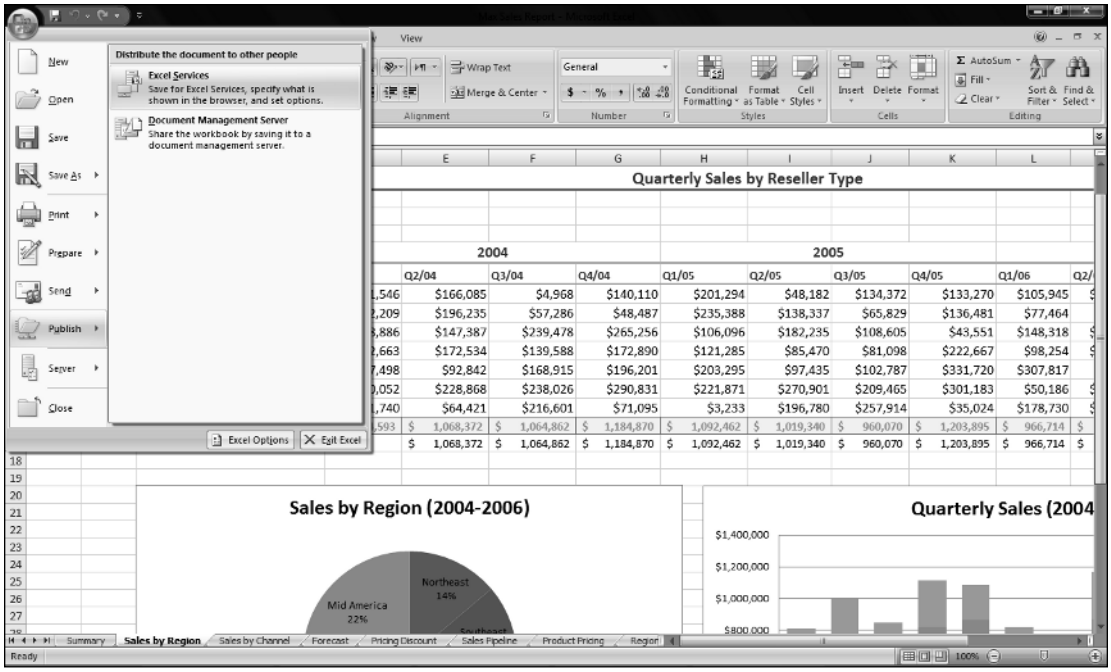

Figure 1-11

The scenario is straightforward. After you author a workbook, you save it to a document library or file share, and provide a link to that location. Figure 1-11 shows a workbook being authored in Excel client.

You can send the link to your workbook to others in an e-mail message, through a notification, or otherwise. Then the recipients just follow the link, and the workbook is calculated by Excel Services and the results are displayed in the browser.

In the browser, users can manipulate the workbook to explore the data in it. For example, they can sort and filter data, drill down or up on a PivotTable, or change input parameters that affect calculation. However, they cannot change the original workbook or values in it. The browser-based rendering looks true to the original, as shown in Figure 1-12.

This sounds simple, at first, but consider how this is different and better than simply sharing the workbook through e-mail or by placing it on a file share and having people load a copy of the workbook in Excel.

When a user opens a workbook in the Excel client from an e-mail attachment or from a file share, a copy of that workbook is downloaded locally. Even if users do not have permission to save over the original workbook, they can save a complete local copy. The next time the user wants to access the workbook, he or she may go back to the copy in the e-mail, or to the copy that was saved locally and, by that time the user is potentially not looking at the correct, updated version of the workbook. The workbook author may have updated it or made changes in the meantime.

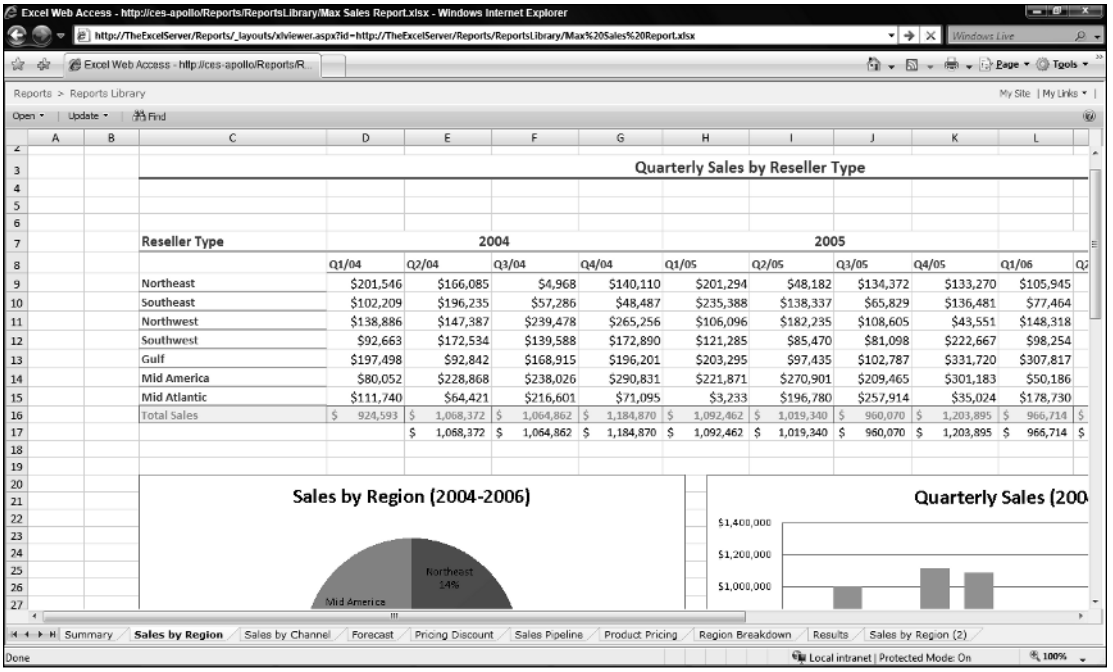

Figure 1-12

When the workbook is shared using Excel Services, only one copy of the workbook exists (where the workbook author saved it). Every time the consumers of the workbook want to view it, they navigate to the link that they were provided. In this way, they are guaranteed to always see the latest version available. In theory, you could also do this by sending an updated attachment or using a simple file share; however, in both those cases the consumers of the workbook have a local copy (either in the e-mail or ones that are saved), so there are no guarantees. Furthermore, for workbooks stored in Office SharePoint Server document libraries and viewed using Excel Services, you can ensure that users do not have permission to download and save a local copy of the workbook. Rather, they will only be allowed to view the workbook through the browser. In this way, it is guaranteed that they are looking at the sanctioned copy.

Accessing workbooks through Excel Services provides additional security. Users viewing workbooks in the browser cannot change the values or formulas in those workbooks. In fact, they cannot see the formulas at all. So, the workbook author is guaranteed that the model developed is protected, and the values in the author's analysis are not changed (either by mistake or on purpose). Authors know that the workbook they distributed for others to view is the one they are seeing. Finally, as you will see in greater detail in Chapter 9, the workbook author can also control which parts of a workbook are made visible to users viewing the workbook through the browser. This provides an additional level of security.

Distributing workbooks through Excel Services and viewing them in the browser meets the need to broadly share workbooks. It ensures that the sanctioned workbooks are being viewed, and that the proprietary data and models in them are secure. There are a couple of additional benefits that are worth calling out.

First, people viewing workbooks in the browser do not need to have Excel installed on their machines. A second benefit is a performance benefit. There are many workbooks that are large. It is not uncommon to see workbooks that are several megabytes in size. When users access these workbooks through e-mail or file shares, they need to download these large files locally. This can be especially slow if the user is working remotely. When you're viewing the workbook in the browser, Excel Services loads the large file, but only the result ranges designated by the workbook author are presented to the user in the browser. Even if the entire workbook is accessible, Excel Services lets the user page through sheets sections at a time. In this way, only a few hundred kilobytes of data need to be transferred across to the local machine, potentially saving significant time.

This is different from simply saving the workbook as HTML and providing a link to generated web page (a feature that exists in Excel today). When a workbook is saved as HTML, it is transformed into what can be considered a snapshot or frozen picture of the workbook. The HTML page shows the values as they were when the workbook author saved it as HTML. Users viewing the HTML page do not see any updated values from external data sources, or volatile functions (such as time-based functions). They also cannot interact with the workbook in any way to change parameters, drill down, or filter values.

On the other hand, when you access a workbook through Excel Services and view it in the browser, it is *live*. When you request to see the workbook through the browser, Excel Services loads and calculates the workbook, which means that any volatile data or functions are updated. The browser-based interface provided by Excel Services also allows users to interact with the workbook and changes are calculated by Excel Services with the results updating in the browser.

The following section describes how you can combine this capability with additional Office SharePoint Server functionality to provide control and management for workbook distribution.

Controlling and Managing Workbook Distribution

Controlling and managing the distribution of workbooks is critical, especially when the workbooks have key business impact. As discussed, this need is even more important given the various regulations and compliance requirements being set. Managing and controlling workbooks with Office SharePoint Server takes advantage of the core capability to calculate a workbook on the server, and display the results in the browser.

Consider a simple example. A workbook author must distribute monthly sales results broadly and must ensure that the results people are viewing are always the latest quarter results so that no confusion occurs. The author also must verify that the people viewing the report only see the results once they have been approved by the division's financial analyst. Finally, each quarterly report must be backed up and retained for future reference.

Office SharePoint Server and Excel Services can provide a solution to this scenario. To begin with, a Report Library is created that will contain all the quarterly reports. The library is configured for content approval and an approval workflow that is triggered every time a major version is checked in. In addition, a retention policy is set up that creates a backup copy of each major version of the workbook in an official file repository.

Only the workbook author and the analyst that reviews it are set up to have permission to view and edit the workbook in the Excel client. Everyone else has only the permission to view the workbook in the browser.

With this setup, the workbook author can manage and control the distribution of the workbook and verify that the intended recipients are always viewing the correct version of the report. Figure 1-13 shows how you can access these features directly from within Excel 2007.

After the workbook author saves and checks in a major version of the workbook, the approval workflow is activated. The financial analyst receives a link to the workbook by e-mail and is asked to review it for accuracy.

At this point, the workbook is not accessible to anyone but the workbook author and the analyst. After the analyst reviews the workbook and approves it, everyone else is authorized to view the workbook. But they can do so only in the browser. This ensures that the workbook they are looking at is the approved distribution copy, and that they cannot download and create copies of this workbook. It also protects any proprietary information represented by the model or sections of data in the workbook.

In addition, after the workbook has been approved, a second workflow is triggered. This one creates a copy of the workbook in the official records-management repository. That document library includes various retention settings to ensure that the document is archived according to corporate regulations.

The site administrator can view an audit log that shows who exactly has seen the workbook, which version, and when.

As you can see, integrating Excel Services capabilities with the rest of the Office SharePoint Server features provides a great solution for managing and controlling workbook distribution.

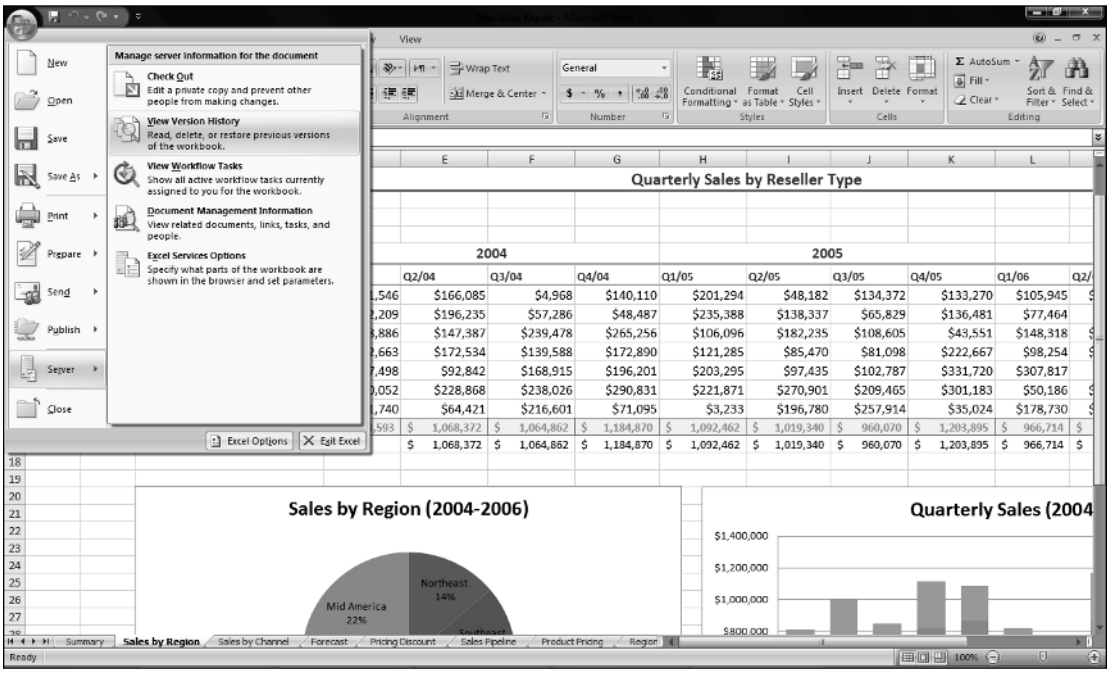

Figure 1-13

Building Business Intelligence Portals and Dashboards

In order for Excel to be considered a complete BI solution, there must be a way to incorporate workbooks that contain data analysis inside of portals and dashboards.

Excel Services and the BI features of Office SharePoint Server provide an answer to this need. With these features, you can use Excel to analyze your business data, and then distribute your analysis through Office SharePoint Server. You can use the BI functionality to build Report Centers and Report Libraries that become the hub for BI and reporting needs.

The Report Libraries can contain Excel workbooks that have been published by the various analysts. Users can then view these workbooks in the browser, ensuring that they remain managed and controlled. They can also incorporate the workbooks (or parts of the workbooks) as part of dashboards built using Office SharePoint Server. The dashboards can contain live workbooks, as well as KPI lists created from Excel.

Leveraging Workbook Calculation in Custom Solutions

As mentioned, Excel Services includes a web services interface. The interface is basic and focused on a couple of core scenarios.

One of these scenarios is building applications that directly leverage Excel workbooks as their business logic. The key capabilities of the Application Programming Interface (API) are loading a workbook, calculating it, setting values into cells, and retrieving values from individual cells or the entire updated workbook.

With these core interfaces, you could conceivably use Excel workbooks as functions in custom-developed applications. The workbooks become an integral part of the application, and are called programmatically to calculate results for specific inputs.

The Excel workbook is calculated by Excel Services, which provides a robust and scalable platform for calculating workbooks. This is very different from calling a client productivity tool, and can be leveraged as part of large enterprise applications.

To better understand how Excel Services can help solve your needs, consider this use case. Assume a pricing analyst in the Marketing Department is responsible for developing pricing and discounts for a specific range of products. The analysts must develop the discount schedule based on various factors (such as the amount being purchased and the type of reseller). The discount model is first developed in the Excel client.

The same discount model must be called by the company's web-based order quotation system. Without Excel Services, a developer would be tasked with recoding the logic represented in the Excel workbook in code so that it can be leveraged as part of the application.

With Excel Services, the workbook author can continue to maintain the workbook, and the developer can leverage the business logic stored in it programmatically. After the details for a new order are entered in the online quotation tool, the program calls on Excel Services to load the pricing workbook, plug the values from the input form, calculate the workbook, and retrieve the resulting values to present the user

with the discount that is offered. Using this method, the developer does not need to track every change in the workbook and update the code. On the flip side, the analyst can update the discount model in Excel with immediate effect on the running application.

As discussed, Excel Services is designed to be a server from the ground up. Thus, the calling application can rely on the server being up and workbooks being calculated, even if many requests are entered simultaneously by different sales representatives asking for quotes.

Offloading Workbook Calculation to Excel Services

A core capability of Excel Services is loading and calculating a workbook on a server. In this first release, Excel Services does not do much to improve the performance of calculating a single workbook when compared with the client. But, because the workbook is calculated on the server, solutions can be built that free up the client computers and by doing that, allow users to continue to work in parallel with the workbook being calculated.

There is no direct out-of-the-box solution for this use case. It relies on the web services interface and a custom solution, such as offloading the workbook calculation through a job-submitting process. You, as a developer, could build a custom interface that enables users to submit a workbook to be calculated by the server directly from within Excel client. Or, you could build a solution that schedules batch calculation of workbooks. Chapter 13 of this book covers the latter example in more detail.

All of these solutions rely on using the web services interface to load and calculate the workbook on the server. The key to these solutions is that the client machines remain free for other activities. Although in general calculation, a workbook on the server does not improve the performance of that specific calculation, there are a number of cases in which it may, such as when all or parts of the workbook are already in the server cache or external data is retrieved. Chapter 3 examines the architecture and discusses the various levels of caching and how they affect performance.

The five use cases described are by no means the complete list of potential uses for Excel Services, although they do represent the key scenarios that Excel Services was designed for in this first version. There are many other ways to leverage server-side spreadsheet calculation and rendering. When these capabilities are incorporated with the rest of Office SharePoint Server's functionality, the uses and solutions that can be built increase even more.

Who Is Excel Services For?

Excel Services functionality and the Office SharePoint Server are server products. Server products require some level of IT-supported setup and administration, and they assume multiple users, so this is probably not a product you will find consumers installing at home. The benefits and scenarios of Excel Services are geared at primarily medium and large businesses. This can be as part of a point solution for a department or branch in a larger organization, or a broader solution for an entire enterprise. Excel Services does not target any specific industry or company type. It also does not target specific workbook applications (such as sales and marketing analysis, or budgeting). The use cases are many and varied.

So who is Excel Services for? Anyone who uses Excel and is looking to solve the needs described can benefit from Excel Services. Whether you are looking for an easy way to share workbooks over the web, or a complete solution for controlling and managing workbook distributions, Excel Services can provide

an answer. If you are building a BI portal and dashboards, you can use Excel Services and the rest of the Office SharePoint Server BI capabilities to do just that. If you are a developer who wants to incorporate the business logic defined in workbooks directly, Excel Services and the web services API it provides could be your solution. Whether you are looking to offload calculations to the server or leverage server-side workbook calculations in any other way, Excel Services can provide the solution.

In addition, if you're trying to solve server-side Excel problems, Excel Services is for you. For example, if you are using Office Web Components to try to build BI dashboards authored in Excel, automating saving workbooks as web pages to broadly distribute them, or automating the client application (Excel.exe) on a server or set of servers to schedule or batch-calculate and distribute workbooks, then you will probably find that Excel Services provides a better solution to your needs.

Also, companies that already have or are looking at deploying Microsoft Office SharePoint Server for other uses (for example, as their corporate portal or document management solution) can gain from the leveraging of the capabilities of Excel Services that they are already deploying.

The next section looks at some of the key things Excel Services is not, and does not do in this first version.

What Excel Services Is Not

This chapter introduced Excel Services and the core needs that are addressed by this new functionality. As outlined, the needs for server-side workbook calculations are real and many. In this first release of Excel Services, a number of these needs are met. But there are many more needs and scenarios for server-side workbook calculation and management that are not solved by this version of Excel Services. A few of these are worth calling out to further define what Excel Services is by describing what it is not.

> *Although the version of Excel Services shipped with Office 2007 does not support functionality listed here, this may change in future versions.*

Following are the top five things Excel Services is not:

- Although Excel Services provides full calculation fidelity with the client Excel.exe through shared calculation code, Excel Services is not the Excel client (Excel.EXE) adapted to run on the server. It is a new set of services designed from the ground up to run on servers and support many users, workbooks, and requests.
- Excel Services is not a product, nor is it a standalone server. Excel Services functionality is packaged as part of Microsoft Office SharePoint Server 2007. It cannot be purchased as a standalone product from Microsoft.
- Excel Services is not a solution for web-based spreadsheet authoring. Excel Services can calculate and render an existing workbook. It is assumed that the workbooks are authored in the Excel client, or programmatically.
- Excel Services does not support simultaneous, multiuser workbook authoring (such as budgeting applications). It is primarily designed for read-only access to workbooks in the context of workbook distribution.
- Excel Services does not support every Excel workbook. Certain Excel features are not supported by Excel Services (as detailed in Chapter 4).

This is not a complete list of what Excel Services is not. And this chapter did not address all the potential uses that Excel Services does address. This is a new technology that is part of a new product. As you discover and use Excel Services, you will find many more uses for this new technology than this book can cover, as well as scenarios and needs that are yet unmet.

Summary

Excel client has been around for more than 20 years and is probably the most commonly used business software. In Microsoft Office 2007, Excel is extended to the server. Microsoft Office SharePoint Server and Excel Services solve problems that exist today, and enable a broad new range of scenarios. Whether it is managing workbook distribution, or building BI dashboards that incorporate Excel workbooks, server-side Excel capabilities are new and exciting.

The rest of this book covers setting up and deploying the server, managing it, and using Excel Services for the key scenarios discussed in this chapter. Chapter 2 walks you through setting up and configuring Office SharePoint Server with Excel Services. You will set up an evaluation copy on a single-box deployment, which will provide you with a server to use in the remainder of the examples in the book.