Pro SQL Server 2008 Analytics: Delivering Sales and Marketing Dashboards

Copyright © 2009 by Brian Paulen and Jeff Finken

The source code for this book is available to readers at http://www.apress.com. You will need to answer questions pertaining to this book in order to successfully download the code.

An Overview of Analytics

Most organizations' existing business applications deliver the capacity to store a wealth of valuable data. The challenge is that this data loses its value when it is not readily available to information workers and business executives in the right format on demand. Far too often, sales managers don't have visibility into their pipeline, and they can't make clear decisions about discounting, hiring, and resource allocation. Equally challenging are the situations when finance professionals can't effectively break down revenue and costs by product line and geography. Yet, with client after client, we encounter situations where significant time and effort have been spent designing and deploying business applications without putting any investment into the reporting. The result is that executives have little visibility into how their business is performing, and information workers and IT professionals spend exorbitant amounts of time manually pulling reports that become outdated almost as soon as they are delivered.

Understanding Analytics

A practical definition of "analytics" would be to think of them as the ways an organization arrives at its decisions. Long-time organization employees may base decisions on historical experience or gut feelings, while those more focused on data analytics or financials will focus on the reports or information generated by frequently used applications.

Throughout this book, we will utilize the terms "business intelligence" (BI) and "analytics" interchangeably. At the most fundamental level, we will be discussing the ability to leverage the available Microsoft tools and data from within an organization to improve the decisions that are made by people or groups of employees. Information can be accessed via a wide range of tools in BI environments. On the simple end, for operational or ad hoc reporting, Microsoft Office products (like Excel) can be combined with reports available within line of business applications to gather enough data to make more informed decisions. Should more complex reporting (often referred to as performance management) be necessary, products like SQL Server Reporting Services, Microsoft SharePoint Server, and even Vista gadgets can be used to provide varying views of information. Finally, analytics, or BI if you prefer, includes the ability to present information to consumers where it makes the most sense for them, whether that is via a desktop PC or a mobile device.

Projects involving analytics frequently include data from a wide range of sources. In addition to the standard sources like customer relationship management (CRM) or enterprise resource planning (ERP) applications, initiatives support the inclusion of data from an organization's web site, disparate spreadsheets, and single-purpose applications like dispatching tools or e-mail marketing technologies.

While some of an organization's data exists in these formats, much of it exists in an unstructured format. Merrill Lynch estimates that more than 85 percent of all business information exists as unstructured data, commonly appearing in e-mails, notes, web pages, instant message discussions, letters, general documents, and marketing information. With the growth of the Internet, considering unstructured data is important. However, this book will focus primarily on structured data and on providing the most valuable information to end users.

The Value of Analytics

Imagine running a business without the benefits of monthly financial information:

Put yourself in sales representatives' shoes: what if they don't have access to customer satisfaction information before they head into a sales call?

Picture running the same marketing campaign over and over again, without being able to completely understand the results, conversion rate, and overall return on investment.

These scenarios are examples of struggles that many organizations have when they're not able to synthesize data and present it in a manageable fashion.

Analytics are valuable because they can help individuals within an organization make well-informed decisions. Whether evaluating employee performance, judging the historical and predicted success of the business, or identifying the next investment opportunity, without intelligent information people are simply guessing at the correct answer.

Why Analytics Implementations Fail

One significant source of concern is the ultimate failure of business intelligence initiatives once the data has been gathered and presented. While this and other concerns seem obvious, many implementations fail because of the following reasons:

- *Differing priorities*: End users, managers, and executives within an organization frequently have varying priorities when it comes to managing data. Finding a solution that addresses all user needs is critical. This can be accomplished by identifying a solution that delivers the right level of analytics for each role within an organization, specifically:

  - End users want information in a timely fashion that helps them better perform their day-to-day activities. This data must be easy to find, specific to a role, and available whenever and wherever the employee needs it.

- Managers need information that helps them evaluate the performance of their specific business and/or employees. This information must be a summary but contain enough detail for managers to provide feedback or change course as needed.

- Executives want data at a very high level. Information is often presented in a data-dense format and at a high, rolled-up level.

- *Data explosion*: Organizations are capturing increasing amounts of data across a broad array of locations. Many companies now have customer and prospect data spread between business applications located on premises, in the cloud, and on local hard drives in Excel and Access files. One of the great challenges facing analytics initiatives is the need to standardize applications that serve as the system of record for key customer interactions. At the same time, those organizations must architect analytics solutions that are adept at pulling data from local servers and cloud-based web applications while applying effective approaches to scrub and standardize this data to ensure that it performs well in business metrics.

  Failure to appreciate the need to define the system of record for key business data and to architect for a heterogeneous data environment will lead to a failed analytics implementation: users won't waste their time with scorecards and dashboards that report on only a subset of key business data. As more and more data is introduced into an analytics solution, the complexity of developing the solution and making the results consistent rises. Identifying quick wins and a manageable set of data for each phase will greatly improve the likelihood of project success.

- *Trust*: Employee trust in the data available to them and willingness to work with the results are crucial to the ultimate success or failure of an analytics solution. While the direct problem of having to work through reporting issues during a meeting is significant, the permeation of the subsequent distrust throughout the organization should be given equal weight when addressing issues.

- *Application adoption*: Many times, BI initiatives fail because the underlying systems have either poor-quality or incomplete data, which leads to limited usefulness from the analytics tools. Failures of this kind demonstrate the need for BI projects to be concerned with having a focus on delivering tools that help drive adoption of key underlying business systems.

- *Software-driven projects*: Frequently, we're asked to visit a client who is struggling to adopt an analytics solution. Invariably, the client has implemented the latest and greatest business intelligence tool but hasn't focused on deciding which metrics really matter for the business and delivering these metrics in a role-based context. One thing this book will do is to heighten the recognition that successful analytics projects must be driven by the business and enabled by good technology choices. Both the software and the business are critical for success, and neither alone can lead to successful analytics projects.

Environment Preparations

To successfully navigate the exercises and code presented in future chapters, you need to familiarize yourself with SQL Server 2008. You may already have a SQL Server 2008 environment to use throughout this book; if you don't, Microsoft also publishes a trial version of SQL Server 2008 Enterprise Edition for use. Please navigate to www.microsoft.com/sqlserver/2008/en/us/trial-software.aspx to download the software and install it in your environment. Additionally, for many of the scenarios presented in this book, having a platform or an application on which to build reports and dashboards is important. Based on our experiences, we have chosen Dynamics CRM and SharePoint as the tools used in these application-specific exercises. If you're using these tools, you will be able to implement the exercises in your environment. If you're using a different CRM or SharePoint-like application, the same concept can be implemented using your application.

As anyone familiar with analytics knows, good data often makes or breaks reports, so as we move through the exercises in the rest of this chapter, we will focus on setting up the SQL Server environment and associated data to allow for the best exercises and examples in future chapters.

Finally, from time to time, Microsoft releases a 180-day trial Dynamics CRM Virtual PC environment for evaluation by prospects and partners. If you would like to use this application, please navigate to www.microsoft.com/downoads/en/default.aspx and search for Dynamic CRM v4.0 to see if the VPC is currently available.

Exercise 1-1. Restoring the Database

Exercise 1-1 walks you through the steps associated with restoring a database in your SQL Server environment. This data will be used in subsequent chapters to build reports and provide a structure for upcoming exercises.

1. Restore the CRMDB.bak file located on this book's page of the Apress web site. Navigate to Start ➤ All Programs ➤ SQL Server 2008 ➤ SQL Server Management Studio.

2. Create a new database by right-clicking the databases folder and clicking New Database. Type **Contoso_MSCRM** in the "Database name" field, as shown in Figure 1-1.

3. Restore the backup file by right-clicking the database and navigating to Tasks ➤ Restore ➤ Database (see Figure 1-2).
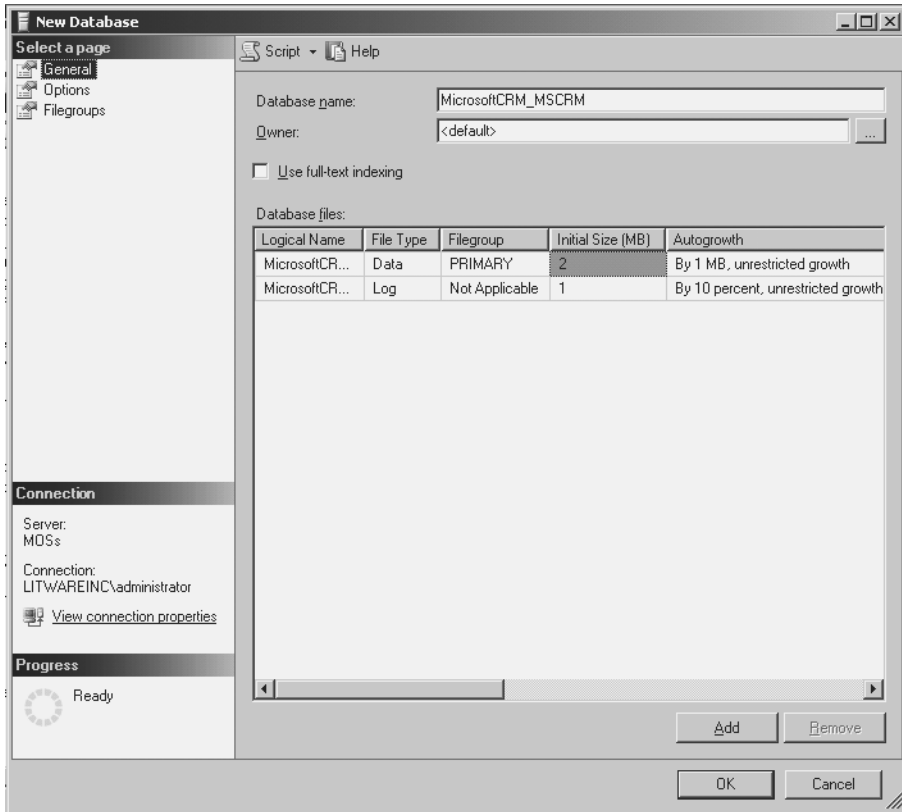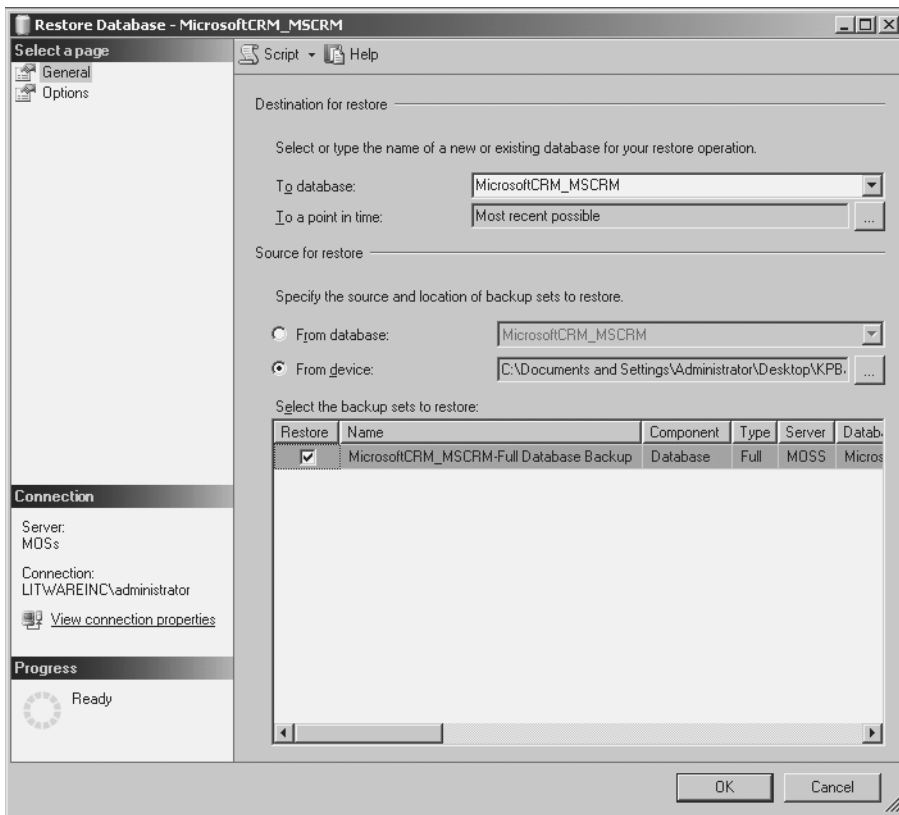
Figure 1-1. *Type the database name.*
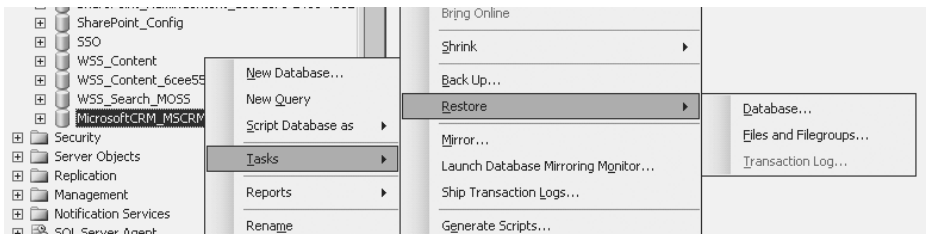
Figure 1-2. *Restore the database.*

4. Select "From device", and navigate to the location of the .bak file (see Figure 1-3).

Figure 1-3. *Backup file location*

5. Navigate to Options. Make sure "Overwrite existing database" is checked. Click OK to begin the restore process. Once the database has restored successfully, close SQL Server Management Studio.

6. The restored database has built-in security that does not allow individuals to query the filtered views without meeting specific security requirements. To ensure that you are able to see data in the views, edit the SystemUserBase table and change the DomainName for AntonK to your own DomainName. For example, if your domain is Adventureworks and your domain name is jsmith, you will input **Adventureworks\JSmith** for the DomainName column of the SystemUserBase table on the row associated with AntonK.

Exercise 1-2 will enable you to familiarize yourself with another important reporting tool, Microsoft Office Excel. This exercise will focus on installing the Data Mining Add-In. We've found that, in deploying analytics solutions, most users are familiar with Excel's pivot tables, but more advanced functionality like data mining remains a mystery. Our goal with this example is to provide a baseline for some of the advanced Excel functionality that we will delve into more deeply in later chapters.

Exercise 1-2. Installing the Data Mining Add-In in Excel

This exercise will walk through installation and use of the Data Mining Add-In in Excel. To use the Data Mining Add-In outside of the demonstration environment, Microsoft Office 2007 and Microsoft .NET Framework 2.0 must be installed.

1. Open Internet Explorer in your environment. Type the URL **http://www.microsoft.com/DOWNLOADS/ details.aspx?familyid=896A493A-2502-4795- 94AE-E00632BA6DE7&displaylang=en**. Download the item displayed in Figure 1-4.
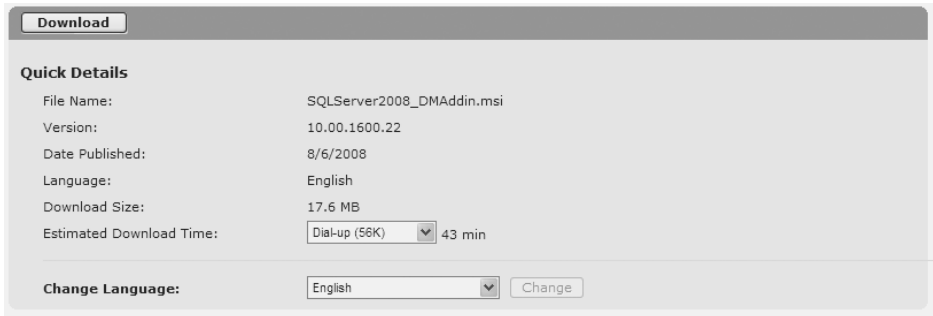

Figure 1-4. *The Data Mining Add-In download*

2. Run the Windows Installer Package you just downloaded. Accept the license agreement, and fill out any remaining questions appropriately. In the Feature Selection page, click the drop-down next to each item displayed in Figure 1-5, and select "This feature, and all subfeatures, will be installed on local hard drive."
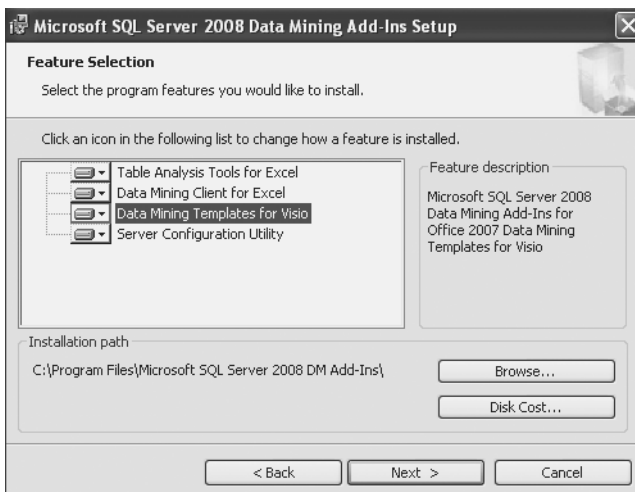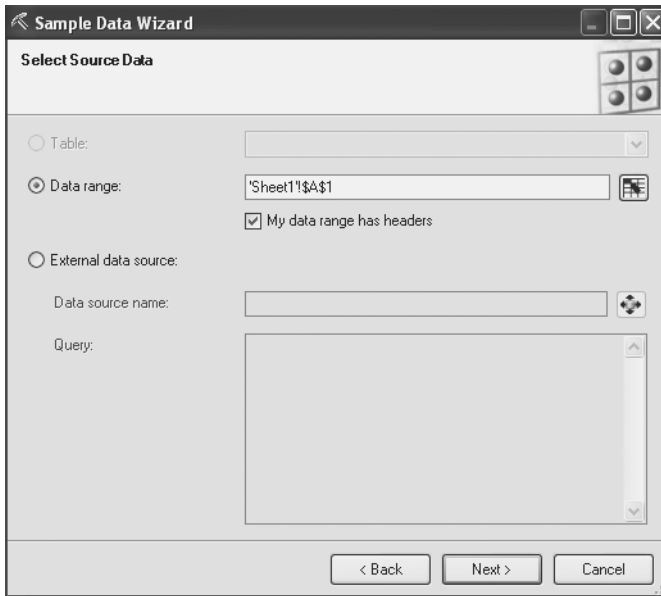

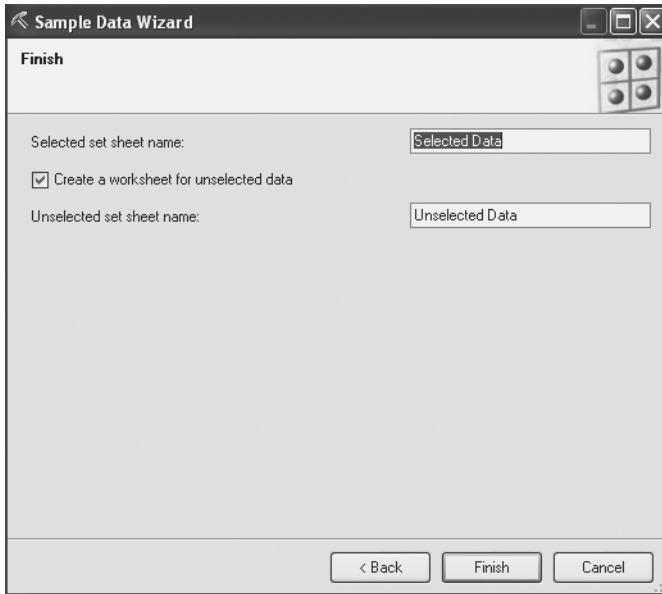**Figure 1-5.** *Data Mining Add-In setup*

3. Configure SQL Server 2008 for the Office 2007 Data Mining Add-ins by navigating to Start ➤ All Programs ➤ Microsoft SQL 2008 Data Mining Add-ins ➤ Server Configuration Utility. Accept all the defaults and click Next on each screens until Finish. Install the Microsoft SQL Server Data Mining Add-In by navigating to Start ➤ All Programs ➤ Microsoft Office ➤ Microsoft Office Excel 2007. Open the file 'DMAddins_ SampleData.xlsx'. Navigate to the Data Mining tab. Click the sample data button to run the Sample Data Wizard. Click Next to get past the introductory page. The screen in Figure 1-6 will be displayed.


Figure 1-6. *Data Mining Add-In setup continued*

4. On the Select Sampling Type page, select the Random Sampling option, and click Next.

5. Select the Table option and a table of your choice, and click Next.

6. The next page is the Finish page shown in Figure 1-7. Accept the defaults on the Finish page, and click Finish.

Figure 1-7. *The final setup step for the Data Mining Add-In*

7. Exit the Configuration Wizard.

8. Exit the Data Mining Add-In introduction page.

9. The sample data will be launched into the selected page. Take some time to look over the information provided.

10. Click the Data Mining tab, and navigate to the No Connection icon shown in Figure 1-8.
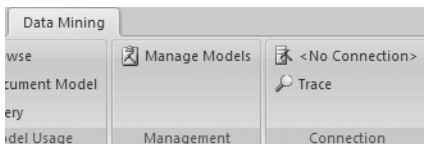
Figure 1-8. *The No Connection icon*

11. Click New, and type **localhost** in the "Server name" field and **DMAddinsDB** in "Catalog name." The friendly name shown in Figure 1-9 will be filled in automatically. Click OK.
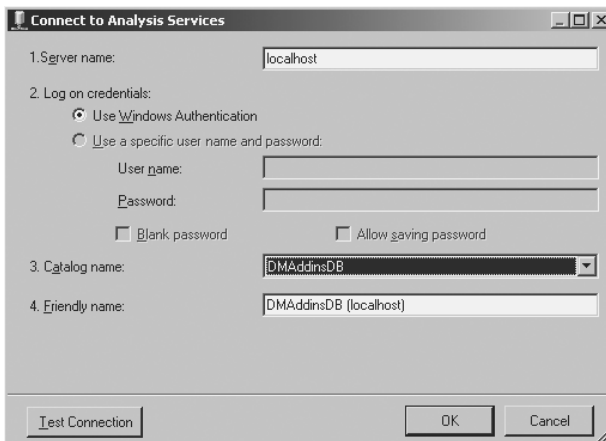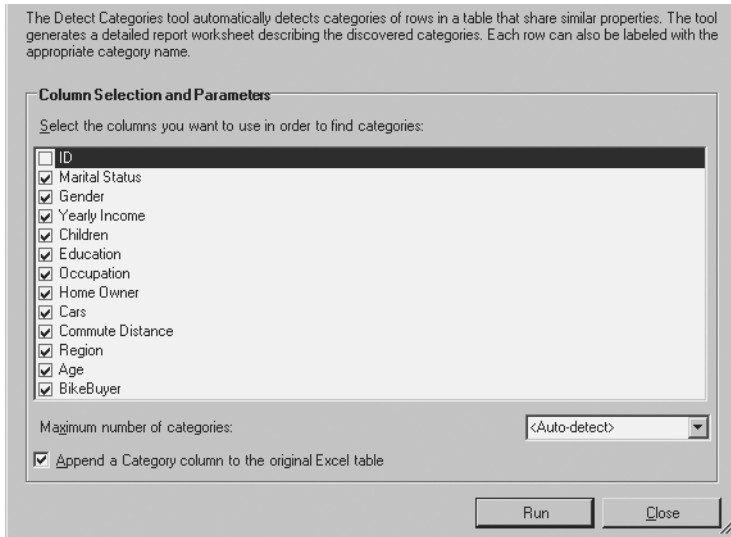

Figure 1-9. *The Data Mining Add-In's friendly name*

12. When the "Connect to Analysis Services" window is closed, the Analysis Services Connections window will open automatically. Utilize this window to make the sure that DMAddinsDB (localhost) is the current connection by clicking the Make Current button displayed in Figure 1-10. Click Close.
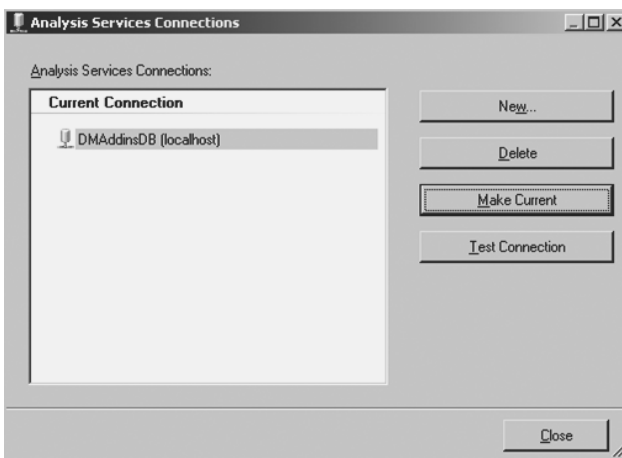

Figure 1-10. *The Make Current button*

13. Navigate to the Training Data sheet and the Analyze tab. This can be found by looking at the sheet names at the bottom of the workbook. The sheet will be called Training Data. Click a cell in the table, and click Detect Categories.

14. Review the columns shown in Figure 1-11, and decide which columns to select. In this case, everything but the ID column should be selected.

The Detect Categories tool automatically detects categories of rows in a table that share similar properties. The tool generates a detailed report worksheet describing the discovered categories. Each row can also be labeled with the appropriate category name.

Column Selection and Parameters

Select the columns you want to use in order to find categories:

- ☐ ID
- ☑ Marital Status
- ☑ Gender
- ☑ Yearly Income
- ☑ Children
- ☑ Education
- ☑ Occupation
- ☑ Home Owner
- ☑ Cars
- ☑ Commute Distance
- ☑ Region
- ☑ Age
- ☑ BikeBuyer

Maximum number of categories: `<Auto-detect>` ▼

☑ Append a Category column to the original Excel table

[Run]   [Close]

Figure 1-11. *Column selection*

15. Click Run, and the window displayed in Figure 1-12 will be shown.

Detect Categories...

Initializing... -- Done!
Reading Spreadsheet Data... -- 19%

[Cancel]
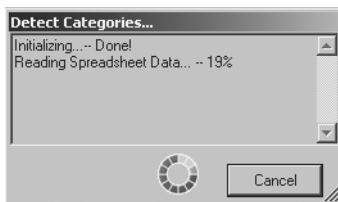
Figure 1-12. *Detecting categories*

16. Once the Detect Categories dialog closes, a series of graphs and charts will be displayed. Use the filter on the category characteristics to see different categories outlined in Figure 1-13. Rename the categories to something more fitting by editing the entries in the Category Name column of the category graph shown in Figure 1-14.

Category	Column	Value	Relative In
Category 3	Yearly Income	Very Low:< 38959	
Category 3	Region	Europe	
Category 3	Occupation	Manual	
Category 3	Commute Distanc	0-1 Miles	
Category 3	Cars	0	
Category 3	Occupation	Clerical	
Category 3	Children	2	
Category 3	Age	Medium:48 - 59	
Category 3	Children	1	
Category 3	Home Owner	Yes	

Figure 1-13. *Category characteristics*

Category Name	Row Count
Educated, Short Commute	1259
Low Income, Long Commute	1096
Low Income, Short Commute	840
High Income, Long Commute	762
Low Income, Young	788
Educated, High Income	670
Low Income, Children	650
Low Income, Many Cars	495
Medium Income, Children	440

Figure 1-14. *Category names*

17. Navigate back to the Training Data sheet. All customers are categorized by their names. Customers can be filtered based on categories and targeted for marketing campaigns. For example, people in the "Medium Income, Children" category could be sent e-mails at the winter holidays describing new children's bikes. People in the "Educated, Short Commute" category could be sent e-mails in the spring detailing the benefits of bicycle riding for individual health and the environment.

Occupation	Home Owner	Cars	Commute Distance	Region	Age	BikeBuyer	Category
Professional	Yes	4	1-2 Miles	Pacific	41	No	Medium Income, Children
Professional	Yes	0	0-1 Miles	North America	47	Yes	Medium Income, Children
Professional	Yes	0	0-1 Miles	North America	37	Yes	Medium Income, Children
Professional	Yes	0	2-5 Miles	Pacific	40	Yes	Medium Income, Children
Professional	Yes	0	0-1 Miles	North America	48	Yes	Medium Income, Children
Professional	Yes	0	0-1 Miles	Pacific	36	Yes	Medium Income, Children
Professional	No	1	0-1 Miles	Pacific	38	Yes	Medium Income, Children
Professional	No	4	10+ Miles	Pacific	32	Yes	Medium Income, Children
Professional	Yes	0	1-2 Miles	Pacific	39	Yes	Medium Income, Children
Professional	No	3	10+ Miles	Pacific	34	Yes	Medium Income, Children
Professional	Yes	2	10+ Miles	Pacific	30	Yes	Medium Income, Children
Professional	Yes	0	0-1 Miles	Pacific	36	Yes	Medium Income, Children
Professional	Yes	4	10+ Miles	Pacific	38	Yes	Medium Income, Children
Professional	Yes	4	10+ Miles	Pacific	35	Yes	Medium Income, Children

**Figure 1-15.** *Additional information*

18. Spend some time navigating through the other tabs and information shown in Figure 1-15. Explore the different ways the tools can be used.

The final exercise in this chapter, Exercise 1-3, helps you to familiarize yourself with the Microsoft tools used to build and maintain SQL Server Reporting Services (SSRS) reports. For many organizations, SSRS will serve as a foundation for beginning an analytics program. We like to introduce SSRS at this time to demonstrate that building reports is within the skill set of both developers and technical analysts and to provide a foundation for the types of data connections you will use when developing reports and cubes for future exercises.

Exercise 1-3. Building a Basic SQL Server Reporting Services Report

The exercise will walk through the initial creation of a reporting services report. The report, entitled Sales Report, will render information regarding opportunities won in the past year and any open opportunities. Selected parameter values will determine the data displayed.

1. SQL Server Business Intelligence Development Studio (BIDS) ships with SQL Server 2008. When you install SQL Server you must make an active decision to install BIDS. Navigate to Start ➤ All Programs ➤ SQL Server ➤ SQL Server Business Intelligence Development Studio.

2. Navigate to File ➤ New ➤ Project. Select the Report Server Project, as shown in Figure 1-16.
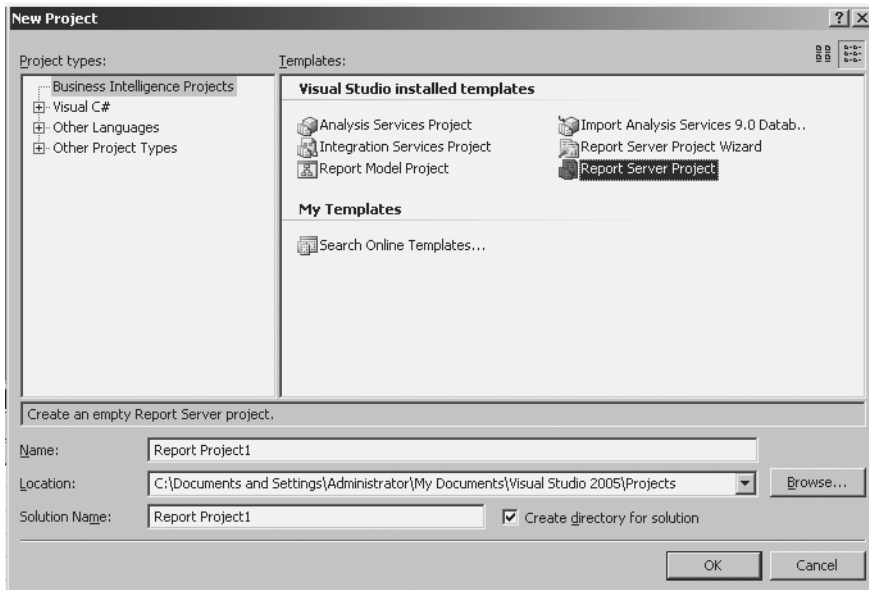

Figure 1-16. *Select Report Server Project.*

3. Give the report a name, and click OK.

4. When the Solution Explorer window shown in Figure 1-17 opens, right-click Reports, click Add, and then click Add New Item.
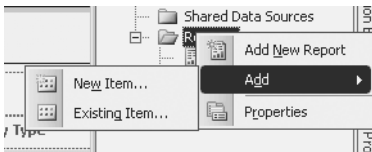
Figure 1-17. *Adding a new item*

5. Select Report, and type **Opportunities.rdl** for the report name. Click OK. The report will open to the dataset page. Click the New drop-down in the Report Data section to create a new data source. The Data Source Properties dialog box, shown in Figure 1-18, is used to create a connection to the database.
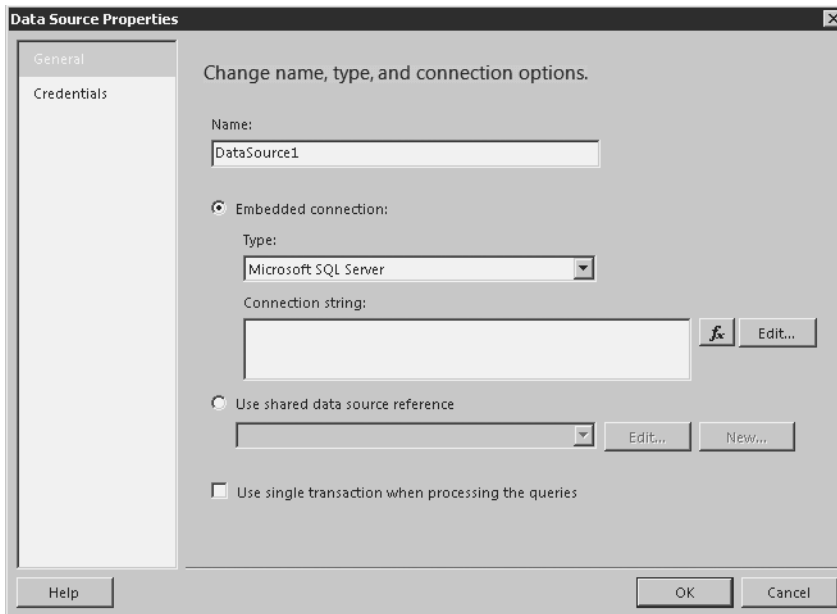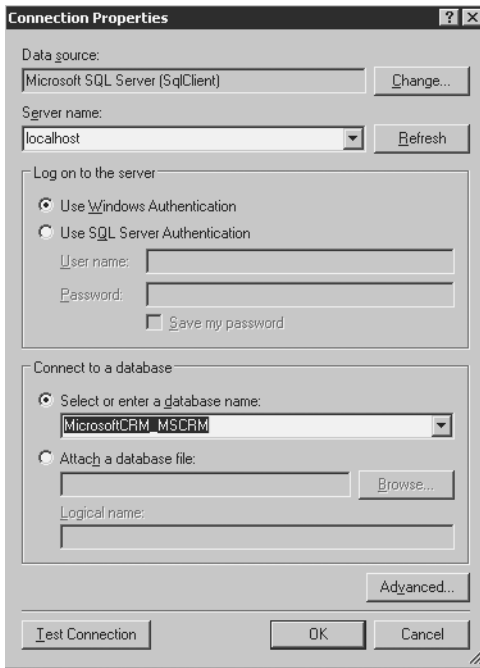

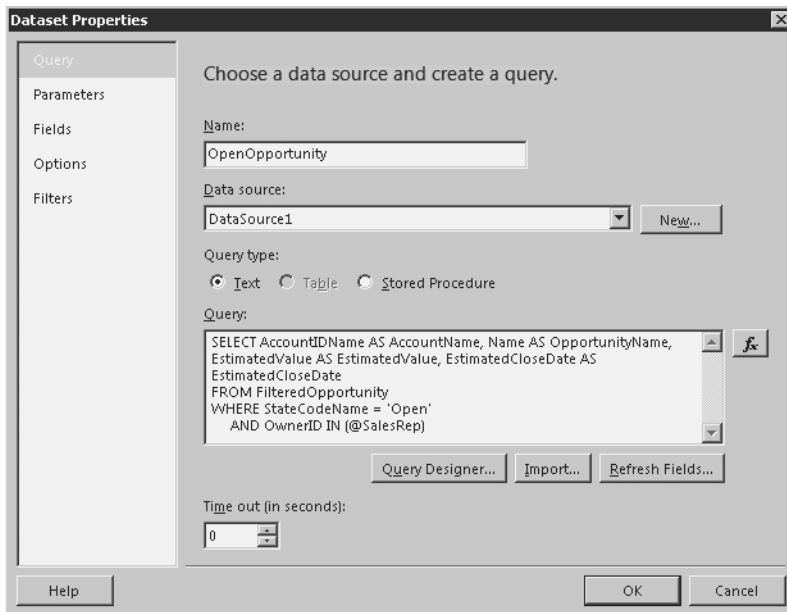**Figure 1-18.** *The Data Source Properties dialog box*

6. Click the Edit button to configure the connection string.

7. In this example, the report will be built on information based in Microsoft CRM. Since the database for Microsoft CRM is located on the same sever the report is being built on, the server is localhost. The database that contains information from CRM is called Contoso_MSCRM. Set the server name to localhost and the database name to Contoso_MSCRM, as shown in Figure 1-19, and click OK. Click OK again to close the Data Source Properties window.

8. Right-click the newly created data source, and click Add DataSet. Change the name of the dataset to **OpenOpportunities**. In the query window, type the following text:

```
SELECT AccountIDName AS AccountName,
Name AS OpportunityName,
EstimatedValue AS EstimatedValue,
EstimatedCloseDate
AS EstimatedCloseDate
FROM FilteredOpportunity
WHERE StateCodeName = 'Open'
  AND OwnerID IN (@SalesRep)
```

When these tasks are complete, the dataset will resemble Figure 1-20.
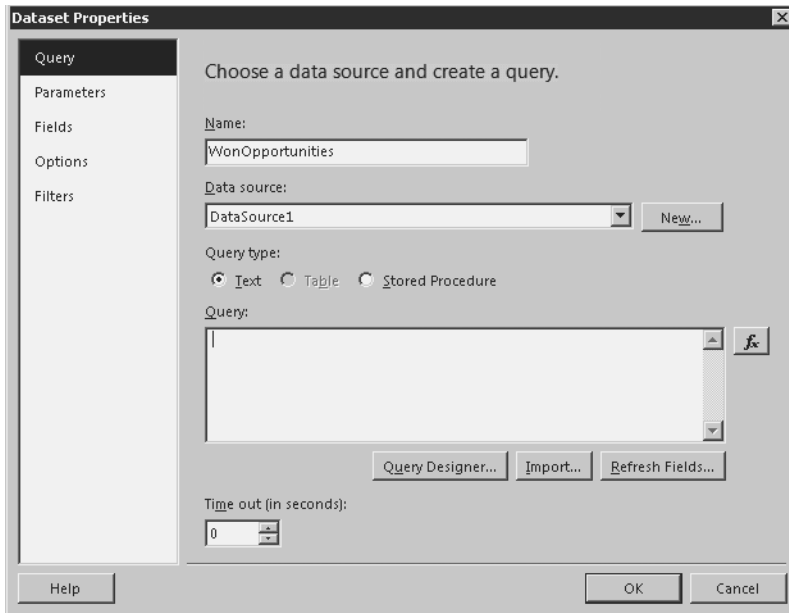

Figure 1-19. *The Connection Properties dialog*

Figure 1-20. *The Dataset Properties dialog*

9. Click OK to close the window.

10. Create a new dataset by right-clicking the data source and clicking New Dataset. In the dialog displayed in Figure 1-21, type **WonOpportunities** in the Name field.

Figure 1-21. *Adding a new dataset*

And type the following SQL code into the Query text box:

```
SELECT DATENAME(MONTH, ActualCloseDate) AS ActualCloseMonth,
MONTH(ActualCloseDate) AS AcutalCloseDateMonthNumber,
SUM(EstimatedValue)
AS EstimatedValue
FROM FilteredOpportunity
WHERE StateCodeName = 'Won'
   AND YEAR(ActualCloseDate) = YEAR(GetDate())
           AND OwnerID IN (@SalesRep)
GROUP BY  DATENAME(MONTH, ActualCloseDate),MONTH(ActualCloseDate)
ORDER BY MONTH(ActualCloseDate) ASC
```

11. Click OK. A window will pop up requesting that you define the query parameters. This box is useful to check the SQL syntax. However, it isn't necessary in this scenario. Click OK to close the window.

12. Create a new dataset named SalesReps. Insert the following SQL code into the Query box:

```
SELECT FullName, SystemUserID FROM FilteredSystemUser
```

13. Click OK to close the window. Expand the Parameters folder in the Report Data window. Double-click the SalesRep parameter to configure the parameter properties.

14. Figure 1-22 shows the general screen for the parameter properties. Set the properties of the SalesRep parameter, which will list all of the sales representatives in the SalesRep dataset. Users will have the ability to select one, all, or a few representatives. The report's default behavior will be to run for all sales representatives. Set the values in the Report Parameter Properties dialog as follows:

The General tab:

- For Name, type **SalesRep**.

- For Prompt, type **Sales Rep**.

- For "Data type," type **Text**.

- Check "Allow multiple values," and leave "Allow null value" and "Allow blank value" blank.

- Set the parameter visibility to Visible.

The Available Values tab:

- Set available values to From Query.

- Set the Available Values to "Get values from a query," the Dataset field to SalesRep, the Value field to SystemUserID, and Label field to FullName.

The Default Values tab:

- Set the Default Values to "Get values from a query." Set the dataset to SalesReps and the Value field to SystemUserID.


Figure 1-22. *General report parameter properties*

15. Click OK to leave this screen. Check to see if the toolbox is located on the left-hand side of the project. If it is not, navigate to View ➤ Toolbox to add the toolbox to the view.

16. One informative and visually appealing way to display revenue from won opportunities over time is to chart the information in a graph. Click the chart icon, and drag it onto the canvas. Once the chart is on the canvas, a window will automatically be displayed that requires the report creator to select a chart type. Select the smooth area chart. Click OK. The results are shown in Figure 1-23.
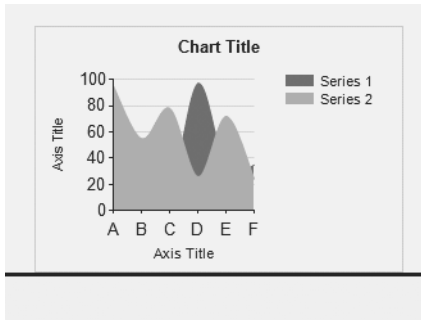
Figure 1-23. *A smooth area chart*

17. Change the chart title from Chart Title to Won Opportunities by clicking the chart title and typing in the new title. Set the title format to Calibri. Right-click the graph, select Chart Properties, and set the color palate to "pastel."

18. Drag the EstimatedValue field from the WonOpportunites dataset on to the section of the graph entitled "Drop data fields here." Drag ActualCloseMonth from the WonOpportunities dataset into the "Drop category fields here" section. The graph will resemble Figure 1-24.
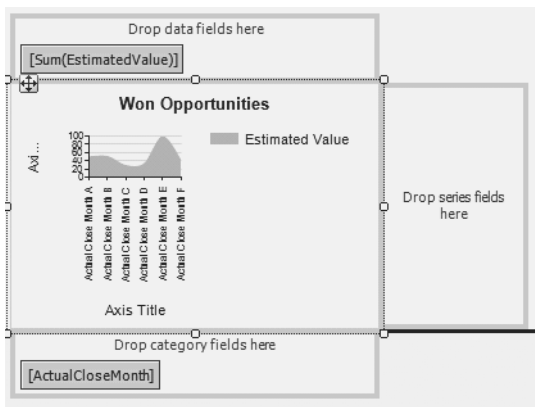
Figure 1-24. *A sample graph design*

19. Click the legend to highlight it. Delete the legend.

20. Delete both axis titles by highlighting each title and clicking the Delete key on the keyboard.

21. Right-click the y axis, and navigate to the properties. Navigate to the Number tab. Select Currency. Set the decimal places to 0, and check the "Use 1000 separator (,)" box. The properties page will look like Figure 1-25.
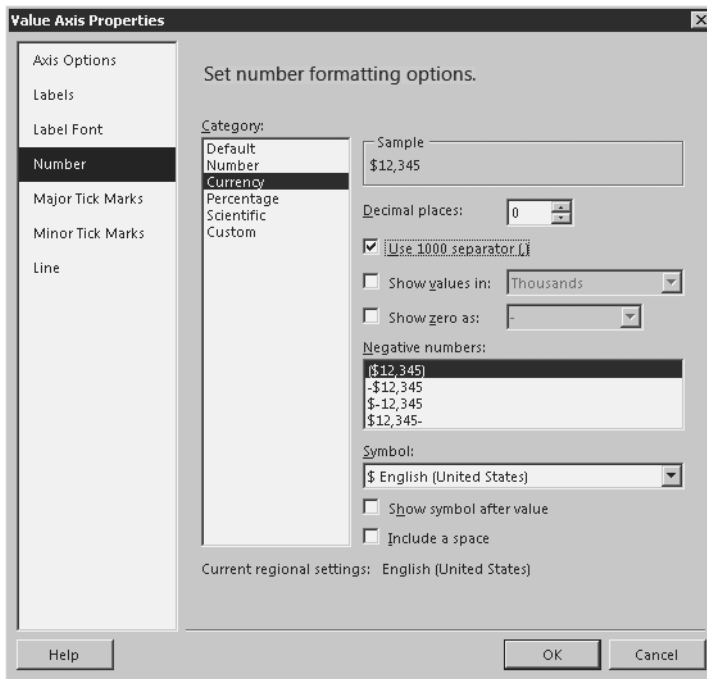

Figure 1-25. *The Value Axis Properties dialog*

22. Drag a text box on to the canvas. Click the text box, and type **Sales Report**. Make the text box larger, and move it to the top of the canvas, as shown in the upper left-hand corner of Figure 1-26.


Figure 1-26. *Adding a text box*

23. Click the text box, and use the toolbar shown in Figure 1-27 to set the font to Calibri and font size to 24pt and to center the text.
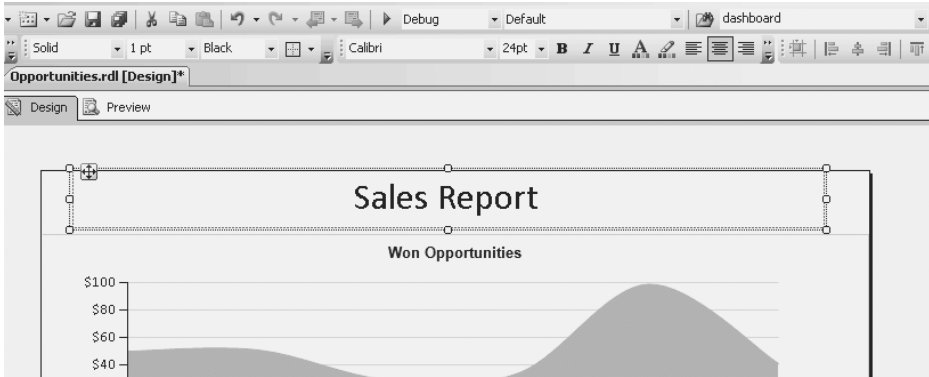
Figure 1-27. *Updating the report font*

24. Open the toolbox. Sales representatives need to keep track of a number of open opportunity metrics, and open opportunity information is easiest to view in a table. Click and drag a table onto the canvas. The table has three rows: one for the header, one for the footer, and one for the details. The details row is marked by three black lines. The table also has three columns.

25. Insert a column to the right by right-clicking a column and clicking Insert Column ➤ Right, as displayed in Figure 1-28.
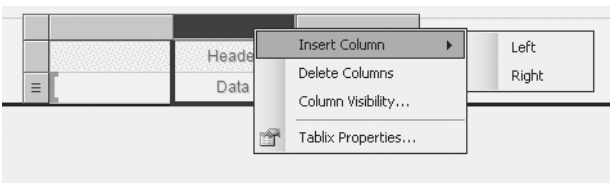
Figure 1-28. *Inserting a column*

26. Figure 1-29 shows what the table will look like with data allotted for each cell. Navigate to the OpenOpportunities dataset. Expand the dataset to view that dataset's fields. Click and drag the following fields from the dataset into the appropriate columns on the table:

   - *OpportunityName*: The detail cell of the first column on the table

   - *AccountName*: The detail cell of the second column

   - *EstimatedCloseDate*: The detail cell of the third column

   - *EstimatedValue*: The detail cell of the fourth column

Figure 1-29. *The updated report*

27. Highlight the table, and use the toolbar to change the font to Calibri.

28. Highlight the top row, and change the background to MediumSeaGreen by clicking the background color button on the toolbar.

29. Right-click the header row, and click Insert row ➤ Above. Highlight the new row and right-click. Choose Merge Cells. Use the properties window to change the background color to SeaGreen, text color to White, font size to 14pt, and text alignment to Center. Type **Open Opportunities** into the new row.

30. To create a totals row at the bottom of the table, right-click the details row, and click Insert Row ➤ Outside Group—Below. Navigate to the totals row. Mouse over the cell where the totals row and the Estimated Value column intersect. A table icon will appear. Click the icon, and select EstimatedValue. The new cell will read [Sum(EstimatedValue)]. The result is shown in Figure 1-30.

Figure 1-30. *Adding row totals*

31. Format the dollar values for Estimated Value by highlighting the applicable cells and navigating to Properties. Set the Format cell to $#,###.

32. The report is ready to be deployed to a report server. Navigate to the Preview tab to see how it looks when an individual runs the report. Figure 1-31 provides an example of this.
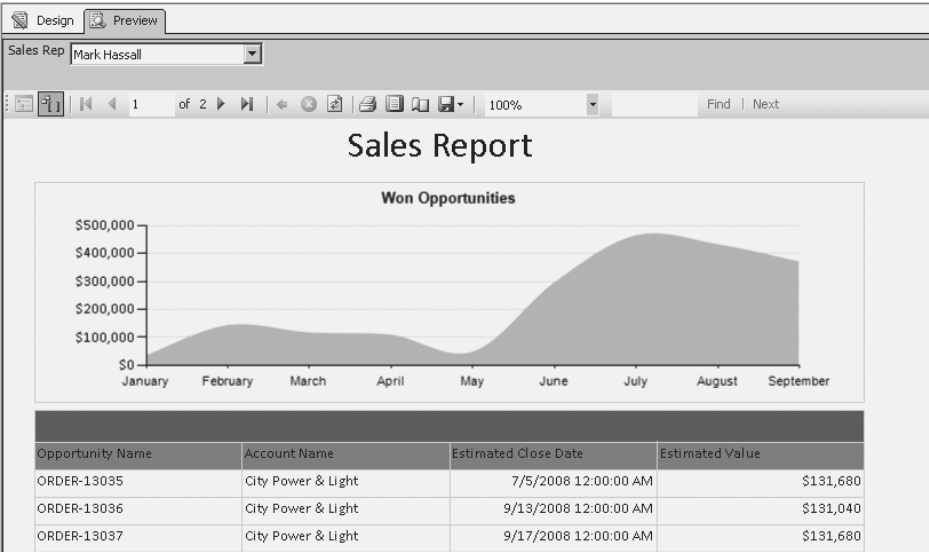
Figure 1-31. *Previewing the report*

33. For extra practice, try to do the following on your own:

a. Add more columns to the open opportunities table for the Owner and Modified On.

b. Alphabetize the names of the sales representatives in the drop-down.

c. Enable sorting on the Opportunity Name, Account Name, and Estimated Value fields.

Summary

This chapter provides a background on analytics and on a number of the tools that will be used throughout this book. The exercises in future chapters will add more detail and complexity to both the product and the analytics system.

In general, most organizations need to go beyond Excel reporting and out-of-the-box application reports but don't understand how to get started and how to plan a long-term strategy. The two upcoming chapters of this book will help in those departments by providing guidelines for successful implementations. These chapters also assist with the most fundamental of tasks, the development of key performance indicators (KPIs).