

Microsoft® SQL Server® 2012

REPORTING SERVICES

Fourth Edition

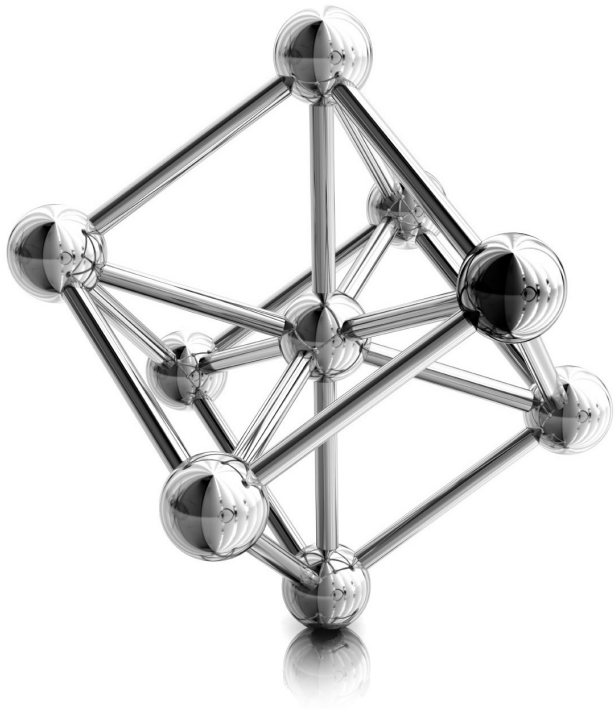


BRIAN LARSON

Foreword by Thierry D'Hers
Principal Group Program Manager for Microsoft SQL Server Reporting Services
Microsoft Corporation

Part I

Getting Started

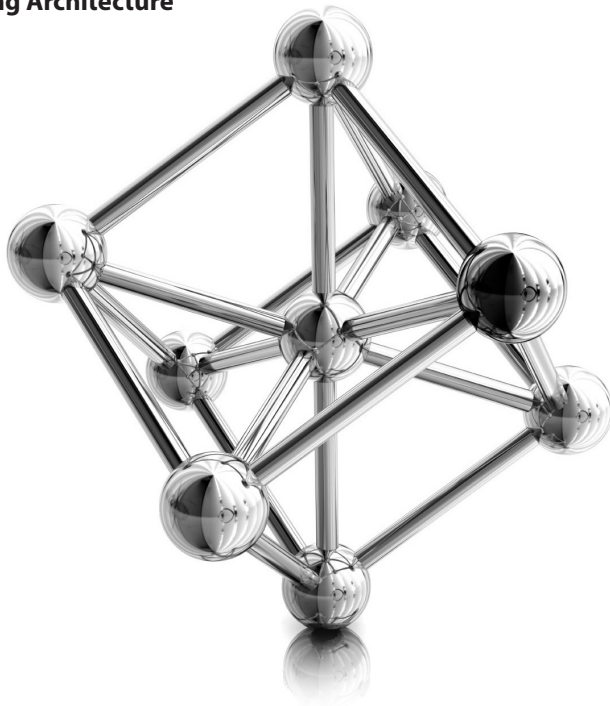


Chapter 1

Let's Start at the Very Beginning

In This Chapter

- ▶ **Sharing Business Intelligence**
- ▶ **Report Authoring Architecture**
- ▶ **Report-Serving Architecture**
- ▶ **Diving In**



SQL Server 2000 Reporting Services was Microsoft's entry into the web-based reporting arena. This first version of Reporting Services enabled you to easily share business information—what is commonly known as “business intelligence” these days—with management, co-workers, business partners, and customers throughout the world. In an interconnected workplace, it makes sense that your reporting solution should offer company-wide, nationwide, and even worldwide communication.

SQL Server 2005 Reporting Services built on the success of the original. Where almost every other aspect of the SQL Server 2005 release represented a completely new platform, Reporting Services simply added to the solid foundation provided by the earlier version to make a great product even better. The 2005 release provided an additional report-authoring environment, improved report-development features, and enhanced capabilities for distributing reports.

The SQL Server 2008 release brought major changes to Reporting Services. The report processing and rendering engine was completely rewritten. This allowed Reporting Services to be more robust, especially when rendering large reports. This overhaul allowed for major changes to the report definition structure and a simplification of the inner workings of the report-rendering engine.

The interim SQL Server release, SQL Server 2008 R2, which came out in 2010, added more data visualizations and report authoring enhancements. Data bars, sparklines, indicators, and maps were added to the array of tools available for presenting information on reports in a graphical format. Individual report items could be saved in a report part gallery for reuse by other report authors.

All that brings us to the latest release of Reporting Services, delivered as part of SQL Server 2012. A release that, once again, brings exciting new features to the product. But before we dive in to the latest release, let's take a quick look back at Reporting Services' beginnings.

Reporting Services was code-named Rosetta during its original development at Microsoft. This name comes from the Rosetta Stone, a stone slab found in 1799 that contains an inscription in both Egyptian hieroglyphics and Greek. This stone provided the key piece of information necessary to unlock the mystery of Egyptian hieroglyphics for the modern world. Just as the Rosetta Stone brought key information across 1,400 years of history, Rosetta, or Reporting Services, is designed to bring key information across distances to unlock the mystery of success for your business.

The Rosetta project, which I was privileged to play a very, very small part in, was originally conceived as a feature of SQL Server 2005. However, as Microsoft told prospective customers about the features in Rosetta and demonstrated the first alpha versions, the reaction was strong: “We need this product and we need it now!” Because of this reaction, Microsoft decided that Rosetta would not wait for 2005, but, instead, would be made its own product to work with SQL Server 2000.

Just what are the features of Reporting Services that got everyone so excited? *Reporting Services* provides an environment for creating a number of different types of reports from a number of different data sources. The reports are previewed and refined using one of several authoring tools. Once completed, the reports are deployed to a *report server*, which makes the reports available via the Internet in a structured, secure environment. Last, but not least, the report management and distribution portion of Reporting Services is free of charge when installed on a server already running SQL Server.

Why did this set of features generate so much excitement? When you put them all together, the result is a product that facilitates the creation, management, and timely use of business intelligence.

Sharing Business Intelligence

Because you are reading this book, you are probably the keeper of some type of information that is important to your organization. You may have information on sales, finance, production, delivery, or one of a hundred other areas. All this information makes up the business intelligence necessary to keep today's corporate, academic, nonprofit, and governmental entities humming along.

The Need to Share

In addition to maintaining this information, you have a need to share this information with others. This need to share may have come from an important lesson you learned in kindergarten ("The world would be a much happier place if we all learned to share") or, more likely, this need to share your information was probably suggested to you by a manager or executive somewhere higher up the food chain. See if any of these situations sound familiar.

The Production Manager

Your company's order-entry system automatically updates the inventory database every four hours. In your company's line of business, some orders can require a large quantity of a given product. Because of this, it is important that the production manager knows about these changes in the inventory level in a timely manner so he can adjust production accordingly.

The production manager has asked you to provide him with an up-to-date inventory report that is created immediately following each update to the inventory database occurring during business hours. He would like this report to arrive on his PC as quickly as possible so he can make changes to the production schedule within an hour of the updates. He would also like to be able to print this report so he can add his own notations to it as he works out his new production schedule.

One more fact to keep in mind: your company's inventory system is in Cleveland, but the production facility is in Portland!

The Vice President of Sales

You are responsible for maintaining information on the amount of credit your company will extend to each of its clients. This information is updated daily in the company database. A report containing the credit information for all clients is printed weekly at corporate headquarters and mailed to each sales representative.

The vice president of sales has requested that the credit information be made available to the sales staff in a more timely manner. He has asked that this report be accessible over the Internet from anywhere across the country. The sales representatives will print the report when they have access to the Internet, and then carry it with them for those times when they cannot get online. He has also asked that this online version of the report be as up-to-date as possible.

The Chief Executive Officer

The chief executive officer (CEO) for your company has a hands-on management style. She likes to participate in all facets of the decision-making process and, therefore, needs to stay well informed on all aspects of the company. This includes the corporate balance sheet, inventory and production, and the company's stock price.

The CEO expects all this information to be available on her desktop when she arrives for work each morning at 7:00 A.M. The information must be in a format that's appropriate to print and share with the corporate vice presidents at their meeting each morning at 9:00 A.M. As you search for solutions to this one, remember no budget is allocated for this project—and, of course, your job is on the line.

Possible Solutions

These situations, and a thousand others just like them, confront businesses each day. In our world of massive connectivity, these types of requests are not unreasonable; even if that is the case, it does not mean these requests are easy to fulfill.

An HTML Solution

The first candidate to explore when you're looking to move information across the Internet is, of course, Hypertext Markup Language (HTML). You could use one of a number of tools for creating data-driven HTML pages. This would include Microsoft's ASP.NET, any of a number of Java environments, PHP—the list goes on and on.

Each of these environments is good at creating dynamic web content. However, they all take time and a certain level of programming knowledge. With deadlines looming, you may not have the time to create custom web applications to solve each of these problems. If you are used to manipulating data with Crystal Reports or Access reporting, you may not be ready to jump into full-blown application development, and you may not have a desire to do so at any time in the near future.

Even if you did create an application for each of these scenarios, one important requirement in each case is this: the information must be printable. HTML screens can look great in a browser window, but they can cause problems when printed. The content can be too wide to fit on the page, and there can be problems with page breaks. These types of formatting issues could make the output difficult for the sales representatives and the production manager to read. Asking the CEO to take this type of a report to the executive meeting could get you fired.

Let's look for another option!

A PDF Solution

Because the capability to control the printed output is important, Adobe PDF should be considered. Portable Document Format (PDF) files look good, both on the screen and in print. You can control where the page breaks occur and make sure everything looks great. However, several issues need to be overcome with PDF files.

First of all, you need some type of utility to produce output in a PDF format. This could be Adobe's full version of Acrobat or some other utility. Once this has been obtained, a document must be created that contains the desired database information. This is usually a report created with a reporting tool or development software. After this document is created, it is converted into a PDF document using an export function or a special printer driver.

Once the PDF document has been created, it can be copied to a website for access through the Internet. However, as soon as the PDF document is created, it becomes a static entity. It does not requery the database each time it is requested from the website. To remain up-to-date, the PDF document must be re-created each time the source data is changed. In addition, you may have to return to your programming environment to control access to the PDF documents on the website.

Perhaps there is a better way.

A Third-Party Reporting Environment

Reporting environments from other companies certainly overcome the limitations of our first two options. These third-party products allow reports to be built without requiring

large amounts of programming. They can also dynamically generate output in a format such as Adobe PDF that will perform well onscreen and in print.

The problem with third-party reporting environments is the cost. Some products can run into the thousands or tens of thousands of dollars. This can be enough to break the budget—if indeed there is a budget—for reporting projects such as the ones discussed previously.

Microsoft Reporting Services

Now you can begin to see why companies get so excited about Reporting Services. It provides an elegant solution for all three of your demanding users—the production manager, the vice president of sales, and the chief executive officer. Reporting Services does not have the drawbacks inherent in the possible solutions considered previously.

No Programming Required

Reporting Services provides a simple, drag-and-drop approach to creating reports from database information. You can use a number of different tools to author reports. The Report Builder and the Report Designer in Visual Studio/SQL Server Data Tools let you truly unlock the power of Reporting Services to convey complex information.

You do not need to be a programmer to create Reporting Services reports. However, if you are comfortable with programming constructs, Chapters 8 and 9 include some simple Visual Basic expressions that can be used to spice up your report's presentation. Note, however, these expressions are not necessary to create useful reports. They are also simple enough that even those who are totally new to Visual Basic can master them with ease.

A Server with a View

With Reporting Services, you can view reports in your browser. Reporting Services provides a high-quality presentation of each report using dynamic HTML. Reports are presented in multiple pages with “VCR button” controls for navigating between pages.

Because Reporting Services uses dynamic HTML, it does not require any additional programs to be downloaded on your PC. There is no ActiveX control to install, no Java applet to download. Any browser that supports HTML 4.0 can view reports.

Plays Well with Printers

In addition to presenting reports in your browser using dynamic HTML, Reporting Services can *render* a report in a number of additional formats. These include an Adobe PDF document, a Tagged Image File Format (TIFF) image, an Excel spreadsheet, and



even a Word document. All these formats look great onscreen when they are viewed, or on paper when they are printed.

NOTE

When Reporting Services renders a report, it gathers the most recent data from the database, formats the data in the manner the report's author specifies, and outputs the report into the selected format (that is, HTML, PDF, TIFF, and so on).

Even when being output in the PDF or TIFF format for printing, a report can be configured to requery the database every time it is accessed. This ensures the report is always up-to-date.

Special Delivery

Reporting Services provides several different ways to deliver reports to end users. Using either the Report Manager website or SharePoint users can access reports via an intranet or the Internet. Reporting Services also includes security features, which ensure that users access only the reports they should.

Users can also subscribe to reports they would like to receive on a regular basis. Reporting Services will send out a copy of the report as an e-mail attachment to each subscriber on a regularly scheduled basis. Alternatively, a Reporting Services administrator can send out a copy of the report as an e-mail attachment to a number of recipients on a mailing list. If that isn't enough, reports can be embedded right in .NET applications.

The Price Is Right

For anyone who has a licensed copy of SQL Server 2012, the price of Reporting Services is certainly right. Free! As long as the report server is installed on the same computer as the SQL Server database engine, your SQL Server 2012 license covers everything. With this single server architecture, it will not cost you one additional penny to share your reports with others using Reporting Services.

Reporting Services to the Rescue

Let's take one more look at the three scenarios we considered earlier—the production manager, the vice president of sales, and the chief executive officer. How can you use the features of Reporting Services to fulfill the requests made by each of them?

The production manager wants a report showing the current inventory. It is certainly not a problem to query the inventory data from the database and put it into a report. Next, he wants to get a new copy of the report every time the inventory is updated during business hours. The production manager can subscribe to your inventory report and, as part of the subscription, ask that a new report be delivered at 8:15 A.M., 12:15 P.M., and 4:15 P.M. Finally, the inventory system is in Cleveland, but the production manager is

in Portland. Because a subscription to a report can be delivered by e-mail, the Reporting Services server can be set up in Cleveland, produce the report from the local data source, and then e-mail the report to Portland.

The solution for the vice president of sales is even more straightforward. He wants a report with credit information for each client. No problem there. Next, he wants the report available to his sales staff, accessible via the Internet. To achieve this, you can publish the report on the Report Manager website. You can even set up security so only sales representatives with the appropriate user name and password can access the report.

In addition, the vice president of sales wants the report to look good when printed. This is achieved with no additional work on the development side. When the sales representatives retrieve the report from the website, it is displayed as HTML. This looks good in the browser, but it may not look good on paper. To have a report that looks good on paper every time, the sales representatives simply need to export the report to either the PDF or TIFF format and then display and print the exported file. Now they are ready to go knocking on doors!

For the CEO, you can build a report or, perhaps a series of reports, that reflects the state of her company. This will serve to keep her informed on all facets of her business. To have this available on her desktop at 7:00 A.M., you can set up a subscription that will run the reports and e-mail them to her each morning at 6:15 A.M.

Finally, because she wants to print this report and share it with the corporate vice presidents, you can make sure the subscription service delivers the report in either PDF or TIFF format. The best part is that because you already have a SQL Server 2012 license, the Reporting Services solution costs the company nothing. You have earned a number of bonus points with the big boss, and she will make you the chief information officer before the end of the year!

Report Authoring Architecture

As mentioned previously, Reporting Services reports are created using either the Report Builder or the Report Designer. The *Report Builder* supports the construction of full-featured Reporting Services reports. It features a user interface similar to that of Microsoft Word 2010 or Microsoft Excel 2010, so it should be familiar for users comfortable with those products. The *Report Designer*, found in SQL Server Data Tools and Visual Studio, also supports all of the features of Reporting Services. In addition, it provides tools for project organization and source-code management for those reporting projects that have a lifecycle similar to that of a software development project (version control, check-in/check-out, etc.).

This book can help you get the most from the incredibly rich report-authoring features available in Report Builder and the Report Designer. These environments

contain everything necessary to create a wide variety of reports for Reporting Services. Everything you need to select information from data sources, create a report layout, and test your creation is right at your fingertips.

Report Structure

Each Reporting Services report contains two distinct sets of instructions that determine what the report will contain. The first is the data definition. The *data definition* controls where the data for the report will come from and what information will be selected from that data. The second set of instructions is the report layout. The *report layout* controls how the information will be presented on the screen or on paper. Both of these sets of instructions are stored using the Report Definition Language (RDL).

Figure 1-1 shows this report structure in a little more detail.

Data Definition

The data definition contains two parts: the data source and the dataset. The *data source* is the database server or data file that provides the information for your report. Of course, the data source itself is not included in the report. What is included is the set of instructions the report needs to gain access to that data source. These instructions include the following:

- The type of source you will be using for your data (for example, Microsoft SQL Server 2012, Oracle, DB2, Informix, or Microsoft Access). Reporting Services will use this information to determine how to communicate with the data source.

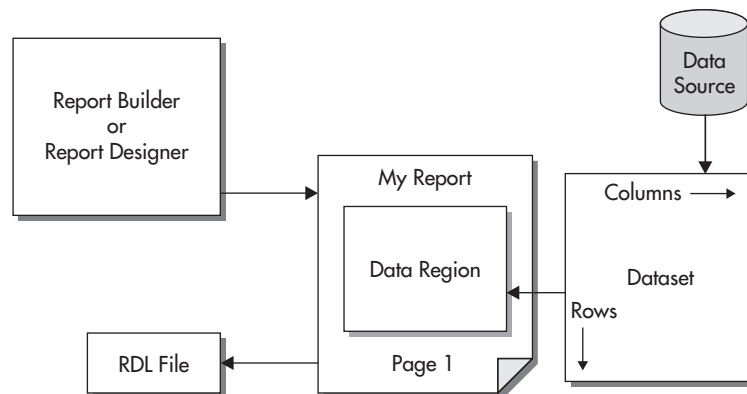


Figure 1-1 Report structure

12 Microsoft SQL Server 2012 Reporting Services

- ▶ The name of the database server or the path to the data file.
- ▶ The name of the database.
- ▶ The login for connecting to this data source, if a login is required.

When the report is executing, it uses the data source instructions contained in the report to gain access to the data source. It then extracts information from the data source into a new format that can be used by the report. This new format is called a *dataset*.

The content of the dataset is defined using a tool called the Query Designer. The *Query Designer* helps you build a database query. The database query may be in Transact-Structured Query Language (T-SQL) for querying relational data, Multidimensional Expression language (MDX) for querying multidimensional data, or Data Mining Expression language (DMX) for querying data-mining data. The query provides instructions to the data source, telling it what data you want selected for your report. The query is stored in the report as part of the data definition.

The data selected by the query into the dataset consists of rows and columns. The rows correspond to the records the query selects from the data source. The columns correspond to the fields the query selects from the data source. (MDX queries are flattened into a table of rows and columns.) Information on the fields to be selected into the dataset is stored in the report as part of the data definition. Only the information on what the fields will be called and the type of data they will hold is stored in the report definition. The actual data is not stored in the report definition, but instead is selected from the data source when the report is run.

Report Layout

The data that the report has extracted into a dataset is not of much use to you unless you have some way of presenting it to the user. You need to specify which fields go in which locations on the screen or on paper. You also need to add things such as titles, headings, and page numbers. All of this forms the report layout.

In most cases, your report layout will include a special area that interacts with the dataset. This special area is known as a data region. A *data region* displays all the rows in the dataset by repeating a section of the report layout for each row.

Report Definition Language

The information in the data definition and the report layout is stored using the Report Definition Language (RDL). *RDL* is an Extensible Markup Language (XML) standard designed by Microsoft specifically for storing report definitions. This includes the data source instructions, the query information that defines the dataset, and the report layout. When you create a report in the Report Designer, it is saved in a file with an .rdl extension.

If you have not worked with XML, or are not even sure what it is, don't worry. The Report Designer, Report Builder, and Reporting Services will take care of all the RDL for you. For those of you who want to learn more about RDL, we'll take a quick peek under the hood in Chapter 8.

Report Designer

Figure 1-2 shows the Report Designer. This is one of the tools you can use for creating and editing reports throughout this book. We will look at some features of the Report Designer now and discuss them in more detail in Chapter 5 through Chapter 9.

Design Surface

The design surface, in the center of Figure 1-2, is where you create your report layout. To do this, you use four of the other areas visible in Figure 1-2: the Report Data window, the Toolbox, the Properties window, and the Grouping pane. You will learn how these work in the following sections. The design surface shares space with the Preview tab. The Preview tab will show you how the report layout and the data combine to create an honest-to-goodness report.

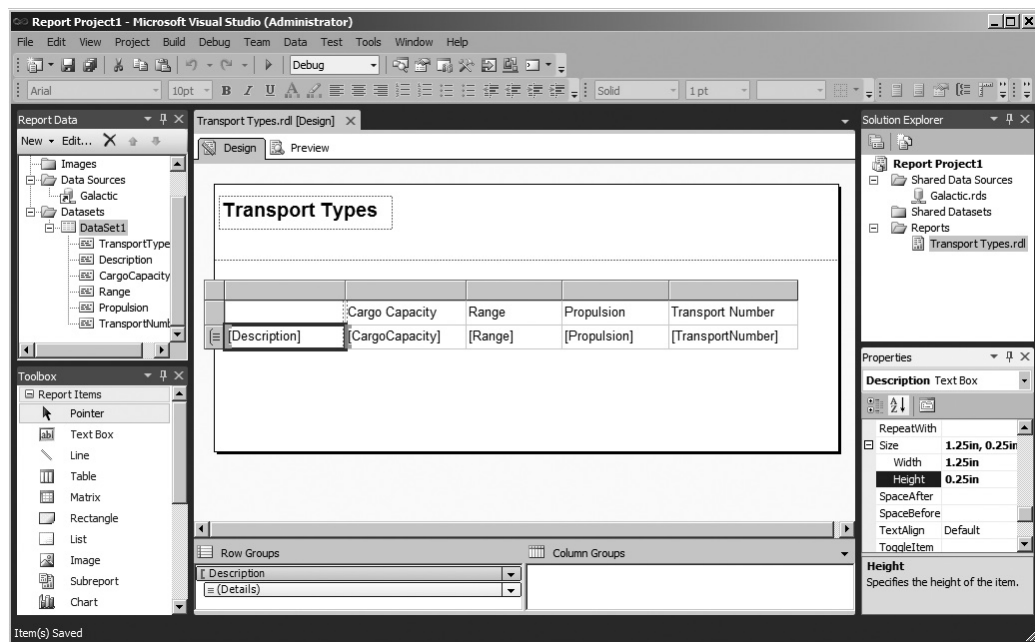


Figure 1-2 *The Report Designer*

Report Data Window

The Report Data window, in the upper-left corner of Figure 1-2, provides a list of database and other types of fields you can use in your report. The Report Data window makes it easy to add database information to your report layout. Simply drag the desired field from the Report Data window and drop it in the appropriate location on your report layout. The Report Designer takes care of the rest.

Toolbox

The *Toolbox* contains all the report items you use to build your reports. These report items, sometimes called controls, are responsible for getting the text and graphics to show up in the right place on your reports. As with any construction project, you can only construct reports properly *after* you learn how to use the tools (report items) in the Toolbox. You learn how to use each of the report items in the Toolbox in Chapters 4, 5, 6, 7, 8, and 9.

As with the fields in the Report Data window, the report items in the Toolbox are placed on the report layout with a simple drag-and-drop. However, whereas fields are pretty much ready to go when they are dropped onto the design surface, report items almost always need some formatting changes to get them just the way you want them. This is done by changing the size, the color, the font, or one of many other characteristics of the report item.

Unlike the layout shown in Figure 1-2, the Toolbox is often hidden on the left side of the screen. This is the default configuration for the Report Designer. When in this layout, the Toolbox is displayed by hovering the mouse over the Toolbox tab.

Properties Window

The Properties window, shown in the lower-right corner of Figure 1-2, is the place where you control the characteristics of each report item. The Properties window always shows the characteristics, or *properties*, for the report item currently selected in the design surface. You will see an entry in the Properties window for every aspect of this report item that you can control.

The top of the Properties window shows the name of the selected report item. In Figure 1-2, the text box named “Description” is selected. The left column in the Properties window shows the name of each property that can be changed for that report item. The right column shows the current setting for each of those properties. For example, in Figure 1-2, you can see the Description text box has a Height of 0.25in.

Grouping Pane

The Grouping pane is at the bottom of the design surface in Figure 1-2. This pane, made up of the Row Groups area and the Column Groups area, is where you control

how grouping operates within the report. The advanced row grouping and column grouping that are now possible in a report are what make Reporting Services a powerful tool for creating complex report layouts.

Report Builder

The Report Builder functions similar to the Report Designer. Figure 1-3 shows a report being created using the Report Builder.

The Report Builder uses the ribbon-style user interface found in Microsoft Office 2010. This is the main difference between the Report Designer and the Report Builder. The Report Builder does not have a Toolbox. Instead, the report items are found on the Insert tab of the ribbon. The other tabs on the ribbon provide additional controls for formatting report items placed on the design surface.

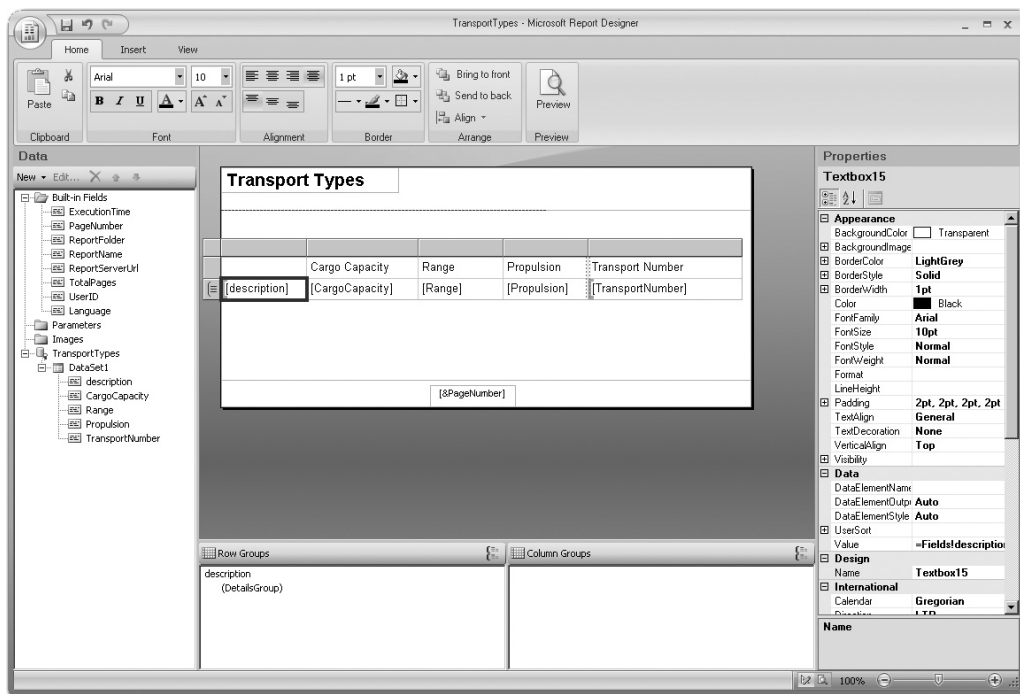


Figure 1-3 *The Report Builder*

Report-Serving Architecture

Once you finish building your report and have it looking exactly the way you want, it is time to share that report with others. This is the time when your report moves from its safe, childhood life inside the Report Designer or Report Builder to its adult life on a report server. This is known as *deploying* or *publishing the report*. Let me assure you, reports pass through deployment much easier than you and I passed through adolescence!

Report Server

The *report server* is the piece of the puzzle that makes Reporting Services the product it is. This is the software environment that enables you to share your report with the masses—at least, those masses who have rights to your server. Figure 1-4 shows the basic structure of the report server.

Report Catalog

When a report is deployed to a report server, a copy of the report's RDL definition is put in that server's Report Catalog. The *Report Catalog* is a set of databases used to store the definitions for all of the reports available on a particular report server. It also stores the configuration, security, and caching information necessary for the operation of that report server.

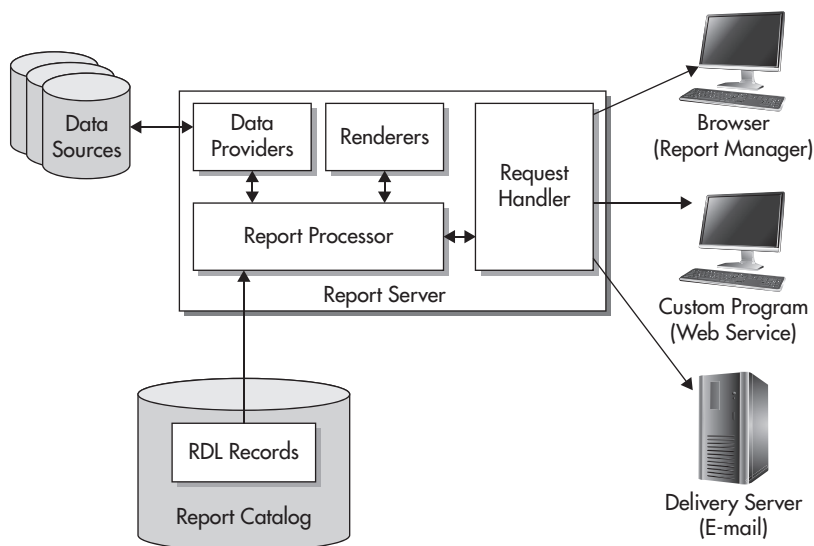


Figure 1-4 Report server architecture

Even though you may use any ODBC- or OLE DB-compliant data source to supply data to your reports, the Report Catalog database can only exist in SQL Server 2005, SQL Server 2008, or SQL Server 2012. The Report Catalog database is created as part of the Reporting Services installation process. Except for creating regular backups of any Report Catalog databases, it is probably a good idea to leave the Report Catalog alone.

Report Processor

When a report needs to be executed, the report processor component of the report server directs the show. The *report processor* retrieves the report from the Report Catalog and orchestrates the operation of the other components of the report server as the report is produced. It takes the output from each of the other components and combines them to create the completed report.

Data Providers

As the report processor encounters dataset definitions in the report RDL, it retrieves the data to populate that dataset. It does this by first following the instructions in the report's data source for connecting to the database server or file that contains the data. The report processor selects a *data provider* that knows how to retrieve information from this type of data source.

The data provider then connects to the source of the data and selects the information required for the report. The data provider returns this information to the report processor, where it is turned into a dataset for use by the report.

Renderers

Once all the data for the report has been collected, the report processor is ready to begin processing the report's layout. To do this, the report processor looks at the format requested. This might be HTML, PDF, TIFF, or one of several other possible formats. The report processor then uses the *renderer* that knows how to produce that format.

The renderer works with the report processor to read through the report layout. The report layout is combined with the dataset, and any repeating sections of the report are duplicated for each row in the dataset. This expanded report layout is then translated into the requested output format. The result is a report ready to be sent to the user.

Request Handler

The *request handler* is responsible for receiving requests for reports and passing those requests on to the report processor. Once the report processor has created the requested report, the request handler is also responsible for delivering the completed report. In the next section, you will learn about the various methods the request handler uses for delivering reports.

Report Delivery

We have discussed how a report is created by the report server. What we have not discussed is where that report is going after it is created. The report may be sent to a user through the Report Manager website. It may be sent in response to a web service request that came not from a user, but from another program. It may also be e-mailed to a user who has a subscription to that report.

Report Manager Website

One way for users to request a report from the report server is through the *Report Manager* website. This website is created for you when you install Reporting Services in Native mode. Figure 1-5 shows a screen from the Report Manager website.

The Report Manager website organizes reports into folders. Users can browse through these folders to find the report they need. They can also search the report titles and descriptions to locate a report.

The Report Manager also includes security that can be applied to folders and reports. With this security, the site administrator can create security roles for the users who will be accessing the site. These security roles control which folders and reports a user

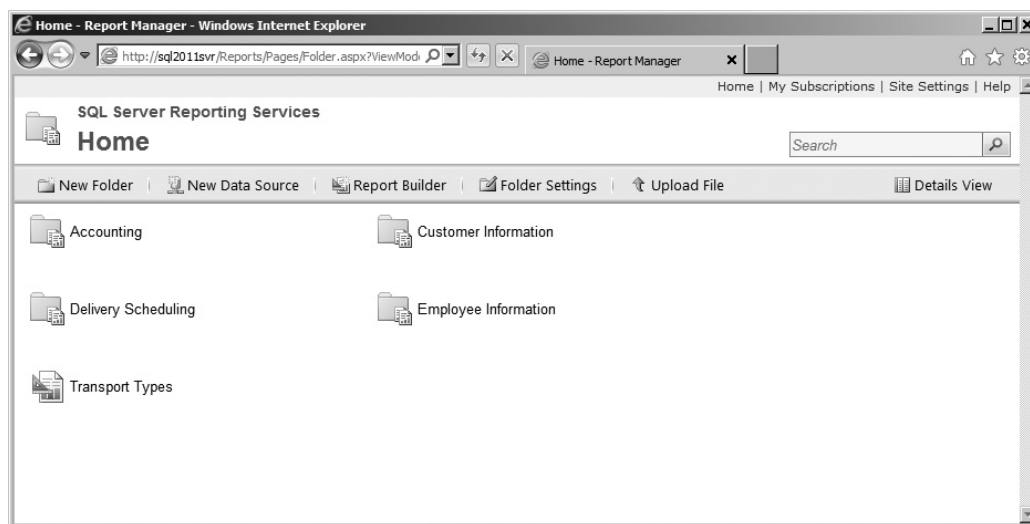


Figure 1-5 *The Report Manager website*

is allowed to access. You will learn about security when we look at the Report Manager in Chapter 10.

In the Report Manager, reports are always displayed using the HTML format. Once a report has been displayed as an HTML page, the user can then export the report into any of the other available formats.

SharePoint

SharePoint may also be set up to serve as a means for users to request reports. This can be done in two ways. The first uses the Report Explorer and Report Viewer web parts. These web parts can be used in a SharePoint web application to allow users to navigate report folders and to view reports on a Reporting Services report server.

The Report Explorer and Report Viewer web parts were originally made available as part of SQL Server 2000 Reporting Services Service Pack 2. They were part of SQL Server 2005 Reporting Services and SQL Server 2008 Reporting Services as well. The web parts continue to be available in the 2012 release, but they have not been upgraded to take advantage of new features and functionality.

The second means of utilizing Reporting Services through SharePoint involves a tight integration of the two products. In this configuration, a SharePoint 2010 installation will actually become the host for the report server's Report Catalog. In addition, the SharePoint user interface replaces the Report Manager website as the user interface for locating and viewing reports, as well as for managing the report server. Accessing reports through SharePoint integration is as easy and intuitive as accessing any other document on the SharePoint site.

Subscription Delivery

If the users do not want to go to the report, the request handler can make the report go to them. In other words, users do not necessarily need to come to the Report Manager website to receive a report. They can have the report delivered to them through a subscription service. The Report Manager enables users to locate a report on the site and then subscribe to it so it will be delivered to them in the future.

When users subscribe to a report, they provide an e-mail address to which the report will be delivered. The content of the report can either be embedded in the body of the e-mail or be included as an e-mail attachment, depending on the requested format. Users can specify the format for the report at the time they create their subscription.

The site administrator can also set up report subscriptions. These function like a mass mailing, using a list of e-mail addresses. Rather than requiring each user to access the Report Manager to create their own subscription, the site administrator can create one subscription that is delivered to every user in a list.

Web Service Interface

In addition to delivering reports to humans, either at their request or on a subscription basis, the request handler can deliver reports to other software applications. This is done through a series of web services. A *web service* is a mechanism that allows programs to communicate with each other over the Internet.

A program calls a web service on the report server, requesting a particular report in a particular format. The request handler relays this request to the report processor, just like any other request for a report. The completed report is returned as the response to the web service request to the program that originated it.

Web services use a standard called Simple Object Access Protocol (SOAP). *SOAP* is supported by both Windows and non-Windows environments, so a program running on a non-Windows computer that supports SOAP can receive a report created by Reporting Services.

Diving In

Now that you have been introduced to all of the capabilities of Reporting Services, I hope you are ready to dive in and make it work for you. In the next chapter, you will learn about the installation and setup of Reporting Services. If Reporting Services has already been installed, you can skip ahead to Chapter 3.

In Chapter 3, we make sure you have a firm understanding of database basics before getting to the actual building of reports in Chapter 4. Chapter 3 also introduces you to Galactic Delivery Services (GDS), the company we use as a case study throughout the remainder of the book. Even if your database skills are tip-top, you should spend a few minutes in Chapter 3 getting to know GDS.