Chapter 10

Intelligent Business Planning and Reporting Using Microsoft Project 2010

In This Chapter

We explore business intelligence options such as Excel Services, PowerPivot, and SQL Reporting Services to help you take some basic steps in moving toward the business intelligence reporting capabilities of Project Server. We also demonstrate some functional steps in creating Pivot Reports using Project Server 2010 data.

What You Will Learn

* A better understanding of the business intelligence options for Dynamic Reporting
* Some helpful steps for Pivot Reporting and Pivot Formatting
* The flexibility of Office Web applications and ways to leverage these for data accessibility
* How to leverage the collaborative Business Portal (SharePoint) to assist in empowering your social reporting and communications network

What Is Dynamic Reporting . . .

. . . and, for that matter, what is business intelligence (BI)? Depending on whom you ask, you might get different answers, and based on their personal context, they may all be right.

For our purposes, we’ll use the next definition of BI:

type="definition"

Business Intelligence

A category of methodologies and technologies for gathering, storing, analyzing, and providing access to data to help enterprise users make business decisions.

Businesses have been doing some version of BI for years, so what’s so special about the BI version? Dynamic Reporting is one of the answers to that question.

In traditional reporting methodologies, you’d make a request to the information technology (IT) department to write a report that tells you something specific, and you’d get that specific report on a regular basis. If that report didn’t slice things quite the way you needed them, you were pretty much out of luck until IT found the time to write you a new report. Another oft-used workaround was to get the details about some subset of data in a report written to Excel, then slice and dice via your own filtering and pivot tables. This alternative is still quite valid, and Excel, PowerPivot, and Excel Services help you leverage that approach. Using this workaround does, however, require a certain level of sophisticated knowledge about data structures and use of Excel.

A solution to this quandary can be found in something called an OLAP (online analytical processing) cube. The idea behind this is that rather than just using a table in Excel, you create a structure that adds more dimensions to the existing data. The base information, typically numeric in form, is called the measures. These measures are categorized and summarized in dimensions. In a traditional table, you’re limited to rows and columns for dimensions. In the OLAP world, you can have many dimensions, and the dimensions themselves can have hierarchies. For example, you could have a time dimension, and within that dimension, you could have a hierarchy of year, quarter, month, and day. At the time the OLAP database is created, a summary process that aggregates the measures for these various dimensions is performed. So, when the summary data are requested from the cube, the processing overhead is minimized, giving the requestor a quick response time.

Once an analytical cube is available, users can view the data in different ways. They can filter for just the subset of data they would like within a dimension, then break it down within the hierarchies, or combine dimensions to group the results in a way that is of most interest. Once users have seen the data in a particular way, they may take an interest in a more specific area of the data or in reorienting or regrouping the way the data are arranged. This selection of data and subsequent reorientation is sometimes called slicing and pivoting. But ultimately what it amounts to is Dynamic Reporting, as the users are in control of asking the questions.

This all sounds pretty good, so why isn’t everyone doing it? Well, there’s a good news/bad news aspect to it. The bad news is that designing and building analytical cubes in OLAP databases calls for a specialized set of skills and a database server that can handle OLAP processing. Decisions have to be made about what dimensions and measures will need to be supported and what hierarchies and summary levels should be supported. The processing overhead to build the cubes can be considerable; in order to give end users a fast response, we have to take the hit for summary-level computations up front. Often, because of all the overhead and processing time cube building can take, it is done only periodically, perhaps weekly. The good news is that Project Server includes support for cubes. The complex design questions and the processes for building the custom Project Server cubes have already been addressed.

Slicing, Dicing, Drill Up/Down, and Pivot

Sounds like something out of a Vegematic commercial, doesn’t it? But in the world of cubes and Dynamic Reporting, these things all have meaning. Let’s take a brief look at each term.

* **Slicing.** A slice is a section of a cube that corresponds to a single value for one or more dimensions. For example, a cube containing resource utilization by work group for years 2005 to 2010 might have the year 2008 sliced out.
* **Dicing.** A dice is slices performed on more than one dimension. For example, after the slice for the year 2008 was performed, a second slice could be taken for a specific work group, resulting in that work group in 2008.
* **Drill up/down.** This is a technique whereby users navigate up or down from data that are more summarized to data that are less summarized, or vice versa. For example, you could drill down from 2008 resource utilization to the quarter-level utilization within 2008, then to the month level, and so on.
* **Pivot** (sometimes also called a rotate operation). The idea is that the data presented are pivoted to show a different dimensional orientation. For example, in our resource utilization cube, we might start by showing resources on the *x*-axis and years on the *y-*axis and then pivot showing years on the *x*-axis and work groups on the *y-*axis.

Project Server 2010 Cubes

The next analysis cubes are provided with Project Server:

* Project Non Timephased data including project and master projects
* Assignment Non Timephased
* Assignment Timephased
* Task Non Timephased
* Resource Non Timephased
* Resource Timephased
* EPM Timesheet: timesheet and Enterprise Project Management (EPM) dimensions (project, task, and resource)
* Timesheet, including specific timesheet dimensions (Timesheet Project, Timesheet Task, and Timesheet Resource)
* MSP\_Project\_SharePoint: virtual cube that combines Project Non Timephased, Issues, Risks, and Deliverables cubes
* MSP\_Project\_Timesheet: virtual cube that combines Assignment Timephased, Resource TimePhased, and EPM Timephased cubes
* MSP\_Portfolio Analyser: virtual cube that combines Assignment Timephased and Resource Timephased cubes (Comparable to the older cubes supporting Portfolio Server, which is now packaged with Project Server 2010)
* Risks: Windows SharePoint Services (WSS) Risk data
* Issues: WSS Issues data
* Deliverables: WSS Deliverables data

The term “timephased” simply means that the cube contains one or more time dimensions (such as time and fiscal time).

These descriptions are fairly generic. In general, the cubes contain most of the appropriate numeric fields as measures but are limited as to dimensions. The Task cube is surprisingly sparse; Assignments has the nitty-gritty details. A detailed description of each cube is beyond the scope of this chapter, but we encourage you to use the reporting tools to investigate further.

These cubes come out of the box with structure oriented around the fields that are native to Project Server. However, it is possible to extend them to include enterprise custom fields. There are some limitations in this area, however, particularly in the area of adding dimensions for text-based fields. Contact your Project Server administrator for more details.

Creating Easy-to-Access Reporting in Project Server/SharePoint BI

SharePoint has always had strength in the area of collaboration, and one of the key factors in collaboration is easy accessibility. Accessibility can be looked at in two ways: how easy it is to get to something and how easy it is for the average user to work with something. SharePoint 2010 and Project Server 2010 provide several ways to create and access BI reporting.

BI Center and PerformancePoint Services

When you work with BI in SharePoint, the obvious place to start is by creating a BI center site via the BI Center Site template. In the case of Project Server 2010, the Project Web App (PWA) already includes a BI Center Site (see Figure 10.1), which is accessible via a link in the Quick Launch Bar. It doesn’t get much easier than that!

Figure 10.1 BI Center [10-01-biCenter.tif]

Source: Advisicon

Once you’ve arrived at the BI Center, you’ll find that it has a front page that introduces you to different aspects of BI, organized into three sections: Monitor Key Performance, Build and Share Reports, and Create Dashboards. Within each of these sections there are links to samples and to learn more about a topic online. In addition, each of the sections contains a link to the PerformancePoint Server (PPS) sample page. (See Figure 10.2.)

Figure 10.2 PPS Page [10-02-PPSPage.tif]

Source: Advisicon

The PPS sample page will help guide you into the world of PerformancePoint, with more links to help you learn about it. But possibly the most significant item on this page is the button that allows you to download and run the Dashboard Designer tool. Should you choose PerformancePoint and Dashboard Designer as a BI tool, this is the front door.

The BI Center that comes with Project Server has a more extended list of sample reports than that available when creating a BI Center via the out-of-the-box SharePoint BI Center template. These sample reports can demonstrate how you might access and display the appropriate information for various Project Report types and provide you a starting point for creating your own enterprise-specific versions of these reports. The samples are Excel-based reports. Click on the Sample Reports link on the Navigation bar on the left to see the list of samples. (See Figure 10.3.) Reports include:

Figure 10.3 BI Center Sample Reports [10-03-biCenterNav.tif]

Source: Advisicon

EPM

* Simple Projects List
* Milestones Due This Month
* Resource Capacity

Share Lists

* Deliverables
* Issues and Risks

Timesheet

* Timesheet Actuals

Portfolio

* Rejected Projects List
* Top Projects
* Workflow Chart
* Workflow Drilldown

In addition, there is a collection of Excel template files for both reporting DB access and OLAP cube access. The template files are essentially empty spreadsheets with predefined data connections/queries. Based on the data returned by those connections, you can customize your own reports. See the Templates link for more details.

PPS provides the key enterprise-level integration services between the SharePoint environment and BI-oriented functionality. Full advantage of this functionality can be taken via the Dashboard Designer. Dashboard Designer allows you to create indicators, key performance indicators (KPIs) at both a specific and overview level, scorecards to hold the KPIs, charts, graphs, strategy maps (Visio diagrams linked to KPIs), and links to reports created in Excel or SQL Server Reporting Services via integration with the Report Builder tool. Once you’ve designed and built the components required for your specific reporting needs, you can mash them all together into a dashboard. Dashboards can contain one or more pages of strategically grouped information, either grouped together on a page or on dedicated pages. Ultimately the goal is to give managers one-stop shopping for all their key reporting needs. All of this functionality is supported behind the scenes with SharePoint Web Part Pages and Web Parts along with connectivity to databases, analytic cubes, and various other data sources. Powerful stuff!

If the BI Center, PPS, and Dashboard Designer seem a bit overwhelming, don’t worry. These tools are really aimed at integrated enterprise-level BI reporting, but you can keep it simpler. Let us consider some alternatives.

SQL Server Reporting Services

As you might guess, SQL Server Reporting Services (SSRS) reporting functionality comes packaged with MS SQL Server. It is used to create “traditional” reporting (i.e., with a specific report layout, going against specific data sources). Those data sources can include both analytical cubes and relational databases, among other items. In the Project Server context, you can access data both from the Analysis Service, (the element of SQL Server that allows for data analytics against databases), [AU: this term hasn’t been used yet; clarify] cubes and the Project Server Reporting database. While it’s possible to access both in the same report, that’s probably not a good idea, as the Reporting database data has near-real-time data and the data in the analysis services cubes is rebuilt only periodically, often only weekly. Mixing the two could lead to inconsistent results.

The Project Server Reporting database is a MS SQL Server relational database that gets updated each time a project is published. While not all data about each project are available, some data is available here that is not available in the cubes, especially for custom fields. The information in the Reporting database is most easily accessed via Views, which resolve custom fields and lookup values.

Core Project Server functionality includes four relational databases: Draft, Published, Archived, and Reporting. Draft is used to store pending changes. Published is updated at the time a project is published for public consumption via the PWA, and includes all project and enterprise details. Archived supports project-level backup and restore as well other recovery components within Project Server. The Reporting database is also updated at publishing time but does not include all project details and is somewhat optimized for reporting queries. In addition to reporting data, it also contains staging data for the cube building process. Microsoft publishes schema documentation only for the Reporting database, as the other four are designed for internal use and could be redesigned without notice.

SSRS reports are created in one of two ways: by a developer using a tool called SQL Server BI Development Studio (BIDS) or by an end user using a tool called Report Builder. Report Builder has a MS Office look and feel. Report definitions can be saved locally or published to the Report Server. Reports published to Report Server can be displayed within SharePoint pages.

Reports are viewed either through the Report Builder tool or via a browser for reports published to the Report Server. Often the browser runs in a SharePoint context, but that is not strictly required. The report viewer allows you to preview and print one or more pages. Reports can include parameters to allow filtering at run time. Report presentation can be tabular, include graphs and charts, can be expanded and collapsed, and can include drill through to other reports. Reports also can be exported to PDF, Excel, and other formats, including formats that can be read as data sources to Excel 2010.

Similar to using the Dashboard Designer with the PerformancePoint Server, writing SSRS reports via BI Development Studio or Report Builder calls for some training and ideally some knowledge of database constructs. There are things that IT or an experienced user can create to make more complex queries accessible to newer users and wizards to help simplify or initiate basic report creation. SSRS Report design and creation is a significant topic in its own right and is outside the scope of this book.

With the advent of SQL Server 2012, SSRS includes a new tool called “Power View” (code-named Crescent). Power View is a Silverlight-based tool that runs from your browser, which enables easy creation of integrated dashboard-level pages containing pivot tables, charts, and filters. It uses a new SQL Server Analysis Services (SSAS) tabular data model as a data source. Another nice feature is that the SSAS can also use PowerPivot files as a data source, which is based on the same model, allowing exported reports to be also part of additional report building[AU: what does this last phrase refer to?]. While the creation of the data models still may require specialized knowledge, once the data model connections are published to SharePoint, Power View will make creating complex graphical reports fairly straightforward for the average data analyst or information worker and doing ad hoc queries easy for the typical information consumer.

Excel and Excel PowerPivot

You’ve heard of a movie with a cast of thousands? Excel is a tool with a cast of millions. Even back in 2009, Microsoft estimated a worldwide number of users at more than 400 million. Some estimates suggest that 50 to 80 percent of enterprises still rely on stand-alone spreadsheets for critical applications like financial reporting or forecasting. Even in shops where BI is available, that last bit of analysis often is done by a chief financial officer with a spreadsheet. Needless to say, ignoring Excel as a reporting tool would be foolish.

You can get instant benefits from publishing some of your existing Excel spreadsheets to SharePoint, gaining the use of Excel Services. Benefits include:

* Reducing the number of copies in circulation as well as the potential variations on those copies introduced by each copy owner.
* The ability to use SharePoint’s document management services.
* The ability to link the workbook to other SharePoint pages, including pages used to summarize BI information.
* Limiting what parts of your workbook are visible.
* The fact that users viewing published workbook components cannot change content. However, cells can be specified as parameters that viewers can enter values into. Excel formulas can key off those parameters to drive modification of other cells. This modification is nondestructive.
* Using published spreadsheets as data sources for other spreadsheets or reports.
* Server-level functionality that allows for enterprise-level efficiencies (i.e., rather than having a server open 20 different copies of the same big workbook for 20 different users, it knows to open one copy and share it).

PowerPivot

So what’s this PowerPivot thing? As if people were not already making Excel jump through enough hoops with the data that currently can be stored in worksheets, PowerPivot is an add-on for Excel 2010 that allows a user to import data from various sources into “tables” within Excel. These sources include relational databases, OLAP cube databases, other Excel files, text files, cloud services, and feeds from reporting service reports.

Once imported, the disparate tables can be linked together via a relationship editor. One warning about how PowerPivot handles relationships: In the early release (and still true at the time this is written) many-to-many relationships are not supported. For example, if you have a task table and a resource table (a task can have many resources and a resource can have many tasks), even if you have an intersection table (Assignments), the aggregation results of something like task work by resource may not be what you expect. There is a rather technical workaround involving the use of the DAX query language, but for average mortals this is still an issue. Presumably Microsoft will address this in the next release.

The data are compressed and stored separately from the standard Excel data, although they still reside in the same Excel file. A special VertiPaq Analysis Services engine is used to process these data. Once in place, the data is displayed via Excel tables, pivot tables, pivot charts, filters, slicers, and others. In short, PowerPivot provides a do-it-yourself data warehouse with limited Analysis Services functionality, accessible via all the user interface functionality you’ve come to expect from Excel.

This begs the question: Why, if I’m a Project Server user and I already have all of these whiz-bang Project Server cubes already built for me, should I care about PowerPivot? The answer comes down to integration. Unless you already have a business process in place to define and update enterprise custom fields from other related business systems within your enterprise, you are likely to need an alternative for combining that data. PowerPivot provides an option there, and does so in a way that does not depend on the time and expertise of a team of IT specialists. As extending custom fields in MSP cubes does have some limitations and a data conversion may be needed to facilitate meeting these limitations, PowerPivot may provide a temporary workaround. You may not get the power of a full IT OLAP cube solution, but in these days of limited resources, it can get you started down the road to integration.

It’s worth noting that the Vertipaq engine is the same engine used by SSAS in its “tabular” mode. In fact, when PowerPivot files are published to SharePoint, Excel Services coordinates with SSAS and a server-level PowerPivot service to transfer the Excel “database” into SSAS for processing. This enables enterprise-level efficiencies, as the same data can be shared by multiple users.

The Vertipaq engine takes advantage of two key concepts: (1) a more advanced data compression technology that allows a larger number of records to be stored in the same space and (2) including the entire data model in memory to speed up queries. This has pluses and minuses. On the plus side, because we’re working at memory-based speeds rather than doing a lot of disk input/output, we can sacrifice some of the complexities that come with optimizing analytical cubes, which puts the data modeling more in reach of nonspecialists. On the minus side, you will need a lot of memory for bigger data models, and for really large models, you need to revert back to traditional OLAP. As such, PowerPivot is considered self-service BI, and if the data or data usage grows to a certain level, you may have to upgrade to a more traditional OLAP solution.

How to Create and Modify an Excel Pivot Report from a Project Server Cube

In this section, we demonstrate how to create a report that includes a pivot table and associated pivot chart, using out-of-the-box Excel 2010 and an out-of-the-box Project Server cube. Most of the same concepts can be applied to any SSAS cube or relational database.

1. Open Excel 2010.

2. Click on a cell in the new spreadsheet.

3. On the Insert tab, Tables group, click on Pivot Table. (See Figure 10.4.) If you’d like a chart as well, click the down arrow under the Pivot Table icon and select Pivot Chart.

Figure 10.4 Insert Pivot Table [10-04-insertPivotTable.tif]

4. A Create PivotTable with PivotChart dialog will come up. (See Figure 10.5.)

Figure 10.5 Create Pivot Dialog [10-05-createPivotDialog.tif]

5. Select the Use an external data source option and click the Choose Connection button. An Existing Connections dialog will open. (See Figure 10.6.) You can see existing connections in the workbook, on your computer, and on the network.

Figure 10.6 Existing Connections [10-06-existingConnections.tif]

Ideally you should use an .odc connection off the network so you can share your spreadsheet more easily later. If the report is going to be used in a BI Center context, data connections should be stored in the associated data connections library for that site. You may need to “Browse for More…” to find it.

a. If you cannot find a connection for the data source you need, you’ll need to create an .odc file first or have someone create one for you. See the section titled “How to Create an .odc” later in the chapter.

b. For our example, we are using a connection to the Assignment Timephased cube in our Project Server Analysis Services Database server. (See Figure 10.7)

Figure 10.7 OLAP Data Connections [10-07-olapDataConnections.tif]

c. Note that the BI Center site that comes with a Project Server PWA installation includes a number of predefined connections that correspond to the sample reports, which access the Project Server Reporting database. If you opt to customize a sample report, your best bet is to create copies of both the sample report and the connection, as the samples potentially could be affected by future upgrades or service packs. Likewise a number of connections to the Analysis cubes have been predefined, with references from Template reports. These will be in a folder in the connections library that is likely named based on the server and OLAP database that contains the cubes.

6. Once you’ve chosen a connection and pressed Ok on the Create PivotTable with PivotChart dialog, you should get a pivot table and a chart. A PivotTable field list should be available on the right. (See Figure 10.8.)

Figure 10.8 Pivot Table Setup [10-08-pivot0.tif]

7. Scroll down to and click on the check boxes next to the Capacity and Work field measures. These are the measures that we will be summarizing. By default, these will be placed in the Values box in the lower right. The pivot table (and chart) will already begin to fill in, but as we haven’t selected any dimensions, all the capacity and work data are summarized into one total each.

type="note"

Note

Including dollars and hours in the same table may not give the best results in a related chart, as the scale for one may not be appropriate for the other. Pivot tables work well for showing multiple columns of summarized data, but if you’re going to do a chart, you should limit the number of values presented, both to avoid mixing apples and oranges and to keep the chart simple and to the point.

8. Scroll to the bottom of the field list and check next to the Time Dimension. By default, Excel placed this dimension in columns across the top, with subcolumns for capacity and work within each year and a grand total on the far right. Note that there are now two items in the Legend Fields box (also called series, or columns if the table is selected, or you might consider it the *x-*axis). The legends values will also provide color-coded legend references on some types of charts. The fields are Time (the dimension placeholder) and Values (the summarized values)(see Figures 10.9. and 10.10).

Figure 10.9 Drag Area if PivotTable Cell Selected [10-09-dragRowCol.tif]

Figure 10.10 Drag Area if PivotChart Selected [10-10-dragLegendAxis.tif]

Figure 10.11 Excel Pivot Chart/Report [10-11-pivot1.tif]

9. Let’s try some dynamic reporting!

a. A minor tweak: In the Legend Fields box, drag the Time field underneath the Values field. See what happens to the column headers (see Figure 10.11).

b. Drag the Values field from the Legend Fields box into the Axis Fields box. This takes all those columns that used to contain Capacity and Work and turns them into rows. However, if you actually look at the summary values in the cell for say, 2010 Capacity, via both views, you’ll find they’re the same (see Figure 10.12).

Figure 10.12 Excel Pivot Fields [10-12-pivot2.tif]

c. Undo your changes so Time and Values are in Legend Fields, then drag Time into the Axis fields. Now we see rows for each year and columns for Capacity and Work. If we have a chart, we’d notice the chart has also pivoted (see Figure 10.13).

Figure 10.13 Excel Pivot Updates and Filtering [10-13-pivot3.tif]

d. You’ll notice a + sign next to each year. This means that this dimension has a hierarchy that you can drill into (see Figure 10.14). Drill down as far as you can go. You should find breakouts for Year, Quarter, Month, Week, and Day. Drilling down also affects what is displayed on the chart. In Excel, the table (grid) and the chart are tied together, and you control the chart drill down via the table. In other BI tools, such as a Performance Point Chart, the chart can stand alone and includes drilldown functionality. Click on the – sign at the top of the hierarchy you’ve expanded to collapse it again.

Figure 10.14 Pivot Drill-Down Row Fields [10-14-pivot4.tif]

e. Add two more dimensions: Resource Is Generic and Resource List. These should land in the Axis Fields (rows or categories or the *y*-axis). You should now have groupings by Year, Resource is Generic flag, and Resource Name. Under the Resource is Generic if grouping = true, Resource Names should be generic assignment names; if false, Resource Names should be specific employees. Capacity and Work summarized values should go all the way down to per resource[AU: any caps needed here?] per year, and Resource is Generic flag per year. There should not be any resources in both the True and False groups.

f. Remove the Generic Flag group. You can do this by either unchecking it in the field picker or by right clicking on the item in the drag area and selecting Delete Field. Note that generic and nongenerics are now mixed together, and we no longer have Generic/Non Generic totals (see Figure 10.15).

Figure 11.15 Dynamic Pivot Chart Update [10-15-pivot5.tif]

g. With all the resources we have now, our chart is getting too busy. Let’s do some filtering (slice and dice). There are a number of ways we can do this:

i. In the field pick list, if you hover over a Dimension field, the field will highlight and a down arrow will appear on the right. Click the down arrow and a value pick list dialog will appear. Pick your desired values. If you use this option, you’ll also notice filter icons are added to the Row Labels cell and the corresponding “button” in the chart.

ii. Use the flip side. Click on the down arrow in the Row Labels cell, get the value pick list, and pick values as above. In this case you’ll also need to pick the field you’re filtering from a drop-down list, as well as the values.

iii. In either of those cases, the dimension selected has to be selected for display; otherwise, it’ll be ignored.

iv. Using one of the methods just described, filter to include only years 2009 and 2010.

v. Drag a Dimension from the field list into the Report Filter box in the drag area. This method is more useful to filter by a field that you are not displaying. If it is a field you’re displaying, it gets converted from a grouping field to a pure filter field.) In this case, the field is added to a filter area at the top of the Pivot Table. For our example, drag Resource Is Generic into the Report Filter box, click on the down arrow in the new filter area field, and select True.

vi. Back when we were introducing Analysis Services terms, we mentioned “slicing.” You might notice Excel has an “Insert Slicer” function. Is this the same thing?(See Figure 10.16)

Figure 10.16 Power Field Slicer Dimension Fields [10-16-slicer.tif]

Well, sort of. Slicing really comes down to filtering on a dimension, and yes, the “Slicer” helps you do just that. In our earlier filtering options, we mentioned a few methods that resulted in popping up a value pick list for a particular dimension. The problem with that is, once the filter is applied, while an

[10-16-00-filterIcon.tif

icon will show that a filter has been applied, it’s not obvious what the filter is. The slicer provides a more user-friendly interface for a dimension filter, providing a button for each value, with a different button background color showing what the selected filter values are. More significantly, the slicer dialog remains up and available during report usage. It can also indicate which dimension values have no related aggregate values and shift those values to the end of the slicer.

The Insert Slicer button can be found in one of two places, depending on whether you’ve selected a cell in the PivotTable or have selected the PivotChart. (See Figure 10.17.) For the PivotTable, look in the PivotTable Tools group, Options tab, Sort & Filter subgroup. For the PivotChart, look in the PivotChart Tools group, Analyze tab, Data subgroup.

Figure 10.17 Insert Slicer Button [10-17-insertSlicerButton.tif]

Go ahead and insert a slicer and, when prompted for a dimension, select Resource List. The slicer will show the available resource items, with the nongeneric items grayed out as we’re already filtering on generic resources. The other resources should be selected. Deselect all but three by using Ctrl mouse click. (See Figure 10.18.)

Figure 10.18 Using a Slicer [10-18-pivot6.tif]

10. Excel provides many different types of charts. To switch chart types, right click in a chart area and choose Change Chart Type, or go to the Design tab in the Pivot Chart Tools group in the ribbon (you’ll need to have selected in the Pivot Chart area for that tab group to be visible). The PivotChart Ribbon group has Design, Layout, Format, and Analyze tabs, and each tab has various subgroups and options for choosing and configuring charts. Look around and experiment. You may wish to save your work before you start, to remember where you started from. See Excel Help, Available Chart Types for a rundown on the different chart types, when you may want to use a chart type, and what the limitations might be.

11. After you’re done looking around, try a pie chart. One limitation of a pie chart is it only shows the first item in a series, so you’ll only see the numbers for work or capacity, but not both, in the chart. You can control which of the values you see by moving their order in the Values section of the Drag area; the topmost value will be the one displayed in the pie chart. Once the pie chart is shared in SharePoint, your selection is locked in.

In our example case, we opted for a layout that shows the summed amount per grouping, no legend, and used the Layout tab and Data Labels option to show the labels on the outside end of the data points.

If you publish your Excel file to SharePoint for viewing via a browser, the position of charts and slicers in the browser becomes fixed. As your pivot table might expand due to removing filter criteria or expanding groups, you should make allowances when you place your slicers and charts. The illustrations show a good example of where not to place a slicer; it’s there now only to fit into the illustrations.

12. Next, lets format the numbers in our table and chart.

a. In the Drag area, click on the drop-down arrow for Capacity, and select Value Field Settings. (See Figure 10.19.)

Figure 10.19 Value Field Settings Button [10-19-valueFieldSettingsButton.tif]

b. Click on the Number Format button (see Figure 10.20) to bring up the Format Cells dialog, and select currency, zero decimal positions. Press OK on both dialogs to set the format (see Figure 10.21).

Figure 10.20 Value Field Settings [10-20-valueFieldSettings.tif]

Figure 10.21 Format Cells [10-21-formatCells.tif]

c. Likewise, select Work, Number Format, Number, and zero decimal positions. Add a comma to the work field.

d. You can also use the standard Excel formatting controls available on the Home tab. Just remember to select all of the appropriate cells in the table before doing your formatting (see Figure 10.22).

Figure 10.22 Power Pivot Chart Change [10-22-pivot7.tif]

13. When you’re happy with your results, save them. Ultimately you will need to save them to SharePoint if you want to share your report, but you can save locally and upload if you prefer. Saving directly to SharePoint via Excel enables you to control what parts of your workbook are visible to those with whom you are sharing. See the section titled “How to Publish Only Parts of an Excel Report and Specify Parameters” for more details.

Office Web Apps

Depending on your environment, if a user navigates to a SharePoint library and opens an Excel file stored there, SharePoint may automatically open up a full page using the Excel Web App Web part. This may cause you a point of confusion, as Excel Web App is a Web part that is a separate but more fully functional version of Excel Web Access.

Earlier we worked with the Excel Web Access Web part, which is essentially read only. While spreadsheets displayed via that Web part can be manipulated to interact with pivot tables and charts and even to accept parameters into certain cells that can in turn trigger formula processing, the underlying spreadsheet remains unchanged.

Excel Web App is a part of a separately licensed suite of tools called Office Web Apps. Office Web Apps are designed to provide near-Excel client-level functionality for MS Office applications run from a Web browser. The Web part includes a view mode (much like Excel Web Access) and an Edit mode. Edit mode presents the workbooks and spreadsheets in a way that looks much like the client. There are some differences, however.

Included Functionality

* Data entry
* Formulas
* Basic formatting
* Tables, including sorting and filtering
* Interaction with existing pivot tables and pivot charts
* Perhaps the most interesting functionality: Multi-user coauthoring (This means if two users are working on the same document at the same time from different browsers, the changes that one user makes will be reflected on the other user’s copy, in near real time. For those who think that this is not a good thing, that behavior can be prevented by using document control [i.e., checking the document out before editing].)

Not Supported

* Creating charts and pivot tables
* Query tables

Both Web Part versions allow you to open the document in the Client version. That option is configurable for Excel Web Access when you’re defining the Web Part properties, as are a number of other options.

The overall suite of Office Web App tools includes online companions for Word, Excel, PowerPoint, and OneNote. As with Excel, each of these apps may have some differences with the client version. Usage requires an Internet connection and Internet Explorer, Firefox, or Safari browsers.

If You Like Dashboards, Wait Until You See PerformancePoint

Excel is a very useful tool for dynamic reporting and self-service BI, especially if you include the PowerPivot add-on. In combination with Excel Services and SharePoint, you can even extend its reach to the Enterprise. But at its heart, it’s still a spreadsheet engine. SharePoint Enterprise also has some native Web Parts for key performance indicators (KPIs), KPI lists, and charts, but their functionality is rather limited. If you really want to do more sophisticated KPIs or build integrated dashboard pages using a range of reporting techniques, you should look into PerformancePoint Services.

With PerformancePoint, you create your dashboards using a custom Dashboard Designer tool. This is used to create a dashboard with one or more Web pages, each potentially containing one or more dashboard components. Components include KPIs, scorecards to arrange and roll up KPIs, KPI details, wrappers to contain Excel reports, as demonstrated earlier, as well as wrappers for SSRS reports or any external Web page. PerformancePoint also contains special components for charts and pivot tables customized for use with SSAS analytical cubes, which provides functionality beyond the reach of the Excel variety. These analytical charts and pivot tables also provide access to a decomposition tree that allows you to dynamically drill into the details of your cubes while displaying the paths you use to see the various facts and dimensions.

Business Users Can Run What-Ifs

Excel 2010 is chock full of functionality, much of which you may not be aware of. One of the lessor-known functions is what-if analysis available from the Data tab.

Doing what-if analysis does require a little work up front. You’ll need to set the stage by creating a worksheet that contains a base scenario, where you can plug a value into one or more of your cells. Based on formulas or values in other cells, a target cell contains the results.

The What-If Analysis button provides three levels of functionality:

**1. Scenario Manager.** Given a worksheet with a parameter cell, a results cell, and a set of values for the parameter cell, the Scenario Manager shows a summary of comparative results.

2. **Goal** **Seek**. Given a worksheet with a results cell and a targeted results value, modify the parameter value to correspond to the target value.

3. **Data Table.** This is similar to the Scenario Manager, but instead of entering each input as a scenario name and generating a results report, it specifies a data range that contains a column of input values and a companion column to hold the results. A variation on this allows two inputs, where the data range has both columns and a top row to specify the two inputs, with the result values being placed in the remaining matrix.

This analysis is great for those who are doing spreadsheets with formulas and would like a way to investigate scenarios based on those formulas. But what does that have to do with Project Server? With Excel 2010, there’s also an extension for what-if analysis that extends to pivot tables. Using this functionality, you can enter changes into the pivot table cells and have the changes reflected throughout the table. If write back has been enabled on the server, you can even publish your changes.

You’ll need to enable this functionality before you can use it. Start by clicking in the pivot table area, then the Options tab under PivotTables Tools. Next click the What-If Analysis button in the Tools group, and then click the Enable What-If Analysis button.

Collaborative Business Intelligence

With the growth of social media and the continuing acceptance and use of collaboration portals, the natural outcome is to make BI data, reports, views, and information available in these environments.

SharePoint is the principal tool for collaboration and quick access to information, reports, and views, while social networks allow end users, stakeholders, and other interested parties to collaborate, review, and post information regarding projects and project information.

We have heard over and over again that the audiences and participants (both recipients and those actively involved in the projects) want to “tell the story” in a manner that is quick, easy, and Web accessible.

SharePoint Collaboration

The title “SharePoint Collaboration” is almost redundant; sharing and collaboration on business documents and processes is what SharePoint does best. From a BI perspective, we’re using SharePoint and its partner services in these ways:

* To create Excel reports and share them via Excel Services
* For browser-based access to dashboards
* To integrate with PerformancePoint services for the creation and execution of dashboard components
* For storage and security for data connections and for reports generated via BI processes
* To create native SharePoint KPIs, KPI lists, and SharePoint charts
* For document control and alerts (i.e., native SharePoint functionality that can be applied to BI documents)

Social Networks

At first glance, you might think BI and social networks are odd bedfellows; after all, why would you want to share the very information that might give you a competitive advantage? But if you revisit the idea in the context of social networking techniques within a company, or even between business partners, the advantages begin to surface. In fairness, the term “social network” is probably too broad; “social sites” or “social applications” might be a more accurate description, as the scope is limited. In any case, the key advantage comes down to collaboration.

Once the BI reporting has been created, sharing it within your business environment and encouraging feedback and dynamic interaction can help you better respond to and tune that information. BI reporting is sometimes thought to be aimed at the executive level, but if you expand its reach to those better aligned to drill into and understand the details behind potential problems, they may be able to deal proactively with problems before they even get to the executives.

While formal social networking tools such as wikis, in-house blogs, and Facebook accounts can be used as avenues to communicate ideas or the existence of useful information, passive “following” can also be useful. This can be as simple as acquiring access to alerts for a SharePoint library of BI reports, even those that seem only peripheral to one’s area of expertise. These alerts can provide related information and ideas on how to approach certain problems, or even become new data sources to augment BI reporting that you’re already working up. Consider the PowerPivot model: One use is acquiring information from potentially different sources and mashing them together for a broader analysis.

Facilitating the social communication that empowers collaboration may be the key contribution of social networking to BI. The most important aspect simply may be encouraging those who have useful content to publish it to a wider interested audience and to encourage the wider audience to look beyond its little silo for information. This communication should be done with some care. If someone thinks they’re pulling in the latest data but they’re really getting two-year-old data because the poster didn’t specify its currency, the results could be detrimental. As such, some standard for “tags” or other metadata should be included with a posting so the context can be interpreted appropriately.

Extending Reporting

A core component to extending reporting is the ability to open and connect to information from other data sources. This section helps you understand the potential and shows steps for connecting to other databases. It explains how reporting [AU: noun missing here]can be an easily attainable activity that will allow you to extend reporting to information from almost any source.

While connecting to external data sources is easy to do, it is absolutely critical to understand the thought processes of those who are storing or creating the data repository in order to creating good reports. This is where the business needs to work with the database administrator to ensure that information is compiled, collected, and stored in a manner that supports the continued and extended reporting capabilities of Excel, Excel Services, Performance Point, and PowerPivot.

How to Create an .odc Connection File

1. Open Excel 2010.

2. Within the Data tab, click the Get External Data button. An initial pop-up menu will appear, showing From Access, From Web, From Other Sources, and Existing Connections. Click the From Other Sources button. (See Figure 10.23.)

Figure 10.23 Getting External Data [10-23-getExternalData.tif]

As you can see, you have many options for external data sources. Each of them has different characteristics and settings and will have corresponding dialogs to control those settings. Some may involve building queries appropriate to the data source. Exploring them all is beyond the scope of this book. We’ll use From Analysis Services for our example, as that has been the source of data for the rest of our Excel reporting presentation.

3. Enter the name of the server that contains the cubes. Normally Use Windows Authentication is the preferred option, but select and enter a user name and password if that is your company’s preference. Click Next (see Figure 10.24).

Figure 10.24 SQL Connection Wizard [10-24-connectionWiz1.tif]

4. There may be several Analysis Services databases on the server. Choose the appropriate one. A list of cubes associated with the selected database should display. Choose the appropriate cube and click Next (see Figure 10.25).

Figure 10.25 Database Connection [10-25-connectionWiz2.tif]

5. Optionally override the file name and friendly name, and enter a description to help others understand what the connection points to and how it might be used.

type="note"

By default, the new .odc file is stored on your computer under My Data Sources.

If you’re always going run spreadsheets that use this connection from your computer, you can leave that. However, if you want to share those spreadsheets, you should use a data connection file that will need to be stored in a trusted Data Connection Library on SharePoint. Use the Browse… button to navigate to the appropriate Data Connection Library (see Figure 10.26). These connection files can be reused.

Figure 10.26 Save Data Connection [10-26-connectionWiz3.tif]

6. Click Finish.

Note that in this case we’ve named an entire cube. If you had selected SQL Server as a data source, it would have led you down a path to select a table. Frequently, queries to SQL involve multiple tables and should include some filtering on rows and columns to cut down on overhead. Should this be the kind of functionality you need, you might want to look into the Microsoft Query Wizard (available via From Other Sources). It’s not the friendliest wizard, so if that doesn’t work out, you might consider consulting with your local SQL super user or IT applications developer to help you tune up the query or create a view that you can select directly.

If you want a native SQL query rather than a view, you can still have the SQL embedded in the connection file. You’ll need to have figured out the quirks of the Microsoft Query Wizard or used some other method to come up with a workable SQL query. Once you have it, getting it into the .odc file takes a bit of a kludge. You actually have to start with an existing .odc file (preferably one that’s already pointing at the database server using the preferred credentials). You can start down that path by creating the .odc using the method we just described, only using a relational database rather than a cube and picking any table or view to get things started. An .odc created and stored locally is fine at this point.

Once that connection exists, you can access it via the Connections button on the Data tab. This will open the Workbook Connections dialog, where you should click on the Add… button. This will bring up an Existing Connections dialog, where you should select and open the appropriate connection. After it’s opened, you’ll return to the Connections dialog with your connection file listed. (See Figure 10.27.)

Figure 10.27 Use the Workbook Connections Dialog to Add, Delete, Modify, or Refresh Connections [10-27-workbookConnections.tif]

From here, press the Properties… button, which will bring up a Connection Properties dialog. This dialog has two tabs: Usage and Definition.

The Usage tab includes some check boxes of interest: Enable background refresh and Refresh data when opening the file. These are of particular interest in the context of Excel Services, as they help ensure that workbook data are refreshed or can be refreshed when being accessed from SharePoint.

The Definition tab contains the path to the connection file, an Always use connection file check box, Connection string, Command type, Command Text, Authentication Settings, and Export button. (See Figure 10.28.)

Figure 10.28 Use the Connection Properties Dialog to View or Modify Various Aspects of a Connection [10-28-connectionProperties.tif]

In this context, we’re interested in the command type and text. You’ll need to click the drop-down for command type and change “Table” to SQL. In the command text, insert (paste) the SQL that you’ve worked up. Note that if you already have a SQL query, the command type will be locked in and the Edit Query button will become enabled. That button will take you back to our old friend, the Microsoft Query Wizard.

The last item of interest is the Export Connection File button. This is what we’ll use to export the connection file from our local machine to the SharePoint document library that is being used to house the data connections. This needs to be a library designated as “trusted” by Excel Services, and naturally you will need to have SharePoint permissions to save to that library. The button opens a Save File dialog, where you can enter the URL for the site that contains your library. From there the dialog will assist you in navigating to the Data Connection Library, and you can save the connection file for general use.

How to Display an Excel Report from a SharePoint Page

Once you’ve created your Pivoting masterpiece, it’s only fair that you share it with the rest of the world (or at least the rest of your work group). You have four options:

1. E-mail it to all your friends.

2. Drop it into a shared network folder.

3. Save it to a trusted document library in a SharePoint collaboration site.

4. Include it in a Web Part on a page where it adds value and context to other items on the page.

All of these options may be valid, depending on the circumstances. But since we’re talking about BI, it’s really the last option that provides the most value added. Options 2 and 3 are somewhat similar, with the exception that, with option 3, you also can take advantage of document control (check out, check in, version history), Excel Services, and more. Option 3 also sets up your report so it can be referenced in other SharePoint pages, which brings us back to option 4.

Displaying an Excel report on a SharePoint page is really quite simple. Since we are using SharePoint, the obvious solution is to use a Web Part designed for showing Excel, and those clever folks at Microsoft have built just such a thing.

There are three variations for viewing your Excel report from SharePoint:

1. Navigate to the document library that contains the Excel report and open it directly.

2. Include the report in a page or section of a page within a PerformancePoint dashboard.

3. Create a Web Part page in a site or workspace, and set up the Excel Web Access part to display your Excel report. We will walk through this option.

Steps for Displaying Excel in SharePoint

1. If your spreadsheet contains a Data Connection (as our How To example does), you need to ensure that the Data Connection exists in a Data Connection Library on SharePoint and that your file is pointing at that connection. That Connection Library also has to be designated as “trusted” by Excel Services, which calls for intervention by a SharePoint administrator. In general, your organization should have a set of data connections predefined in a Data Connection Library for targets of interest, with the appropriate security context being used for the connections. Since the Web Part routes its requests through Excel Services, security has to be considered in that context.

2. Your Excel file also needs to have been saved to a SharePoint document library that has been configured as “trusted” by Excel Services. Contact your IT department if you do not know which libraries have been configured in this way. Ideally, this library should be in the same site as your Web page, although it is possible to display a file from a trusted library in a different site if you know the URL.

3. If you do not have a page, you can create one by going to the site of interest, clicking the Site Actions button in the upper left corner, and choosing the New Page button.

4. Go to the SharePoint page you like to include the report in, select Edit, and open the Insert tab in the Editing Tools group on the ribbon. (See Figure 10.29.)

Figure 10.29 Editing SharePoint Page [10-29-editPage.tif]

5. Click on Web Part (see Figure 10.30). In the Categories list, select Business Data. In the Web Part list, find and select Excel Web Access (see Figure 10.31), and click the Add button.

Figure 10.30 Inserting a Web Part [10-30-insertWebPart.tif]

Figure 10.31 SharePoint Web Part Selection [10-31-insertExcelWebAccess.tif]

6. Click on the link: Click here to open the tool pane. (See Figure 10.32) The Excel Web Access properties should pop up.

Figure 10.32 Linking to Excel Workbook [10-32-selectAWorkbook.tif]

Figure 10.33 File Workbook Locator [10-33-workbookButton.tif]

7. In the Workbook Display section, enter the location of your Excel file. Click on the … button (see Figure 10.33). This will bring up a Select an Asset dialog to help you navigate to the library where you’ve saved your spreadsheet. Navigate, select your file, and hit OK.

8. Leave the “Named Item” blank. Note that you have the option of restricting what portion of a workbook gets shown in the Web Part via this mechanism. This can be useful for security purposes. You could use this to restrict the displayed item to a pivot table, a chart, or a named range. You can also control what parts of a workbook can be displayed at the time you publish the workbook to SharePoint from Excel.

9. Review the options for toolbar and title bar. Note that if you want to modify the title, you will need to uncheck the Autogenerate Web Part Title check box here, but the actual title is set in the Appearance section.

10. Click on the + for Appearance, and change the height to 600 pixels.

11. Keep all the remaining default options for the remainder of the Web Part properties, but take a moment to look them over. Click the OK button when you’re finished.

Barring data connectivity issues, you should now have a working Web Part that will dynamically refresh when you bring it up and, in our sample case, allow you to filter, drill down, and the like.

How to Publish Only Parts of an Excel Report and Specify Parameters

One of the advantages of publishing an Excel workbook to SharePoint is you gain the ability to limit what parts of the workbook are visible. This mechanism can help protect sensitive information. For example, you might have a worksheet that contains raw data that are used to feed a pivot table on another worksheet. Using the publish options, you can limit what worksheets are seen or even which pivot table on a worksheet can be seen.

A second consideration is that when the Excel report is viewed via the browser, it cannot be modified directly. However, you can configure certain cells to act as parameters that viewers can enter data into. Excel formulas or filters that point to those parameter cells can trigger changes to the content of the remaining worksheet cells. These changes are nondestructive (i.e., the base workbook file is unaffected).

To take advantage of this functionality:

1. From the File tab, use the Save & Send option.

2. Choose Save to SharePoint. (See Figure 10.34.)

Figure 10.34 Save to SharePoint [10-34-saveToSharePoint.tif]

3. In the upper right corner, press the Publish Options button. This will bring up a Publish Options dialog.

4. In the Show tab, pick the level to share at (Entire Workbook, Sheets, Items in the Workbook). (See Figure 10.35)

Figure 10.35 Selecting Specific Worksheets [10-35-pubOption1.tif]

Figure 10.36 Publish Specific Objects [10-36-pubOption2.tif]

Items would include pivot tables, charts, and named ranges (see Figure 10.36). If you opt for Items in Workbook, they end up being displayed individually in the browser, where a View drop-down box allows you to look at the specific item.

type="note"

The View drop-down is a configurable item for the Web Part.

5. To specify a cell as a parameter value, you must first have replaced its default cell name (such as A1) with a named range name. To add the parameter, select the Parameters tab, click the Add… button, and select the named range name (or names) of interest (see Figure 10.37). These cells can be accessed via a Parameters pane in the Excel Web Part (if that is enabled) or used in a dashboard to connect Web Parts.

10.37 Selecting Sheet Parameters [10-37-parameters.tif]

Some additional considerations:

* A named cell can be a single cell or a merged cell. There is one cell for each parameter; a parameter cannot represent a range that contains two or more cells.
* The named cell must be a cell in the workbook. The cell cannot be located in a table or in a pivot table, and you cannot use an external cell reference.
* A named cell can contain a report filter field from a pivot table report. However, the report filter field will not appear in the Parameters pane. To change the range filter field, you can use a filter Web Part.
* The named cell must not reference another named or absolute cell.
* The cell cannot have data validation defined, cannot be locked, and cannot be on a protected worksheet.

6. Press the Save As button, navigate to the SharePoint document library that is to contain your published report, and press Save.

Printing

Printing from the Excel Web Access Web Part is a bit problematic. Your best bet may be to do any filtering, drill up/down, and so on, until you get the view you like, then use the File button to download a snapshot to your Excel client. From that point, you’ll have the full power of Excel client printing rather than being limited to browser-based printing.

Mobile Applications

Office Web Apps can extend MS Office tool functionality into the mobile world. Some of the mobile functionality requires Office Mobile 2010, which is sold separately. For SharePoint, SharePoint Workspace Mobile (part of Office Mobile) can be accessed by a Windows Phone 7.

SharePoint Workspace Mobile helps ensure that the latest versions of your documents stored on a SharePoint site are also available on your phone. When you’re connected to the Internet, you can:

* View content hosted on a SharePoint 2010 site.
* Open and edit Word, Excel, PowerPoint, and OneNote files hosted on a SharePoint 2010 site.
* Browse SharePoint 2010 sites, lists, and document libraries.
* Obtain secure remote access to corporate resources through a Forefront Unified Access Gateway (UAG), if your company uses one.

You can also use SharePoint Workspace Mobile to take your SharePoint 2010 files offline and put them on your phone. You can then open and edit the documents and save them back to the SharePoint site when you’re back online.

Important Concepts Covered in This Chapter

In this chapter, we reviewed the reporting capabilities that come with a PPM 2010 solution and got a taste of the power of BI, including a walk-through and exploration of these features:

* PowerPivot and using the slicer for reporting
* Extending reporting to other data sources, and mapping and modeling in Excel
* Displaying reports in different formats in SharePoint to extend user accessibility
* Localizing what you present (limiting views and parts of Excel reports in SharePoint)
* Overall understanding of the BI capabilities that come with Project Server and SharePoint together