

PART I

Getting Started

What exactly is SQL Server Reporting Services? How is it used and what are its capabilities and boundaries? Is it a product, a part of SQL Server, or a development platform? The three chapters in Part 1 will get you started with understanding the capabilities of Reporting Services at a high level. You will become acquainted with the entire SSRS platform, the components it encompasses, and their capabilities.

You'll learn about the new features introduced in SQL Server 2016: the new web portal, key performance indicators, and mobile reports. Chapter 2 introduces several key integrations with the Microsoft business analytics platform and advanced visualizations. You'll also see how to install and configure Reporting Services tools and the server so you can get up and running.

- **CHAPTER 1: Introducing Reporting Services**
- **CHAPTER 2: What's New in SQL Server 2016 Reporting Services**
- **CHAPTER 3: Reporting Services Installation and Architecture**

1

Introducing Reporting Services

WHAT'S IN THIS CHAPTER?

- Identifying who uses Reporting Services
- Using dashboards, reports, and applications
- Understanding application integration
- Using Business Intelligence (BI) reporting
- Using mobile reports and KPIs
- Choosing a report tool
- Optimizing report performance

Welcome to SQL Server 2016 Reporting Services. This chapter provides an overview that includes a high-level introduction featuring not only concepts and capabilities of this powerful reporting tool, but also of the Microsoft data analysis platform. Reporting Services embodies a rich history as a rock-solid reporting tool. Although many features have been part of the product for more than 12 years, some features are new, have changed, or were introduced in later versions.

This is the fifth edition of this book. Reporting Services was officially released in early 2004. Since that time, I gained assistance from trusted and experienced colleagues who contributed to previous book editions, and this edition draws upon that foundation of expertise. In areas where the product has matured and evolved forward, I share advanced capabilities and patterns for solving new business problems. The book includes material and techniques using the new or existing features more effectively.

As a Microsoft Data Platform MVP, a specialist, and a respected contractor for Microsoft, I spend considerable time working with different organizations to design reporting solutions. For many years, I frequently have had the opportunity to work alongside the Reporting Services product team. Through leadership changes, product development cycles, and industry

trends, the development team has maintained a relevant and durable reporting product that focuses on the needs of the modern business. As you continue to read, you will learn to appreciate the depth of this product.

In 2003, a few months before the product was released, I started using pre-release versions of Reporting Services. At the time, I was doing web development and database work, and found Reporting Services to be a perfect fit for the reports I needed to add to a web application. Since then, SQL Server Reporting Services (SSRS) has grown to become the de facto industry standard reporting tool. SSRS provides a foundation upon which you can construct complete report, scorecard, dashboard, and mobile 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. In 2016, the product expanded beyond classic “paginated reports” to add mobile reporting, key performance indicators (KPIs), and integration with cloud-based and on-premises dashboard and self-service analytic tools.

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. My team was assigned to evaluate every major reporting product and give them an unbiased analysis. We worked with the client to identify about 50 points of evaluation criteria. Then I contacted the major vendors, installed evaluation copies, explored features, and spoke with other customers and with those who specialized in using these various products. It really helped the team see the industry from a broad perspective, and resulted in 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.

WHO USES REPORTING SERVICES?

The various titles given to someone who creates reports in different organizations is an interesting topic. An observation I have made over the years and in different work environments is the perception of this role. In some places, people who write reports are called *report developers*. In some environments, application developers assign the name *report users* to people creating reports.

Business users fit into a few categories when you consider how they use reports. Some are *report consumers* only. They’re content to use reports that have been written and published for them. Others prefer to create their own reports without becoming mired in the intricacies of programming code and complex database queries. Maybe they just want to browse information to look for trends and to understand how the business is measuring up against their goals. In recent years, a new generation of *data consumers* has changed the landscape of self-service reporting and business data analytics. These are the data scientists and the data analysts who collect, wrangle, sculpt, model, and explore data using analytic reporting tools like Power BI and advanced add-ins for Excel.

Traditional roles have changed. New reporting and analytic tools have matured to accommodate the business climate. Not long ago, a typical IT group at most large organizations had three common roles: system administrators, application developers, and project managers. Where does the report designer fit in the organization? People who design business reports often 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 hard-core computer geeks.

If you're a business-type person, you probably don't care about integrating your reports into custom applications and websites, or about writing complex programming logic. Some of us live for that. What you may care about is giving your savvy business users the capability to easily visualize important key metrics to see what products are performing well in their sales territories. Maybe you want to enable business leaders to access important metrics and performance indicators on mobile devices.

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 roles in their organization. The following roles represent the majority and describe some of the trends noted as the industry continues to change.

Information Workers and Data Analysts

People in this role have strong computer skills, but do not 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 fastest-growing group of report tool users in the industry.

A rapidly growing subset of the IW community is the *self-service analyst*. This persona not only has an aptitude for working with data and analytic tools, but also understands a particular field of business and what the data means in business terms. This subset may have specialized skills in a discipline of science or statistics. Data analysts usually have a mind for numbers and perhaps an artistic propensity for graphical presentation and storytelling with data.

An interesting transition has occurred in more progressive business environments, but has not yet happened in more traditional places. Whereas Excel has long been the primary data analysis tool, forward-thinking business data analysts are adopting tools like Power Pivot, Power BI, and Tableau Desktop to curate, deeply analyze, and visualize data to dredge out insights and valuable opportunities in order to take action. The only thing they need from IT is to give them access to reliable data so they can analyze it themselves. This new generation of analysts insists that they have access to data and permission to use their own tools to manage their areas of the business. The former generation of leadership insists that the IT and report "developers" export their spreadsheet-like reports into Excel so that the data can be manipulated and spreadsheet functions (full of calculation formulas that reference worksheets, and that do more calculations) can be performed to make it all line up and balance.

Spreadsheets and spreadsheet-style reports are the heart and soul of many financial organizations for good reason. However, some people experience a world outside the traditional, two-dimensional view, and break the routine to see things a different way.

Report Builder was designed for advanced report users and business-centric report designers. The capabilities are nearly identical to SQL Server Data Tools (SSDT) for Visual Studio, but it is simple and streamlined for the advanced user, rather than the developer.

Information Consumers

In the traditional user role of a user who runs or receives reports (perhaps through a portal or by e-mail), *information consumers* simply view information. Individuals within this group may be occasional report users and business workers, or consumers who use reports to perform a specific task, rather than interacting with data.

This role will always exist. But just as people are becoming more experienced and proficient with analytic tools, many consumers are also becoming occasional IWs and business data analysts.

TIP *A common temptation for the experienced report developer (present company included) is to try to convince users that they need advanced report features. Be cautious about selling users on “cool” tricks and capabilities they may not need in their reports.*

In light of many different reporting scenarios, it is important to acknowledge and serve the needs of users who simply need to run or print reports. Making that experience as convenient and trouble-free as possible can make a huge impact on streamlining a business.

Business Managers and Leaders

If you are a *business manager*, you are interested primarily 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 IWs, they have little interest in the implementation details or technology used to make it work. As IWs, managers may create their own reports to analyze the productivity of their team or area of responsibility.

Managers tend to view reporting from one of two different perspectives:

- They may need (or prefer to use) operational reports prepared for them that they can simply run and view results in a static, predictable format.
- They may extract data to a format suitable for analysis and manipulation (such as into an Excel workbook, or a tool they can use to further analyze and visualize results).

Mobile reporting solutions enable line managers, traveling business leaders, and IWs to access information on touch-optimized mobile devices. These reports are best suited for scenarios with key metrics and aggregated results, rather than multi-page, detail reports.

Software Developers

To achieve advanced reporting features, *software developers* write 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 is 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. Reporting Services offers several opportunities to integrate reports into custom-developed software solutions.

Developers and serious report designers will typically prefer to use developer-centric tools for report design such as SSDT for Visual Studio.

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 be created to help monitor system usage and maintenance statistics to make a system administrator's job easier.

DASHBOARDS, REPORTS, AND APPLICATIONS

What exactly is the difference between a report, a dashboard, and a scorecard? It depends on a few factors, but there is some overlap between these concepts.

Quite a few years ago, a shift developed from client-based processing toward applications that ran on servers. Web technologies have proven to be an effective way to make systems available to a large number of people. Like a web application, browser-based reports do not always offer the same tactile and responsive user experience as a client application. Now, with the advent of smart mobile devices and applications, the climate has shifted once again from client, to server, and then to a balance of client/server technologies that support both connected and disconnected user experiences.

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 SSRS reports are used today. However, the capabilities do not stop there. Reporting Services lets you run reports in a variety of modes and applications. If we have learned anything from decades of computer system evolution, it is 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. Having the capability to operate in a disconnected mode offers tremendous advantages over purely server-side, connected systems.

Application Integration

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. Many intranet sites run on web portals, rather than custom-programmed websites, and Reporting Services naturally plays well in practically any web portal environment.

If you are a 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 have 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.

Not to single out the programmers (who are generally a pretty smart bunch), the same concept applies to practitioners of any single discipline. The point is that different tools and technologies solve different problems, and sometimes it's important to look outside of one's discipline to gain a fresh perspective and ensure that you're using the right tool for the job.

User Interaction

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 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. 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.

Rather than cramming all the details into one report, a summary report or dashboard presents high-level information and key metrics. As shown in Figure 1-1, users can click a chart or summary value to navigate to a detail report in the context of the selected item, revealing more details and relevant facts. Using report navigation actions, drill-down, and other interactive features, reports may be orchestrated into complete solutions that enable a data exploration experience.

Techniques can 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, inline frame (`<iframe>` tag), or `ReportViewer` web control
- Programmatically writing reports to files available for download from a website
- Using a web part to embed reports into a SharePoint Web portal
- Fully integrating the report server in SharePoint Integration mode

Ch12 - Products by Category and SubCategory (Drillthrough source)

Category	Subcategory	Product	Standard Cost	Color
Bikes	Mountain Bikes	Mountain-100 Black, 38	\$1,898.09	Black
		Mountain-100 Black, 42	\$1,898.09	Black
		Mountain-100 Black, 44	\$1,898.09	Black
		Mountain-100 Black, 48	\$1,898.09	Black
		Mountain-100 Silver, 38	\$1,912.15	Silver
		Mountain-100 Silver, 42	\$1,912.15	Silver
		Mountain-100 Silver, 44	\$1,912.15	Silver
		Mountain-100 Silver, 48	\$1,912.15	Silver
		Mountain-200 Black, 38	\$1,105.81	Black
		Mountain-200 Black, 42	\$1,251.98	Black
		Mountain-200 Black, 44	\$1,105.81	Black
		Mountain-200 Black, 46	\$1,251.98	Black
		Mountain-200 Black, 48	\$1,105.81	Black
		Mountain-200 Silver, 38	\$1,117.86	Silver
		Mountain-200 Silver, 42	\$1,265.62	Silver
		Mountain-200 Silver, 44	\$1,117.86	Silver
		Mountain-200 Silver, 48	\$1,265.62	Silver

Product Details

Mountain-100 Silver, 48

List Price: \$3,399.99 Profit: \$1,487.84

Std Cost: \$1,912.15 Prod #: BK-M82S-48

FIGURE 1-1: Using navigation links.

NOTE When compared with custom-developed solutions, Reporting Services provides useful functionality and business value for relatively little investment. Be mindful that Reporting Services provides the means to extend its capabilities through custom programming, but the cost (in terms of time and effort) may be considerably higher, and, in some cases, may be more restrictive than using custom programming components.

Numerous creative ways exist to integrate reports into a web or desktop application. These techniques range from simple (for example, requiring a little HTML script) to complex, custom methods. And, if it is not enough to be able to embed reports into custom web pages, it is possible to use custom program code to embed additional content into reports.

The Reporting Services ReportViewer control can be used to view server-based reports in a form. These reports are 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, meaning that reports built into a custom application can run independently from the report server.

SharePoint Integration

Reporting Services has native integration with Microsoft SharePoint Server and it works quite well. SharePoint is an abundant platform for document collaboration, as well as for managing document workflows and approval processes. At the same time, it is complex to administer and manage.

I have learned some valuable lessons about using Reporting Services with SharePoint. If you had asked me eight years ago about whether to include SharePoint in your reporting and business intelligence (BI) platform, I would have likely echoed Microsoft's recommendation to use SharePoint as the backbone for most solutions. Today, I am more cautious with my recommendations and ask more questions. SharePoint can be expensive, as well as complicated to set up and support. It adds processing overhead, which can affect performance and hardware requirements.

If your organization has invested in SharePoint on-premises and you are enjoying business value from the many services and capabilities that the platform offers, adding Reporting Services may be a natural fit for you. I have worked with several large organizations to fold SSRS, along with SQL Server Analysis Services (SSAS), Power Pivot, and Office into their SharePoint platform to build integrated business reporting and analytic solutions with great benefit.

Business Intelligence and Analytics Solutions

Once upon a time, reports were little more than transaction records printed on paper, also 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 a method to dump data 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 transaction lists. It is important to realize that, as the sophistication of the business user grows, the complexity of the data and the reporting medium also increases. Sophisticated analytics uses a historical perspective to look into the future. Using accurate and reliable data from the past and present, as well as appropriate reporting models, allows analysts to forecast and predict trends and future activities.

A 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 not effectively 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. Most BI solutions integrate data from multiple sources to measure business success and trends. Consequently, this often requires a data warehouse, data mart, and/or semantic data model, as well as data extract, transform, and load (ETL) processes. Recent enhancements to the SQL Server database engine (such as in-memory column store indexes) may improve performance without radical database redesign.

Using modern analytic modeling tools, smaller-scale BI solutions can be created with a comparatively moderate investment. Complex analysis solutions often require tabular or multidimensional data structures created with SSAS. Microsoft developed the SSAS multidimensional database

technology, often called *online analytical processing (OLAP)*. This uses cubes and dimensions storing data in a pre-grouped and pre-aggregated format on disk. The data is quickly available for reporting and browsing.

In SQL Server 2012, Microsoft released a “tabular,” in-memory implementation of Analysis Services that has matured significantly in 2016. Tabular and multidimensional semantic models each offer unique strengths for efficient analytic reporting. In many cases, tabular models are easier to design, more efficient, and faster for reporting and analysis. But multidimensional SSAS includes complex and mature features. Both flavors of Analysis Services can be queried from SSRS reports using the MDX query language.

Past editions of Microsoft platform tools required an investment in SharePoint Server to fully implement BI solutions. SharePoint (either online or on-premises) still serves an important purpose today, but it is not a requirement to do BI right.

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 IWs 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 make informed decisions.

Yesterday’s static reporting applications have given way to BI solutions. BI is more than the capability 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. Data may then be processed into a semantic model (multidimensional cube, or tabular model). Reports may use a relational data warehouse, data mart, or semantic model. A variety of reports can be created to support business leaders and the important decisions and processes. These decision-support reports may take on many different forms, such as charts, detail summaries, dynamic drill-down and drill-through reports, dashboards, and business scorecards.

Mobile Reports and KPIs

Mobile device reporting is a completely different paradigm from traditional desktop reporting, with the goal being to present important information in a simple and touch-friendly medium.

All report types, paginated reports, and KPIs are managed and accessed through a new web portal. The web portal (shown in Figure 1-2) can be accessed in a web browser and on mobile devices using the Power BI mobile app.

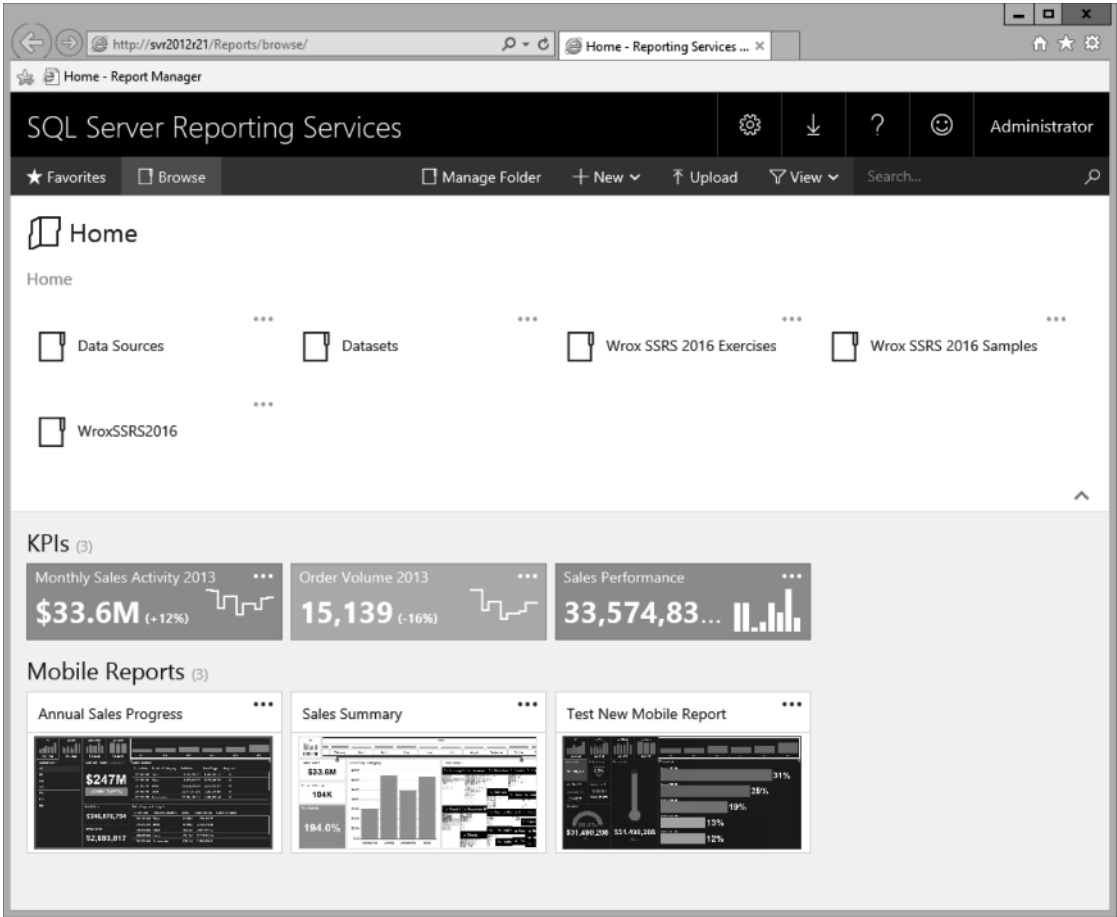


FIGURE 1-2: Web portal.

Figure 1-3 shows the Web portal in the Power BI app for the iPhone.

The mobile reporting addition to SQL Server 2016 Reporting Services is a new and unique capability. It is different for a couple of reasons. Using a fresh perspective, the feature was developed by another organization and was designed to provide a different user experience than conventional SSRS reports. Mobile reports are simple with focus on the user experience. They can be easy to design, but require some data and query preparation.

Mobile reports are designed using the Mobile Report Publisher, a standalone tool connecting results from predefined datasets. Reports are styled using a color palette theme, and individual layouts are applied to a report for desktop, tablet, and phone devices. After a report is published to the report server, alongside other Reporting Services content, users can connect with their mobile devices.

Using a freely available app installed from the device provider's application store, users interact with offline data. The offline report cache is synchronized on-demand, or at scheduled intervals using shared datasets. These are part of the standard Reporting Services server architecture.

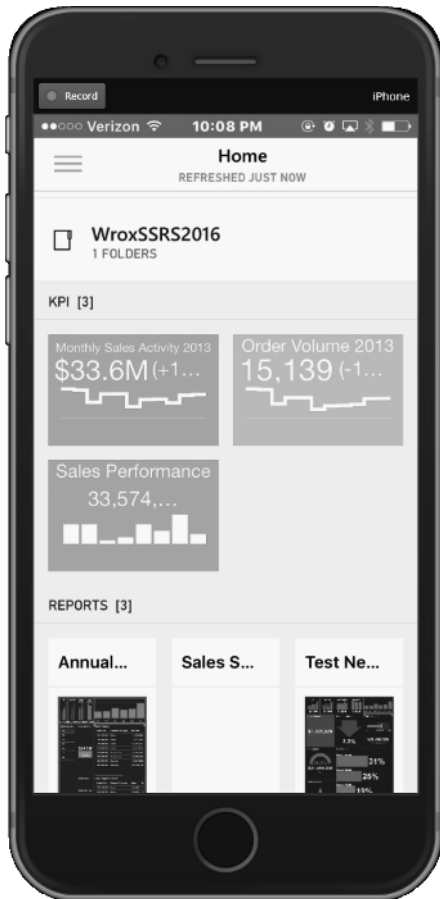


FIGURE 1-3: Web portal on iPhone.

Mobile reports are optimized for use on mobile devices using the Power BI mobile app for each mobile device platform and form factor. The appropriate app for a user's device is downloaded and installed from the Windows, Android, or Apple App Store. Figure 1-4 shows a mobile report optimized for Portrait layout in the iPad app.

KPIs are a standard feature of the new web portal. They also get data through SSRS shared datasets. After datasets are prepared, KPI design is very simple and performed through a web interface. Key metrics are visualized using color, text, and bold graphics to indicate metric status comparing it to a target or goal, as well as trends using simple sparklines and chart visuals.

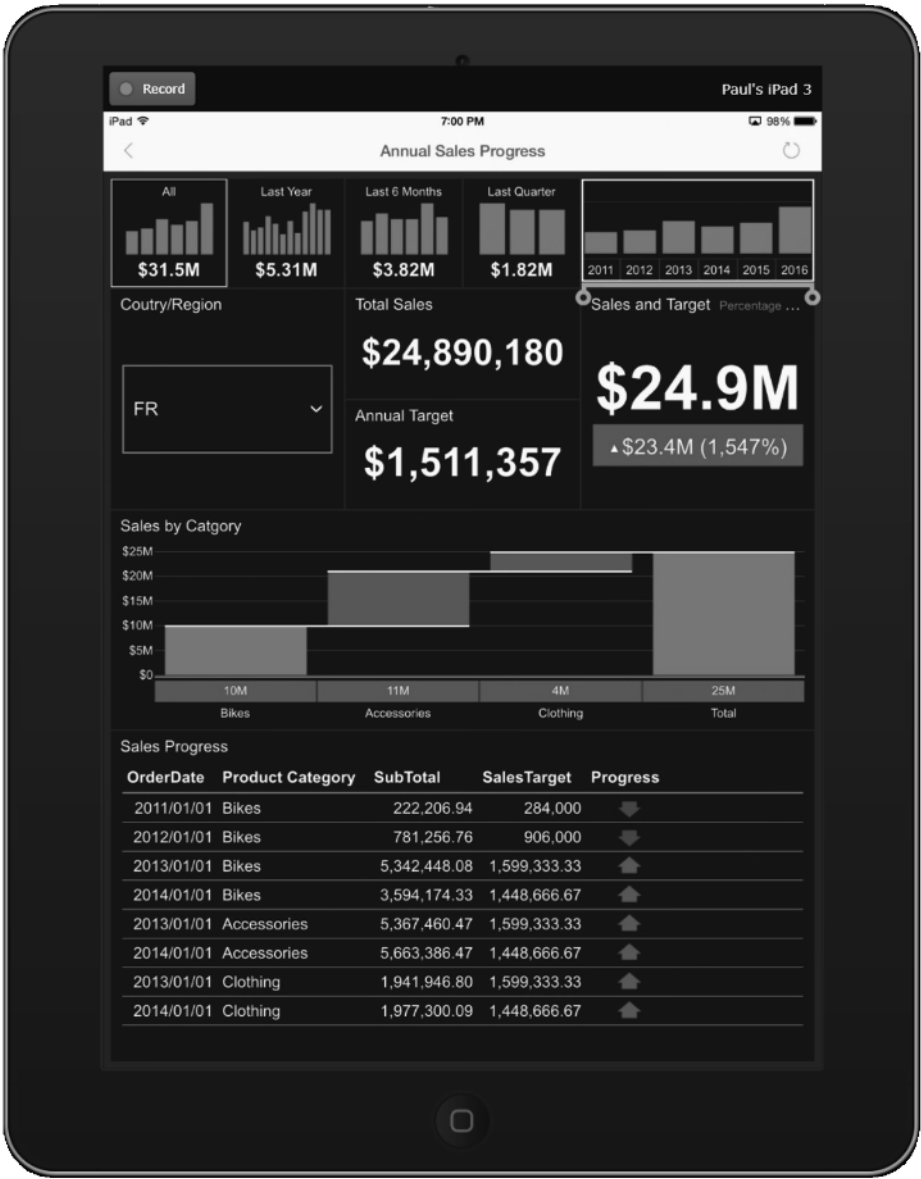


FIGURE 1-4: Mobile report on iPad.

REPORT TOOL CHOICES

The universe expands. Software vendors add more applications to their collections much faster than they sunset the old ones. Likewise, Microsoft continues to add applications and features without distinct use case boundaries between them. As a result, if there were two different options before,

now there are three or four—and it is up to us to decide which choice is best for our needs. Like it or not, this is the nature of the technology-saturated world in which we live and work. I spend much of my time giving advice about the pros and cons—advantages and feature gaps—between different reporting tools. Throughout this book, this topic is addressed, best practices are called out, and proven design patterns are described as learned from various projects and field experience.

The majority of new Reporting Services implementations for most organizations use the de facto web-based web portal 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 Report Builder to produce a simple 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 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 do not require users to know a lot about complicated things like programming, writing queries, and building expressions.

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.

IT-Designed Reports

Reporting Services was first designed and optimized for programmers and application developers who were accustomed to using Visual Studio. The report project design add-in for Visual Studio, originally called Business Intelligence Development Studio (BIDS), is now called SQL Server Data Tools (SSDT). Advanced capabilities are accessible using a variety of tools familiar to application developers. Like other Visual Studio solutions, report definition files can be managed as a single deployment unit to publish reports and related objects to the appropriate folders on a report server.

Likewise, in application development projects, reports, data sources, shared datasets, and all other design elements can be managed with integrated version control in the SSDT environment. Developers can use Microsoft Team Foundation Server, GitHub, or other source code management systems to collaborate as a team and recover from file loss.

User-Designed Reports

The industry’s quest to create the perfect easy-to-use BI tool has produced many different products, each with its own unique capabilities. Under the Reporting Services umbrella, two self-service reporting tools serve different needs. The current incarnation of Report Builder is based on the mature report definition architecture. Report Builder reports can span the spectrum from simple to complex, with many design options.

Report Builder creates reports that are entirely cross-compatible with SSDT, and that can be enhanced with advanced features. Incremental product improvements over the past few versions have made out-of-the-box report design even easier. 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.

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

TABLE 1-1: Report Designer and Visualization Options

REPORT DESIGNER	BACKGROUND
SQL Server Data Tools (SSDT)	This implementation of the Visual Studio shell is typically used by IT professionals to design reports with a project and team focus. It currently uses the Visual Studio 2010 shell.
Report Builder	Successor to previous tools introduced in 2005 and 2008, Report Builder has had incremental improvements in 2012, 2014, and 2016.

REPORT DESIGNER	BACKGROUND
Mobile Report Publisher	This is a new addition to the 2016 product. Mobile reports are designed separately from paginated reports and deployed to the common report server. Reports may be viewed on most any mobile device (phone or tablet) using the Power BI mobile app or Datazen mobile app. These reports can also be viewed in the web browser through the web portal.
Web portal KPI Designer	KPI tiles (with sparkline trend and thumbnail comparison charts) are designed in a web interface using the web portal. Data for each KPI element uses dataset queries stored in report server folders.

Server-Based Reports

Reports can run on either a report server, or in a standalone 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.

NOTE *Local Mode reports use a special report definition file with an RDLC extension. These reports run within a Windows or Web form control that is deployed with the hosting application. Some programming code is necessary and they are typically best used for low data volume applications.*

It is important to understand the difference between SQL Server Reporting Services and a desktop reporting tool such as Microsoft Access. Reporting Services is not 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 and typically runs on a dedicated server. Likewise, reports may be integrated into SharePoint Services to be managed, secured, and administered alongside other shared corporate documents and assets. At the same time, Reporting Services can be used in a simple standalone deployment with relatively little administrative overhead.

Reporting Services is scalable and adaptable for use by a handful, as well as thousands, of users, for reporting on large sets of data stored in a variety of database platforms. Just because Reporting Services is a business-sized product does not mean that reports need to be complicated or difficult to design.

Report users need to be connected to a network, or perhaps to 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, PowerPoint, and Adobe PDF, or as a PNG, JPEG, GIF, or TIFF image. Reports can be saved to files in these and other formats for offline viewing. Reports can also be scheduled for automatic delivery by the report server by e-mail, or can be saved to files. These features are standard and require only simple configuration settings and minor user interaction.

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.

NOTE *I have used Reporting Services to connect to many different data sources, including products outside the Microsoft product portfolio. Although SSRS can connect effectively to sources like Oracle, Teradata, IBM DB2, SyBase, MySQL, PostgreSQL, XML files, and SharePoint lists, sometimes it is easier to transform data into SQL Server so that you can connect trouble-free. The optimal choice will depend primarily on the complexity of the data.*

Aside from sheer complexity, it is not uncommon for midsized companies to store terabytes of data. Storage space is fairly inexpensive when compared with equally capable systems a few years ago. There may be great value in tracking orders, shipments, calls, cases, and customers, but all this data adds up over time. Recording all this activity means 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—now there is 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.

Modest data aggregation can be performed on large sets of data from a data warehouse. In contrast, deep analysis requires special data storage technology, as well as a more capable mathematical and statistical reporting engine.

Enterprise Scale

Delivering reports to many users requires a scalable reporting environment. Reporting Services processes queries and then renders reports on the report server. Because it uses industry-standard 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, many users can run reports at the same time while consuming minimal server resources. To serve more users, report servers may be scaled out using load balancing and distributed server farms.

The Reporting Services report server exposes its functionality in the same way that a standard ASP.NET website is hosted for users. Reports can 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 and accessed through the web portal web interface installed with Reporting Services. Reports can also be exposed in custom-developed web applications using practically any set of web technologies or development tools.

OPTIMIZING PERFORMANCE

Often, system performance is 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 drive the solution's maturity to include OLAP cubes. This does not necessarily mean 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.

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. Throughout this book, you learn to plan for, manage, and configure these features.

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

Performance

While on a consulting assignment, I developed complex financial formula reports 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 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! Although this makes for a good story, the fact is that today people expect results quickly.

Users typically have little concern for the complexity of a database solution or the technology used to deliver data. They simply need results fast, and that's usually what they expect. The task is left for us to architect solutions to deliver results and perform calculations and metrics from large volumes of business data. Optimal performance is achievable using several innovative features of the SQL Server reporting architecture. Examples include in-memory storage and column store indexing, in-memory tabular and multidimensional semantic modeling, report instance caching, and report page-level rendering. Mobile reports can also use client-side data cache to optimize report performance and provide off-line viewing.

SUMMARY

Your role as a report designer or solution developer will determine how deep you need to immerse yourself into the complexities of Reporting Services. The needs of report users vary from simple to complex, and the time and energy you invest could vary from hours to months, depending on the solution scope.

Some users need to simply run or print reports. Others need or want to be more self-sufficient—either designing reports by themselves, or using self-service tools to perform design and data exploration.

Business Intelligence (BI) reporting solutions include dashboards, scorecards, KPIs, and interactive mobile reports that enable business information workers and business leaders to get insights from data. These solutions often use data modeling technologies like SSAS with visual reports and BI tools. Mobile reports allow users to interact with business data on their mobile devices, tablets, and smartphones on every device platform. Comparatively, mobile reports allow users to operate with disconnected data on touch-enabled devices.

Reports are integrated into applications and custom solutions using web service components, page frames, and form controls. Reporting Services integrates with applications and enterprise solutions using a variety of options. The spectrum of integration options is vast. Your solutions may be very simple using “out of the box” features or tightly integrated with SharePoint, custom applications, Power BI, and the entire Microsoft reporting ecosystem.

If you are new to SSRS, start small and learn the platform. With a little experience, you will figure out which features to use to meet your business and user needs. If you have been using earlier versions SSRS for a while, I will show you how the product has grown and demonstrate new patterns in a best practice.

Chapter 2 introduces you to the new web portal for report navigation and management. You’ll learn about several significant report rendering enhancements and modernized features.

