

PART I

Getting Started

- ▶ **CHAPTER 1:** Introducing Reporting Services
- ▶ **CHAPTER 2:** Reporting Services Installation and Architecture
- ▶ **CHAPTER 3:** Configuring SharePoint Integration

1

Introducing Reporting Services

WHAT'S IN THIS CHAPTER?

- Understanding report designer roles and the tools used to design reports
- Understanding dashboards, reports and applications
- Examining Business Intelligence solutions
- Discovering multidimensional and tabular semantic models

You're holding this book, trying to decide if it will help you solve a problem or teach you essential skills to create reports with Reporting Services. If you and I were having this conversation in person, I'd ask you to tell me what you need. I teach classes and travel to companies to create report and BI solutions, and at the beginning of every class or consulting engagement, I ask what the student or client needs. What are the requirements? What questions does your report need to answer? What's not working? What needs to be fixed, and what will it take to build a solution to help you reach your goals? So, I ask you, what do you need? Why are you reading a book about Reporting Services? Do you have a specific problem to resolve, or do you just need to develop some basic report design skills? Do you need to build an entire reporting solution? Who are the users of these reports? Are they department workers, business managers, or financial analysts? Maybe your user is the CEO of a major corporation or other business executives who need to know if the company is on the right track. Maybe you need to create reports for your own business to make sure it's profitable and achieving its goals. Whether you are creating an invoice to sell arts and crafts out of your garage or a BI dashboard to help manage a multinational corporation, the reports you will create are important. Therefore, you need to make sure they deliver accurate information and are designed correctly, using industry best practices. That's a big responsibility.

Whatever your needs, we'll cover all these bases and address each topic thoroughly. I've enlisted some of my most trusted associates to share their experiences.

This chapter is a high-level introduction to the concepts and capabilities of this powerful reporting tool and the data analysis platform of Microsoft SQL Server 2012. It introduces common reporting scenarios, beginning with the most basic and then moving to the more advanced. In subsequent chapters, you will explore these capabilities in depth and learn how to use them in your own reporting solutions.

SQL Server Reporting Services has grown to become the de facto industry standard reporting tool by which others are measured. It is a foundation upon which you can construct complete report, scorecard, and dashboard solutions for business users. Today, it does everything from simple ad hoc data reporting to delivering enterprise-ready integrated reporting into business portals and custom applications.

Not long ago, the information technology (IT) group for a large financial services company wanted to make sure that they were using the best reporting tool on the market. They decided to hire a consulting company to evaluate every major reporting product and give them an unbiased analysis. I was lucky to land this assignment. We worked with the client to identify about 50 points of evaluation criteria. Then I contacted all the major vendors, installed evaluation copies and explored features, and spoke with other customers and with those who specialized in using these various products. It really helped us see the industry from a broad perspective and was a valuable learning experience. There are some respectable products on the market, and all have their strengths, but I can honestly say that Microsoft has a unique and special platform. As a consultant, contractor to Microsoft, and Microsoft SQL Server MVP, I have had the opportunity to work alongside the Reporting Services product team for many years. They have a vision, and they're passionate about their product. I have a great deal of respect for the fine people who continue to develop and improve Reporting Services, version after version.

WHO USES REPORTING SERVICES?

Business users fit into a few categories when we consider how they use reports. Some are report consumers only. They're happy to use reports that have been written and published for them. Others prefer to create their own reports using business tools they understand and use for other things, as Excel is used for planning and financial analysis. Maybe they just want to browse information to look for trends and to understand how the business is measuring up against their goals. Still other business users want to use more sophisticated tools to create powerful reports. A typical information technology group at most large organizations has three common roles: system administrators, application developers, and project managers. Usually everyone else in the department supports these roles. Where does the report designer fit in the organization? Good question. Honestly, I don't have a simple answer. The fact is that people who design business reports don't come from a common pool of IT professionals. In fact, many people who spend the majority of their time creating reports are part of the business community and are not your typical hard-core computer geeks.

Microsoft has a long history of building highly technical products that appeal to the technical community. In more recent years, Microsoft has begun to enhance its product culture for more suit-and-tie-wearing folks who talk about things such as business performance management strategy and market share rather than remote procedure calls and polymorphic object inheritance. If you're a business-type person, you probably don't care about integrating your reports into custom

applications and web sites or writing complex programming logic to make them sing and dance. Some of us live for that. What you may care about is giving your savvy business user the ability to drag and drop report parts from the gallery to visualize important key metrics to see what products are performing well in their sales territory. However, to enable that experience, a certain degree of technical expertise is necessary.

Over the years, I've taken inventory of the people who consider themselves report designers. They generally fall into one of two camps — business-focused or technology-focused. There has been a significant shift toward more accessible reporting tools for those who have less technical aptitude. The following roles represent the majority and describe some of the trends we're seeing as the industry continues to evolve.

Business Information Workers

People in this role have strong computer skills, but they don't spend their time writing code and using programming tools. Their primary interest is exploring information and finding answers, rather than designing complex reports. If you're an information worker (IW), you need easy-to-use tools to browse data and create simple reports quickly and with less technical expertise. IWs typically create a report to answer a specific question or address a particular need, and then they may discard the report or save it to a personal area for reuse. They tend to create a separate report for each task and may or may not share these reports with others who have similar needs. This is by far the largest and fastest-growing group of report tool users in the industry.

Business Managers

If you're a business manager, you're primarily interested in your own domain of the business. Managers need reports to support specific processes to address their analytical needs and to help them make informed decisions. Like information workers, they have little interest in the implementation details or technology used to make it work. As information workers, managers may create their own reports to analyze the productivity of their team or area of responsibility.

Software Developers

To achieve advanced reporting features, software developers write complex queries and custom programming code to process business rules and give reports conditional formatting and behavior. Developers typically feel right at home with the report design environment because it's similar to familiar programming tools. However, report design is not the same as application development. Designing a report can be faster and easier in some ways than developing software. Advanced report design can involve writing code and even developing custom components.

System Administrators

If you are a system administrator, you are typically concerned with the setup and ongoing maintenance of servers and the infrastructure to keep reporting solutions available and working. Administrators typically spend their time and energy managing security and optimizing the system for efficiency. Reporting Services has an administrative component that is especially important in large-scale implementations.

In smaller organizations, the same person may play the role of system administrator, developer, and report designer. Reports can also be created to help monitor system usage and maintenance statistics to make a system administrator's job easier.

Reporting Services meets the needs of information workers and technology professionals with different report design and data browsing tools. Report Builder is simple, focused, and familiar, with a user interface similar to Microsoft Office applications. A more advanced report designer, called SQL Server Data Tools (SSDT), is available to application developers and other technical professionals. It is integrated into the Visual Studio solution design environment shell.

DASHBOARDS, REPORTS, AND APPLICATIONS

From a software perspective, a computer system can present data to a user in different ways. Most legacy reporting systems ran on the client desktop computer.

Just recently we've seen a major shift toward self-service reporting. Many different tools have been developed to provide the right balance between simplicity and capability. It's taken a few years for the dust to settle on all these tools and for the best options to emerge. Most business leaders want a simple dashboard interface that answers key business questions, or a tool they can use to browse a simplified view or semantic model without having to design reports and write queries. This led Microsoft to take two different approaches. First, it created a streamlined report design tool, Report Builder, that makes it easier for less-technical users to design powerful reports using the conventional report definition architecture. Second, it developed a separate visualization tool, Power View, that leverages the Reporting Services architecture but offers a design and user experience that is separate from conventional reports. To manage expectations, we'll omit the Power View tool from our discussions and treat it as separate from the rest of Reporting Services. You'll read about tabular models and Power View in Chapters 12 and 13.

Quite a few years ago we saw a shift from client-based processing toward applications that ran on web servers. This has proven to be an effective way to make systems available to a large number of people. When Reporting Services was first released, it was available only as a server-based solution, with reports delivered almost exclusively through the web browser — and this is primarily how SQL Server Reporting Services (SSRS) reports are used today. However, the capabilities don't stop there. Reporting Services lets you run reports in a variety of modes and applications. If we've learned anything from the past 20 years of computer system evolution, it's that centralized server-based solutions and client-side applications each offer unique advantages and trade-offs in terms of features, capabilities, interactive user experience, and scalability.

It's important to note that you can install the product and then design and use reports without a lot of fuss and technical expertise. Later in this book, we will discuss how Reporting Services can be used in more advanced and creative ways.

Blurring the Line Between Applications and Reports

With Reporting Services, you can integrate reports into applications in such a way that users may not be able to tell the difference between the report content and the application interface. With a little bit of programming code, reporting features can be extended to look and act much like

applications. When do reports start replacing application functionality? What, exactly, is the difference between a report, a dashboard, and a scorecard? The lines have become quite blurred. Your task is to decide which tool best meets your needs. Many intranet sites run on web portals, rather than custom-programmed web sites, and Reporting Services naturally plays well in practically any web portal environment. In particular, Reporting Services has native integration with Microsoft SharePoint Server.

The exciting news is that you now have a tool that can do some incredible things. As my favorite superhero's uncle said, "With great power comes great responsibility." If you are a simple report designer with simple needs, the good news is that using Reporting Services to design simple reports is, well, simple. If you are a software developer and you intend to use this powerful framework to explore the vast reaches of this impressive technology, welcome to the wonderful world of creative custom reporting.

After years of experience with this product, I've learned an important lesson on this topic. They say that to a hammer, everything looks like a nail. Likewise, to a programmer, a lot of challenges may look like an opportunity to write program code. That may be the right solution under certain conditions. But often, the most effective solution is to simply use a feature already baked into the product — and implement that feature as it was designed to work. I often have this conversation with programmers after they have spent hours writing a complicated solution to a simple problem. My father often advised me to follow the well-known KISS principle: "Keep it simple, stupid." The last part was just to make the phrase memorable, and he meant it in the most affectionate way (at least, I'm pretty sure he did).

Launching Reports from an Application

Hyperlinks and application shortcuts can easily be added to documents and custom applications. Much of the standard report-viewing environment may be controlled using parameters passed to the report server in the URL. Reports may be designed to prompt users for parameter values used to filter data and to modify the report format and output. These parameters may also be incorporated into a URL string. This way, one hyperlink displays a report with one set of data, and another hyperlink displays the same report with different data. Parameters can even be used to change display attributes such as font sizes and colors, and to hide and show content.

User Interaction

In the past, many reports were nothing more than a list of values with totals. Now reports can be a starting point that can guide users to the information they need to make decisions. Report elements such as text labels, column headers, and chart data points can be used to navigate to different report sections and new reports. Because navigation links may be data-driven and dynamically created based on program logic, report links may also be used to navigate into business applications. Imagine using your reports to launch programs and to navigate to document libraries and online content!

As the user clicks items or data points in the report, content and the layout of a report can change based on parameter values set in the background. Summary headers may be used to expand and collapse detail sections, giving users the ability to drill down to more specific information, as shown in Figure 1-1.

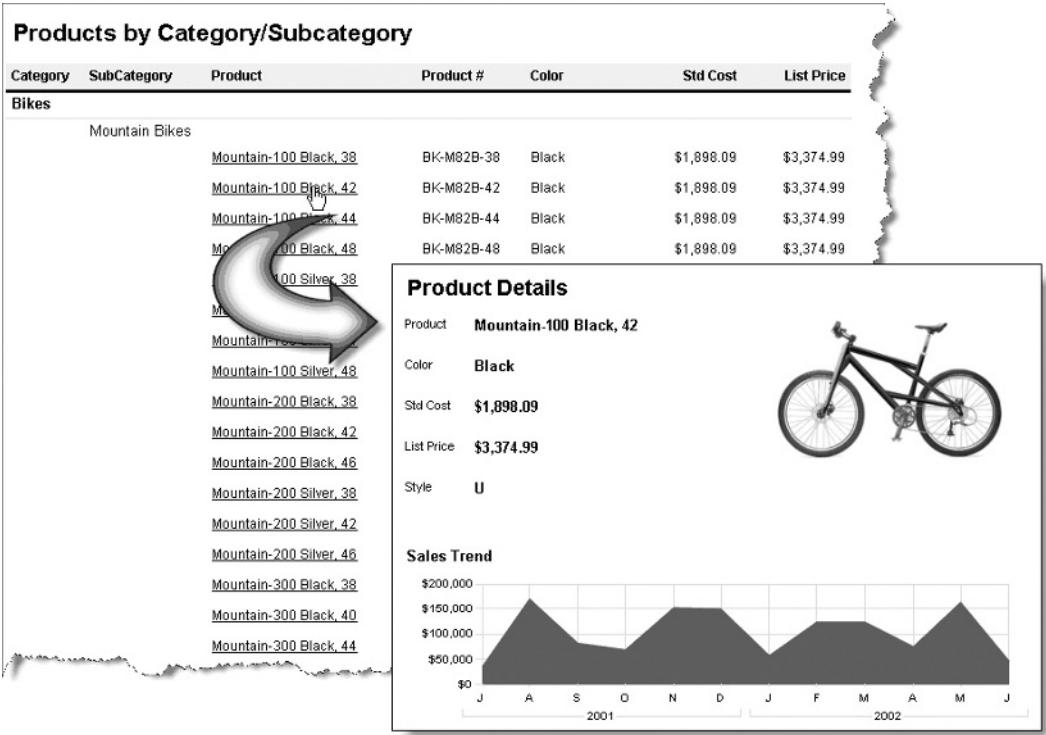


FIGURE 1-1

Integrating Reports and Applications

One of the reasons that Reporting Services integrates so easily with modern web applications is that it natively supports Hypertext Markup Language (HTML), the standard markup language used to create web pages. Techniques may be used to incorporate reports into a web application in a variety of ways:

- Hyperlinking to navigate the web browser window to a report
- Hyperlinking to open reports in a separate web browser window, with control over report display and browser features
- Embedding reports into a page using a frame, IFrame, or ReportViewer web control
- Programmatically feeding report content to an Active Server Page (ASP or ASPX) using server-side custom code
- Programmatically writing reports to files available for download from a web site
- Using a web part to embed reports into a SharePoint Web portal
- Fully integrating the report server in SharePoint Integration mode

There are a lot of creative ways to integrate reports into a web or desktop application. These techniques range from simple, requiring a little HTML script, to complex, custom methods. And if it's

not enough to be able to embed reports into custom web pages, it's also possible to use custom program code to embed additional content into reports. Imagine the possibilities. Actually, you don't have to imagine anything. Just keep reading!

SharePoint integrated mode allows all your reports and report administration to be managed completely within SharePoint. If you choose to manage the report server separately from your SharePoint portal, you can still use SharePoint web parts to navigate folders and reports and to view reports hosted on the report server running in Reporting Services Native mode.

The ReportViewer control or embedded web browser may be used to view server-based reports in a form. These reports are still managed on the report server and maintain all the security settings and configuration options defined by an administrator. Queries and data access are still performed on the server. The other option is to embed these reports directly into the client-side application. The Windows Forms ReportViewer controls can act as a lightweight report-rendering engine. This means that reports built into a custom application can run independently from the report server.

ENTERPRISE REPORTING

Delivering reports to the masses requires a capable reporting environment. Rather than bringing data from source databases to the desktop for processing, Reporting Services processes queries and then renders reports on the report server. Because it uses Windows Services, shared server-based components, and HTTP web services, all the processing occurs in an efficient and secure environment. Standard data-source connection providers for SQL Server and other enterprise-class databases promote efficient use of server resources. In simple terms, this means that many users can run reports at the same time while consuming minimal server resources.

A business intelligence solution integrates data from multiple sources into a data warehouse, data mart, or semantic data model. Complex analysis solutions often require tabular or multidimensional data structures. If you're not familiar with the terms "OLAP," "tabular," and "semantic model," this might at first be a little confusing. A quick history lesson will clear that up. In the late '90s Microsoft released a product that came with SQL Server 7.0, called "OLAP Services." This was their multidimensional database technology that performed online analytical processing, storing data in cubes and dimensions, rather than tables. In SQL Server 2000, OLAP Services became Analysis Services. Multidimensional (OLAP) databases store data in a pre-grouped and pre-aggregated format on disk so the data is available quickly for reporting and browsing. In SQL Server 2012, Microsoft rebranded their analytical database technologies as "Business Intelligence Semantic Models" or "BISM." They offer two technologies under the BISM umbrella; multidimensional (OLAP) and tabular models, which store and process analytic functions in-memory. In some cases, tabular models are easier to design and may be more efficient and faster for reporting and analysis. The chapters in Part III help you understand the advantages of BI data sources and how reports are designed to work with analytic data and semantic models.

The Reporting Services report server exposes its functionality in the same way that a standard ASP.NET web site is hosted for users. Reports may be accessed from anywhere within or outside of the corporate firewall and are still available only to selected users. In SharePoint integrated mode, reports are available to users through document libraries and are secured and managed within the SharePoint server environment. In Native or nonintegrated server mode, reports are managed

through a web application called *Report Manager*. Reports can also be exposed in custom-developed web applications using practically any set of web technologies or development tools.

SOLUTION TYPES

An impressive aspect of Reporting Services is that there are so many different ways to implement reports into a business environment. However, giving people a lot of choices doesn't necessarily solve their problems. In fact, providing users with too many options can just be confusing and overwhelming. As report system designers, our job is to provide the right kind of solution for our users that is simple, uncluttered, and easy to use. Reporting Services has become such a multifaceted platform that we often must clarify what we mean when talking about "Reporting Services reports." Part V, "Solution Patterns," gives you prescriptive guidance about how to create and manage reporting solutions using best practices and solution patterns.

The majority of new Reporting Services implementations for most organizations use the de facto Web-based Report Manager interface or are integrated into a company SharePoint site. Other options to integrate reports into custom applications or web pages may be used to meet specific business needs but are less common. In reality, reports can be integrated into a variety of custom solutions with relative ease. Here are some software solutions that might incorporate reports:

- Out-of-the-box, server-based reporting features, using reports created by report designers and deployed to a central web server.
- Reports integrated into web applications using URL links to open in a web browser window.
- Reports integrated into SharePoint Services applications using SharePoint web parts.
- Custom-built application features that render reports using programming code. Reports can be displayed within a desktop or web application or may be saved to a file for later viewing.
- Interactive data visualizations using the Power View visualization tool for data exposed through a tabular semantic model.

Simple Report Design

If you need to create common report types to summarize or output information contained in a database, Reporting Services offers some great tools that make this easy to do. For example, suppose you have a record of customers and the products they have purchased. You want to produce a list of customers that contains the number of transactions and the total amount the customers have spent. You can use the Report Wizard to produce a table report that includes this information. If you want to compare the sales for each customer, day-to-day, over a period of time, you can use the Report Wizard to generate a line chart report to view the sales trend. The point is that common report types can be easy to create with tools and features that don't require users to know a lot about complicated things like programming, writing queries, and building expressions.

Categorically, report solutions may be created by information technology staff or business users; a variety of tools have been created to support the needs of each group. After a brief explanation, Table 1-1, shown later, summarizes the report designer options we've seen in the current and past product version.

Managing a fully scaled corporate BI solution can be complex and expensive. Fortunately, all the components of a working solution can be scaled down to a single server if necessary. Small and midscale reporting solutions may use a single, multipurpose database serving as an operational data store and a reporting data structure. As the solution matures, the eventual separation of these databases is almost inevitable. A small-scale data mart, populated from operational databases at regular intervals, will provide a simpler data source for reporting that doesn't compete with users and applications for system resources.

Simple reports are easy to design and deploy for short-term use. With a little planning and discipline, you can design reports to meet future requirements. Properly designed, your reports can include advanced features that meet simple needs now and more sophisticated needs in the future.

The five chapters in Part II, "Report Design," begin with the fundamentals of basic report design and then progressively demonstrate how to add more advanced functionality.

IT-Designed Reports

When Reporting Services was first released, the report design experience was optimized for programmers and application developers who were accustomed to using Visual Studio, a product designed to help technically inclined programmers create custom software. When the product team completed the first-generation product, they immediately went to work on a set of tools to enable business users to design their own reports. This offering came to the market in stages with subsequent product releases. A brief understanding of this history will help you better appreciate how we arrived at the current set of tools and capabilities.

For the more technically inclined, the report design experience in the Visual Studio shell, called the SQL Server Data Tools (SSDT), is both familiar and powerful (see Figure 1-2).



In previous versions of SQL Server, the Visual Studio-based design tool was called "Business Intelligence Development Studio" (or BIDS for short). If you hear one of us SSRS old-timers refer to this tool by the former name, you can assume that we're talking about SSDT. It's going to take me a while to get use to this change and to start using the new name.

Simple reports are fairly easy to design, and advanced capabilities are possible using a variety of tools that will make most application developers feel right at home. Like other Visual Studio solutions, report definition files are placed in folders that can be managed as a single deployment unit. Reports, data sources, shared datasets, and all other design elements can be managed with integrated version control in the SSDT shell.

Many technical experts use either Report Builder or SSDT to design their reports. Both tools include a simple Report Wizard that can lead you through designing common reports. Table, grouped, matrix, and chart reports are relatively easy to build just by following the Wizard prompts and setting a few properties.

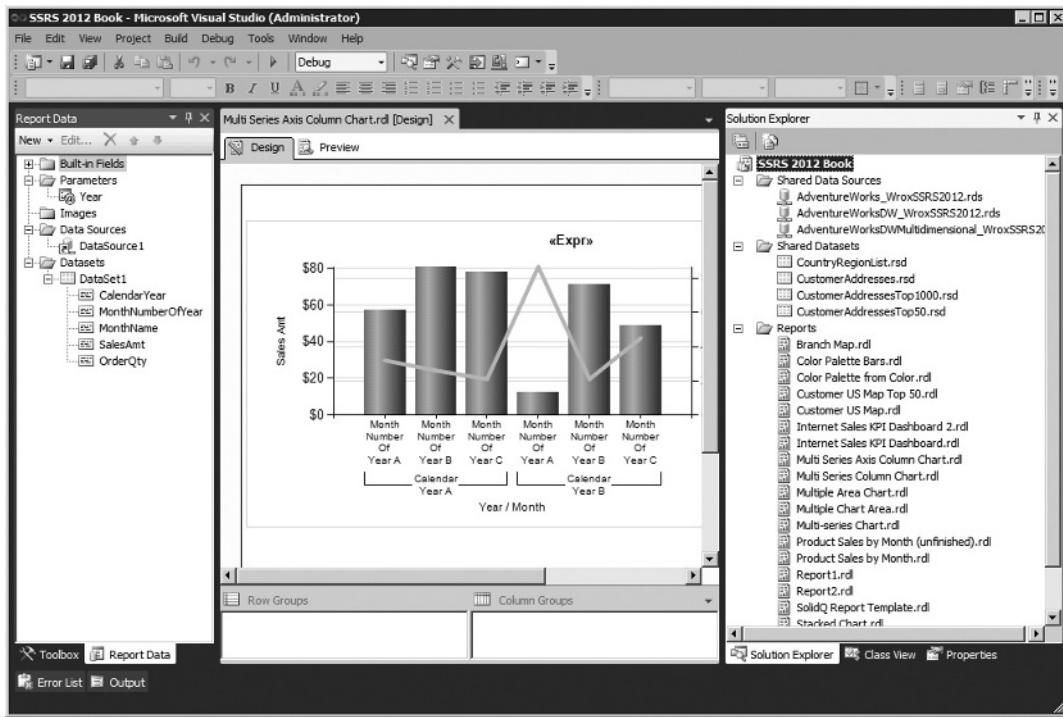


FIGURE 1-2

User-Designed Reports

SQL Server 2012 brings us full circle with two capable self-service design tools. Self-service reporting has been on the minds of many people in the industry for a long time. For Microsoft, the quest to create the perfect easy-to-use BI tool has produced several different products, each with its own unique capabilities. Under the Reporting Services umbrella, two ad hoc reporting tools serve different needs. The current incarnation of Report Builder is based on the mature report definition architecture we've seen progress over the past eight years. Report Builder reports can span the spectrum from simple to complex, with many design options. The Power View visualization report tool introduces an exciting and dynamic data browsing and exploration experience. Power View reports are based on a tabular semantic data model and are surfaced in an intuitive SharePoint-based interactive designer, as shown in Figure 1-3.

The SharePoint environment provides a rich content management and presentation interface for a variety of reports and content types. Figure 1-4 shows the PowerPivot Gallery, with thumbnail previews of Power View reports and PowerPivot workbooks.

Chapters 12 and 13 cover the Power View tool and tabular models for users and for those charged with supporting and enabling this capability in their organization.



FIGURE 1-3



FIGURE 1-4

The first generation of self-service reporting in SSRS was a step toward the robust capabilities in the current product. Report Builder 1.0 was a basic tool introduced with SSRS 2005 that produced a simple but proprietary report with limited capabilities. It was a great tool for its time that allowed users to simply drag and drop data entities and fields from a semantic data model to produce simple reports. Today, the latest version of Report Builder creates reports that are entirely cross-compatible with SSDT and that can be enhanced with advanced features. Consider Report Builder 1.0 yesterday's news. If you're using it now, I strongly suggest making the transition to the newer tool set.

The 2008 product version introduced Report Builder 2.0, a tool that is equally useful for business users and technical professionals. For user-focused designers, Report Builder 2.0 was simple and elegant. Incremental product improvements over the past few versions have made out-of-the-box report design even easier in Report Builder. Users can design their own queries or simply use data source and dataset objects that have been prepared for them by corporate IT so that they can drag and drop items or use simple design wizards to produce reports. In Report Builder, each report is managed as a single document that can be deployed directly to a folder on the report server or in the SharePoint document library. The version number has been dropped from the Report Builder name; now it is simply differentiated from previous versions by the version of SQL Server that installs it. Figure 1-5 shows the current version of Report Builder (installed with SQL Server 2012) with a map report in design view.

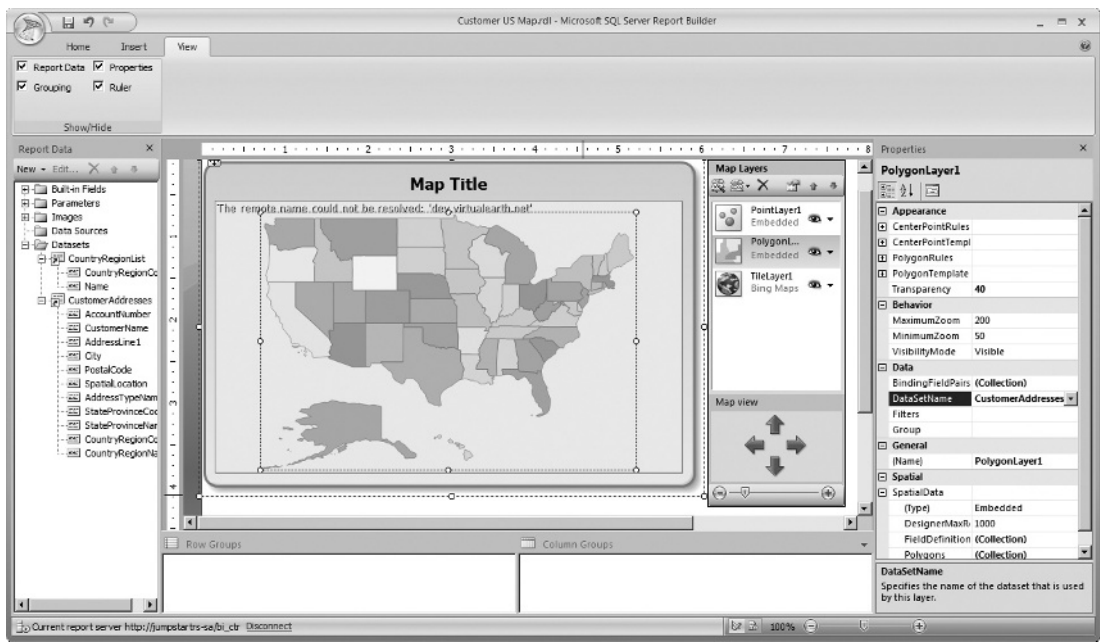


FIGURE 1-5

Table 1-1 summarizes the report design tools available in the past and current product versions.

TABLE 1-1: Report Designer and Visualization Options

REPORT DESIGNER	BACKGROUND	STATUS
SQL Server Data Tools (SSDT)	Implementation of the Visual Studio shell typically used by IT professionals to design reports with a project and team focus. Currently uses the Visual Studio 2010 shell.	Supported.
Report Builder 1.0	Released with SSRS 2005 and supported in SSRS 2008 and 2008 R2.	Limited support. Ongoing use is discouraged.
Report Builder 2.0	Released with SSRS 2008.	Supported for SSRS 2008.
Report Builder 3.0	Released with SSRS 2008 R2.	Supported for SSRS 2008 and 2008 R2.
Report Builder	Released with SSRS 2012.	Supported for SSRS 2008 R2 and 2012.
Power View Visualization	Released with SSRS 2012. Requires SharePoint Enterprise.	Interactive Power View reports are unique and are not cross-compatible with Report Builder or SSDT.

Server-Based Reports

Reports can run on a server or in a stand-alone application on the client computer. It is important to note that Reporting Services is designed and optimized for server-based reporting first. The client-side option, called Local Mode, is possible with some custom programming and takes a little more effort and expertise to implement. For the remainder of this chapter, the discussion is limited to server-based reporting.

It's important to understand the difference between SQL Server Reporting Services and a desktop reporting tool such as Microsoft Access. Reporting Services isn't an application you would typically install on any desktop computer; rather, it is designed for business use. It requires Microsoft SQL Server, a serious business-class relational database management tool. Likewise, reports may be integrated into SharePoint Services to be managed, secured, and administered alongside other shared corporate documents and assets.

For this and other reasons, Reporting Services runs on a server instead of a desktop computer. Whether used in a small business or a huge corporation, Reporting Services is scalable and adaptable for use by a handful or thousands of users and reporting on large sets of data stored in a variety of database platforms. But just because Reporting Services is a business-sized product doesn't mean

that reports have to be complicated or difficult to design. Businesses need not host their own servers and can utilize Azure Reporting, Microsoft's cloud-based report server offering.

Report users need to be connected to a network, or perhaps the Internet, with connectivity to the report server. When a report is selected for viewing from a folder on the report server or the SharePoint library, it is displayed as a web page in the user's web browser. Optionally, the same report can be displayed in a number of different formats, including Word, Excel, and Adobe PDF, or as a PNG, JPEG, GIF, or TIFF image. Reports may be saved to files in these and other formats for offline viewing. Reports may also be scheduled for automatic delivery by the report server by e-mail or may be saved to files. These features are standard and require only simple configuration settings and minor user interaction.

BUSINESS INTELLIGENCE REPORTING SOLUTIONS

A business intelligence (BI) solution is the foundation upon which a capable business reporting platform can be constructed. Depending on your needs and business environment, this may simply entail designing a new database. Just because you need to analyze business data doesn't necessarily mean that you need to build a full-scale BI solution. However, if you need to aggregate large volumes of data to analyze business performance with key metrics and trends, relational databases designed for transaction processing may no longer serve this purpose. Understanding these core concepts and investing in BI before report design will often reduce costs and enable you to create an enduring reporting platform for your business users and leaders.

With the current set of Microsoft reporting and BI technologies (namely, SQL Server and SharePoint), you have some choices. These are discussed in appropriate detail in later chapters. Simply put, the data source for a report could be the relational database used to manage transactional data, a data mart, or a data warehouse stored in a relational database management system (such as SQL Server or Oracle). For better performance and flexible analysis over a large volume of data, data can be stored in a semantic model. This can be either a multidimensional structure (often called a cube) or a tabular semantic model that performs aggregations and calculations in-memory. Figure 1-6 illustrates the relative complexity and maturity of a BI solution.

To say that the scope of a reporting or BI solution is relative to the size of a business would be a gross generalization. In some cases, small businesses manage large data volumes, and sometimes big organizations have simple needs. The point is that as your data grows, so does the need to store, manage, and analyze it in the best way.

A BI solution enables business leaders to use the right tools to proactively make informed decisions about their business. Sophisticated reporting and analytics allow information workers and leaders to look beyond the history of their business data. By examining the past and present, you can spot trends and patterns. You can use reliable business analytics to forecast future trends, to plan for improved business processes, and to make informed decisions.

Once upon a time, businesses ran reports to keep track of simple things such as sales totals, invoices, inventory, and production runs. As an industry, we had reached a point where we were quite proficient at gathering and storing data. Most businesses had gigabytes or terabytes of data to report on.

What we (as an industry) were less proficient at was transforming that data back into useful and actionable information. Today, we compete on a global scale. Businesses must be efficient, competitive, and adaptable. Large corporations merge, acquire, outsource, downsize, and realign their strategies more often than ever before. Today's business leaders must be adaptable and prepared to react to industry trends and opportunities in order to thrive.

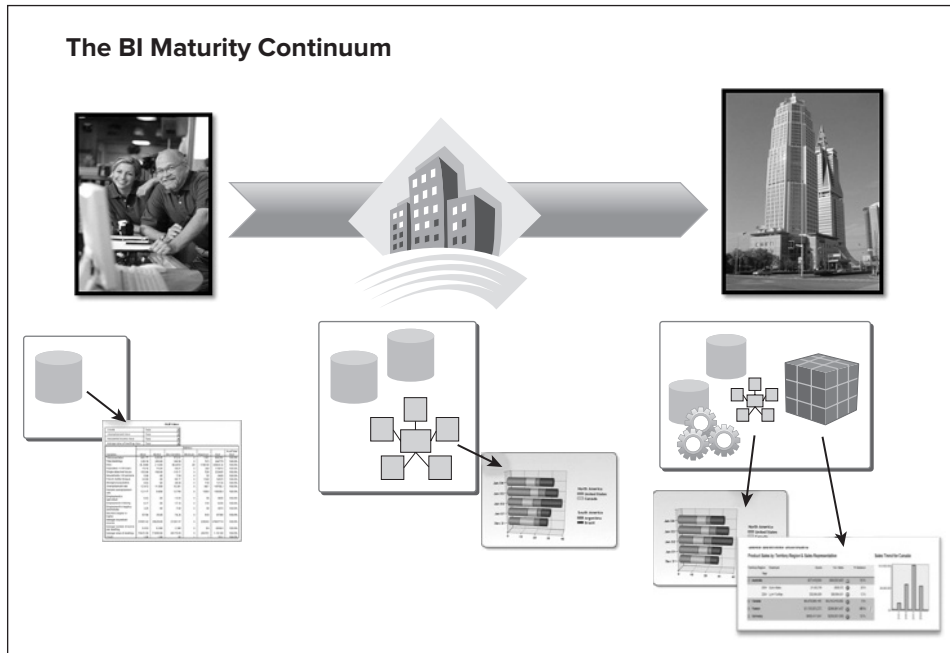


FIGURE 1-6

As a result of this demanding environment, yesterday's static reporting applications have given way to BI solutions. BI is more than the ability to "go get" data. It involves mechanisms that put high-level intelligence in front of leaders in the form of self-service report tools, dashboards, and business scorecards. It proactively alerts users when important events occur and when thresholds are exceeded.

At first, a simple reporting application may use data from a data source or two, but eventually reports may be based on multiple data sources. Sustainable BI solutions are designed around consistent and reliable data sources engineered specifically for reporting. Data is transformed from multiple sources into a central repository using data transformation packages and then may be processed into a semantic model (multidimensional OLAP cube or tabular model). Reports may use a relational data mart, data warehouse, or semantic model. A variety of reports can be created to support business leaders and the decisions they need to make about important business processes. These decision-support reports may take on the form of charts, detail summaries, dynamic drill-down and drill-through reports, dashboards, and business scorecards.

Report Data Sources

Every report has at least one data source and query or reference to the entities that return data values, called a *dataset*. Operational data stores are often the most complex databases. Some packaged systems have databases with thousands of tables. As the dependence on databases and data-driven computer systems increases, most organizations cross a threshold in three areas:

- The complexity of each database grows to accommodate more complex processes.
- The volume of the data increases.
- The number of different databases increases to handle different business data that management needs.

Aside from sheer complexity, it's not uncommon for even midsize companies to store terabytes of data. Storage space is fairly inexpensive when compared with equally capable systems just a few years ago. There may be great value in tracking orders, shipments, calls, cases, and customers, but all this adds up over time. Recording all this activity means that you have a lot of data on hand for reporting. Putting data into a database is the easy part. Getting intelligent, useful information back out — there's the challenge! Finally, different systems are used to manage the same types of data in different ways. For example, a customer relationship management system tracks sales leads and potential customers for a marketing organization differently than an order management system does to support the sales team. In each of these two systems, you may track something called a “customer,” but in these systems, the definition may vary. Perhaps a “customer” may represent a consumer, contact, or company in one system, and a lead, vendor, or reseller in another system. Larger companies may have similar records duplicated across other systems such as enterprise resource planning, human resources management, benefits, vendor management, accounting, and payables and receivables systems.

At some point, most solution designers conclude that to obtain valuable reporting metrics from all these operational data sources, they will have to be consolidated into a central, simplified data store specifically designed to support business reporting requirements. A *data warehouse system* is a central data store used to standardize the data extracted from these complex and specialized data sources. It typically makes use of the same relational database technologies used to house the operational data stores, but it does so in a protected, Read Only environment to keep reporting simple and straightforward.

A data warehouse or data mart is a simplified data structure built using relational database technology, designed specifically for reporting. Reports based on these data sources are easier to design and may perform better than reports designed using transactional, operational data stores. Many businesses need to do more than just list transactional records on reports and add up the totals. Simple data aggregation can be performed with large sets of data from a data warehouse, but deep analysis requires special data storage technology and a more capable mathematical and statistical reporting engine.

Analytic Data Sources and Self-Service BI

Once reports were little more than transaction records printed on paper, called ledgers, journals, and lists. As the need for more useful information arose, so did the sophistication of reporting. Today, reports serve as more than just a method to dump records to the printed page. Users need to

gain insight and knowledge about their business. Dynamic reports allow users to interact and investigate trends in their business environments, rather than just view static lists. As the sophistication of the business users increases, so does the complexity of the data and the reporting medium. The general trend is from a historical perspective to the future. The more accurate and reliable data you have about the past and present, with appropriate reporting models, you can use this to forecast and predict trends and future activities.

As the requirements grow, so does the need for more complex data. In a mid-sized business with moderate data reporting needs and a few data sources, a small data mart or warehouse may serve as a complete BI reporting solution. However, in a larger, sophisticated business environment, a comprehensive data extract, transform, and load (ETL) solution; data warehouse; and business intelligence semantic model may be used to feed the appetite for deeper reporting and analysis. Semantic models open the door to high-level analytics using simple, user-enabling self-service reporting and data-browsing tools.

Similar to Analysis Services multidimensional cubes, tabular models can be used to aggregate large volumes of business data. OLAP cubes and models are also much easier for users to explore and browse using self-service reporting tools. BISM tabular models are based on the same in-memory aggregation technology as Microsoft's PowerPivot add-in for Excel. PowerPivot models can be created with the PowerPivot add-in for Excel and published to SharePoint to be shared and hosted as report data sources. Excel may be used to browse these models using pivot tables and charts. In addition to ad hoc reporting and cube browsing, standard Reporting Services reports may be used to report on cube data using a special query language called *Multidimensional Expressions (MDX)*.

Self-service BI solutions put the power of data analysis in the hands of business users. Enabling effective analysis has required information technology groups, already stressed by resource constraints, to design enterprise BI solutions that require specialized skills and extensive planning. Recent innovations in self-service BI tools such as Microsoft PowerPivot, tabular models, and Power View have bridged this gap. Tabular semantic models can serve up and aggregate large volumes of business data for browsing and reporting, with the added benefit of being completely server-hosted and secure. The tabular model technology actually utilizes the Analysis Services storage engine. The entire model is loaded into the server memory to aggregate and return results very quickly.

Tabular and multidimensional semantic models each offer unique strengths for efficient analytic reporting, and Reporting Services may use any of these options as report data sources.

Complexity and Report Performance

System performance is often one of the most significant drivers of an effective BI solution. As an organization's reporting needs become more sophisticated and the data's complexity and volume increase, the cost is usually measured first in performance. Queries take longer to run and compete for resources on the report and database servers. In this case, IT professionals typically react by recognizing the value of and need for a simplified database. Whether this is to be a truly enterprise-ready data warehouse, a departmental data mart, or a simple "reporting structure," the basic concept is usually the same — simplify the database design to focus on reporting requirements.

As mentioned, some performance and advanced analytical requirements may also drive the solution's maturity to include OLAP cubes. This doesn't necessarily mean that all the reports designed

against other data sources must be updated. A variety of reports may work just fine with an operational data source or relational data warehouse. But other, more sophisticated reports require specialized data sources (such as OLAP cubes) to perform well.

I recall a consulting assignment in which I developed reports with complex financial formulas using the original database structure as the report data source. The T-SQL queries were complex and difficult to debug. The client was thrilled when one of the more complicated reports took only 45 minutes to run instead of the 90 minutes it took before we “optimized” the query. After transforming the same data into a simplified data mart structure, it took less than 3 minutes to run the same report. With an OLAP cube in Analysis Services, the same report ran in just a few seconds. Needless to say, the “acceptable” 45-minute report rendering time was no longer acceptable after the users found out that they could run the same report in a few seconds!

One important consideration in this equation is that the cost of a data-warehouse design or an OLAP database can be low compared to poor report performance and unnecessarily long report development cycles. SQL Server Analysis Services is an impressive and compelling technology with tremendous value for even small and midsized businesses. If you have not explored this option and you need to do reporting beyond the basics, we urge you to take a serious look at this impressive tool set so that you can appreciate its value.

CUSTOMIZING THE REPORTING EXPERIENCE

Reports may be delivered in a variety of ways (not just when a user navigates to a report in real time). Reports may be automatically rendered to the server cache so that they open quickly and don’t burden data sources. They may be delivered via e-mail and to file shares on a regular schedule. Using data-driven subscriptions, reports may be “broadcast” to a large audience during off-hours. Each user may receive a copy of the report rendered in a different format or with data filtered differently. You will learn to plan for, manage, and configure these features.

You will learn how to optimize, back up, and recover the Report Server database, web service, and Windows service. You also will learn how to use the management utilities, configuration files, and logs to customize the server environment and prevent and diagnose problems.

SUMMARY

This chapter provided a context for the rest of the book and introduced the concepts, skills, and capabilities you will learn about in subsequent chapters.

The chapters in Part I give you the direction you need to get started with Reporting Services by installing the tools and configuring them in your environment. You will learn how to set up Reporting Services in native mode and integrated with SharePoint 2010.

The five chapters in Part II take you, step-by-step, from a novice report designer to a report rock star. The discussion of basic report design has you use wizards and simple tools to create typical business reports such as tables, a matrix, lists, and simple charts. You’ll learn how to lay out and format each report using recommended techniques. You’ll design queries for SQL Server and other

data sources, and learn the basics of T-SQL to create filtered and parameterized datasets. The discussion of advanced report design covers the more compelling features and creative ways to make reports come to life with dynamic formatting and actions. We've dedicated a full chapter to chart report design to cover more of the useful and advanced visual chart capabilities.

Parts III and IV cover many of the new BI innovations introduced in SQL Server 2012 and SharePoint 2010: semantic tabular models and the Power View visual experience. These parts also cover reporting on multidimensional cubes and MDX queries. Whether you are using Reporting Services with SharePoint or in native mode, you will learn to use Report Builder and a self-service reporting tool to empower business users to answer important business questions using easy-to-use browsing and design tools.

The SQL Server BI and reporting platform gives you many tools and options that can be a bit overwhelming. Part V provides you with prescriptive guidance that helps you benefit from our many years of solution design work and successful reporting projects. We have outlined a number of best practices for report solution architectures, designs, and custom programming techniques.

Report server administration and content management are covered at length in Part VI. You will see how to integrate Reporting Services with SharePoint and manage a native report server, as well as monitor and optimize for reliable, secure, and optimal performance.

Part VII is for the programmer who wants to integrate reports into a custom solution and for the advanced report designer who can use custom program code within report design. You will learn how to extend the capabilities of reports with expressions and custom functions and extend applications by adding reporting capabilities.

