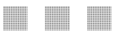
THE EXPERT'S VOICE® IN SQL SERVER

Pro SQL Server Reporting

Pro SQL Server 2012 Reporting Services

Third Edition



Brian McDonald
Shawn McGehee
Rodney Landrum

Apress®

Pro SQL Server 2012 Reporting Services, Third Edition

ISBN 978-1-4302-3810-2

ISBN 978-1-4302-3811-9 (eBook)

President and Publisher: Paul Manning
Lead Editor: Jonathan Gennick
Technical Reviewers: Rodney Landrum and Sherri McDonald
Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Louise Corrigan, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh
Coordinating Editors: Adam Heath, Stephen Moles and Kevin Shea
Copy Editor: Chandra Clarke
Compositor: Bytheway Publishing Services
Indexer: SPi Global
Artist: SPi Global
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `www.springeronline.com`.

For information on translations, please e-mail `rights@apress.com`, or visit `www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `www.apress.com/bulk-sales`.

This book is dedicated to my beautiful wife and my bubbly best friend, Sherri, my two amazing children (Bailey and Kylie), my mother and father, and to everyone else who believed in me and supported me throughout the years.

–Brian K. McDonald

I dedicate this book to my wife, my children, and my parents.

–Rodney Landrum

I dedicate this book to my friends and family, who support me in everything I do.

–Shawn McGehee

Contents at a Glance

- About the Authors xv
- About the Technical Reviewers xvi
- Acknowledgments xvii
- Introduction xix
- Chapter 1: Introducing the Reporting Services Architecture 1
- Chapter 2: Report Authoring: Designing Efficient Queries 19
- Chapter 3: Introduction to Reporting Services Design with SQL Server Data Tools 39
- Chapter 4: Laying Out a Report 61
- Chapter 5: Implementing Dashboard-Style Report Objects 89
- Chapter 6: Building Reports 125
- Chapter 7: Using Custom .NET Code with Reports 185
- Chapter 8: Deploying Reports 213
- Chapter 9: Rendering Reports from .NET Applications 243
- Chapter 10: Managing Reports 279
- Chapter 11: Securing Reports 329
- Chapter 12: Delivering Business Intelligence with SSRS 361
- Chapter 13: Creating Reports Using Report Builder 1.0, 2.0, and 3.0 401
- Index 483

Contents

- **About the Authors** xv
- **About the Technical Reviewers** xvi
- **Acknowledgments** xvii
- **Introduction** xix
- **Chapter 1: Introducing the Reporting Services Architecture** 1

Understanding the Benefits of SSRS 2

SQL Server 2008 R2 and 2012 Reporting Services Enhancements 3

Report Builder/Data Modeler 3

SSRS 2012 Integration with Microsoft Office SharePoint 4

Tablix Report Properties 4

Enhanced Charting and Report Item Visualizations 4

Enhanced Performance and Memory Management 4

Embeddable SSRS Controls 5

HTML Text Formatting 5

Microsoft Word Rendering 5

Report Parts 5

Lookup Functions 6

Shared Datasets 6

SSRS and Business Intelligence 6

Business Intelligence Development Studio and SQL Server Data Tools 6

SQL Server Management Studio (SSMS) 6

SSRS Architecture 7
SSRS Databases 8
The SSRS Report Server 9
SSRS Web Service Interface 9
Authentication Layer 10
The Report Processor 10
Data Processing 10
Report Rendering 11
Scheduling and Delivery 11
Client Applications 12
Report Manager 12
Business Intelligence Development Studio (BIDS) and SQL Server Data Tools (SSDT) 13
Command-Line Utilities 14
Report Builder 14
Custom Clients 15
Installing and Configuring 15
Deploying SSRS Securely 16
Summary 17
Chapter 2: Report Authoring: Designing Efficient Queries 19
Introducing the Sample Relational Database 19
Introducing the Schema Design 20
Knowing Your Data: A Quick Trick with a Small Procedure 21
Introducing Query Design Basics 23
Creating a Simple Query Graphically 23
Creating an Advanced Query 26
Testing Performance with SQL Server Management Studio (SSMS) 28
Optimizing Performance: Dividing the Load 30

Using a Parameterized Stored Procedure 32
Using ISNULL to Evaluate the Parameters 34
Query Performance and Indexes 34
Column and Table Aliasing 35
Testing the Procedure 35
Summary 36
Chapter 3: Introduction to Reporting Services Design with SQL Server Data Tools 39
Exploring the Elements of BIDS 40
Setting Up a Basic IDE 43
Understanding Report Definition Language (RDL) 45
Adding a Report 46
Setting Up Data Sources and Datasets 47
Creating a Data Source 47
Creating a Dataset 49
Creating Other Data Sources 54
Configuring Parameters 54
Setting up Filters 56
Expressions 58
Summary 59
Chapter 4: Laying Out a Report 61
Setting Up Pagination 61
Using Report Objects 62
Implementing a List 63
Implementing a Textbox 69
Implementing a Table 75
Implementing a Rectangle 80

Implementing a Matrix 85
Summary 88
Chapter 5: Implementing Dashboard-Style Report Objects 89
Understanding the Chart Data Region 90
Implementing an Image 95
Implementing a Gauge 100
Implementing a Map 103
Implementing a Data Bar 110
Implementing a Sparkline 115
Implementing an Indicator 119
Summary 123
Chapter 6: Building Reports 125
Creating a Report with the Report Wizard 126
Building Reports from Scratch 129
Formatting the Output 133
Adding Subtotals 134
Adding Interactivity 136
Document Mapping 137
Visibility 138
Interactive Sorting 141
Hyperlink Actions 143
Adding a Bookmark Link 147
Adding a URL Link 149
Building the URL Link with a Report Parameter 150
Jumping to a Report 151
Adding Hyperlink Formatting and Tooltips 157

Setting Report Parameters with Stored Procedures 158
Working with Multivalued Parameters 165
Applying a Filter 170
Adding a Chart 171
Adding Tablix Elements 175
Configuring Report and Group Variables 177
Adding the Gauge Control 179
Adding the Final Touches 181
Summary 183
Chapter 7: Using Custom .NET Code with Reports 185
Using Embedded Code in Your Report 186
Using the ExceedMaxCosts Function 187
Using the ExceedMaxCost Function in a Report 190
Accessing .NET Assemblies from Embedded Code 194
Using Custom Assemblies with Your Report 195
Adding a Class Library Project to Your Reporting Solution 195
Deploying a Custom Assembly 198
Adding an Assembly Reference to a Report 202
Debugging Custom Assemblies 207
Troubleshooting Your Project 210
Summary 211
Chapter 8: Deploying Reports 213
Using Report Manager 214
Using Report Builder 3.0 218
Using BIDS and Visual Studio 2012 221
Configuring Report Deployment Options 221

Setting Up Deployments Using the Configuration Manager 223
Deploying Reports Through the Solution Explorer 224
Using the rs.exe Utility 226
Using the Report Server Web Service 230
Accessing the Web Service 231
Laying Out the Form 233
Coding the Form 234
Allowing Users to Enter a Server Name 235
Populating the TreeView Control With a List of Folders 235
Opening the RDL File and Uploading It to the Server 237
Running the Application 239
Summary 242
Chapter 9: Rendering Reports from .NET Applications 243
Implementing URL Access 245
URL Report Access Path Format 246
URL Parameters and Prefixes 246
Report Parameters 247
HTML Viewer Commands 247
Report Server Command Parameters 248
Credential Parameters 250
Report Viewer Web Part Commands 250
Example URLs 251
Integrating SSRS 2012 with .NET Applications 251
Building a custom Report Viewer Using the WebBrowser Control 251
Building the Report Viewer Using a Report Viewer Control 255
Rendering the Report Locally 259
Creating the Report's Data Source 259

Designing the Report 260
Using the Report Server Web Service 262
Web Services Method Categories 263
Creating the GetParameters Form 263
Coding the Report Parameters Form 266
The GetParameters_Load Event 266
Calling the Web Services GetItemParameters Method 267
Rendering the Final Report 271
Building the Report Viewer in ASP.NET 274
Summary 278
Chapter 10: Managing Reports 279
Exploring Management Roles in SSRS Deployment 279
Managing Content 280
Setting Up Shared Schedules 280
Creating a Shared Schedule 281
Configuring a Report to Use a Shared Schedule 282
Updating and Uploading the RDL File Using Report Manager 284
Setting Up a Data Source for the Report 290
Altering Report Data Sources 292
Creating Snapshots for the Report History 293
Processing Reports and Performing Caching 295
Managing Subscriptions 297
Managing Standard Subscriptions 298
Creating a Standard Subscription 299
Configuring the Subscription 301
Managing Data-Driven Subscriptions 303
Designing the Subscription Query 303
Creating the Data-Driven Subscription 305

Performing Execution Auditing and Performance Analysis 308
Configuring SSRS Logging 309
Transforming the ExecutionLog Table 309
Designing the Log Report 312
Monitoring Performance 313
Controlling SSRS Programmatically 315
Controlling SSRS with SOAP 315
Controlling SSRS with WMI 326
Summary 327
Chapter 11: Securing Reports 329
Encrypting Data 329
Introducing Encryption 330
Securing Network Traffic Using SSL 330
Analyzing HTTP Traffic 330
Applying an SSL Certificate 334
Capturing HTTPS Traffic 335
Securing Data Storage in SSRS 337
Setting Up Authentication and User Access to Data 339
Introducing SSRS Roles 340
Testing SSRS Role Assignments 344
Filtering Report Content with User!UserID 352
Setting Data Source Security 354
Setting SQL Server Permissions 355
Auditing Reports 356
SSRS Auditing 356
Introducing Log File Auditing 358
Summary 359

Chapter 12: Delivering Business Intelligence with SSRS 361

Building SSRS Reports for SQL Analysis Services 362

Using an Analysis Service Cube with SSRS 367

Setting Up the Analysis Services Data Source 368

Working with the Graphical MDX Query Builder 369

Incorporating SSRS with Microsoft SharePoint 2010 373

Installing SharePoint 2010 and SQL Server 2012 on a Stand-Alone Server 373

Installing SharePoint 2010 374

Installing SQL Server 2012 Reporting Services in SharePoint Mode 375

Configuring SharePoint 2010 377

Installing and Starting the Reporting Services SharePoint Service 379

Creating a New Reporting Services Service Application 381

Configuring Reporting Services Integration with SharePoint 384

Deploying Reports in a SharePoint-Integrated SSRS Installation 388

Creating a Simple Dashboard to Display SSRS Reports 394

Creating Data Alerts 397

Summary 399

Chapter 13: Creating Reports Using Report Builder 1.0, 2.0, and 3.0 401

Getting User Feedback 402

Introducing the Report Model 404

Create a Report Model using BIDS 2008 R2 405

Adding a Data Source 406

Creating a Data Source View 409

Creating a Report Model 417

Creating Reports with Report Builder 1.0 423

Creating a Table Report 427

Creating a Matrix Report 435

Creating a Chart Report 441
Creating Reports with the Report Builder 2.0 Wizard 447
Creating Reports with Report Builder 2.0 454
Creating Reports with Report Builder 3.0 464
Report Parts 473
Summary 482
Index 483

About the Authors

Brian K. McDonald, MCDBA, MCSD, is a business intelligence and data warehouse geek working for Acosta, the top sales and marketing company in the US. Brian was born and raised in Cincinnati, Ohio but he currently lives in Jacksonville, Florida with his beautiful wife and his two little ninjas, Bailey and Kylie. As the former owner of Business Enterprise Solution Technologies and an independent consultant, Brian has experiences in architecting data warehouses, data modeling, database administration, SQL Server Reporting Services, Integration Services, Analysis Services, and PowerPivot. He has also worked as an application developer, network administrator, and database administrator throughout his 13 years in the IT industry. A characteristic groomed while serving his country in the United States Marine Corps is the pride Brian takes in all his projects and he diligently works to not only meet expectations, but greatly exceed them. He continually strives to improve his skill-set, both professionally and personally. In addition to having a Business Management and Computer Information Systems, he is a Microsoft Certified Solution developer (MCSD) and a Microsoft Certified Database Administrator (MCDBA). Brian is an active member in the SQL Server community. He enjoys giving back to the SQL Server community through training, virtual and public speaking engagements, as well as by writing articles and blogs for many industry leading websites like SQLBIGeek, SQLServerCentral, and SQLServerPedia. You may have seen Brian presenting at the inaugural SQL Rally in Orlando, FL, or at one of the many SQL Saturdays, Code Camps, or User Groups that he has been a part of throughout the years. In those moments when he is not giving back to the community or working, Brian enjoys "geeking" out with his wife Sherri and allowing his children to practice their UFC style moves on him.

Shawn McGehee is a DBA and manager for a large insurance company in the US. He has been working with SQL Server in some fashion since 7.0 and has been working in the IT field since 1998. Starting as a developer, he has gradually made the move to a full time administration role, but still enjoys the challenges he can solve using his development background whenever he can. He is active in the SQL Server community and enjoys speaking at user groups, SQL Saturdays and the odd Code Camp. He is currently running the Orlando SQL Server User Group OPASS and has been involved with the SQL Server community for the last few years. You can usually find him roaming about at a SQL Saturday in the SouthEast, wherever it may be.

About the Technical Reviewers

Rodney Landrum has worked with SQL Server longer than he can remember. He writes regularly about technologies, including Integration Services, Analysis Services, and Reporting Services. (Rodney is the original author of this book). He has authored *SQL Server Tacklebox* and three Reporting Services books. He contributes regularly to SQLServerCentral, SQL Server Magazine, and Simple–Talk. His day job involves overseeing a large SQL Server infrastructure in Orlando. He swears that he owns the phrase "Working with Databases on a Day to Day Basis." Anyone who disagrees is itching to lose an arm wrestling match.

Sherri McDonald is a BI Developer who has specialized in SQL Server Reporting Services as a trainer of Reporting Services and the fundamentals of TSQL. She also is known for her Reporting Services DVD, co-produced with SSWUG. She has served many roles, with more than 14 years in the service industry, while most recently focusing on the Microsoft BI Stack. Sherri is an active blogger, a member of her local SQL Server User's Group, and has presented at SQL Saturday and Code Camp events throughout Florida. Sherri was born and raised in Cincinnati, Ohio and currently lives in Jacksonville, Florida with her husband Brian and two children. When not learning the latest in technology, Sherri enjoys movies, going on cruises, and other travels with her friends and family.

Acknowledgments

First and foremost, I must thank my bubbly best friend, fellow SQLBIGeek, biggest fan, and the mother of my brilliant children—my beautiful wife Sherri. Without her, I would never have achieved many of the successes that I have achieved. Sherri, you are amazing and I could not have done this without your support, guidance, patience, and endless love. With that being said, I would also need to thank my remarkable children, Bailey and Kylie. Thank you for all of the understanding, support, and patience while I locked myself away in the dungeon for so many hours writing this book. Without you two being such wonderful children, I could not have completed this book.

Next, I would like to thank my friend, coauthor, and technical reviewer, Rodney Landrum, for all of his support and belief in me throughout the years. I am so thankful to be able to call you a friend and I look forward to any future opportunities that we have to mash our brains together. Thank you!

To another friend and coauthor, Shawn McGehee: without all of your hard work and dedication to putting out another great release, this book would not have come to a close. Thank you so much for all of your ideas and contributions.

I would like to thank the Apress editorial team, Kevin Shea, Stephen Moles, Adam Heath, and an extra special thanks to Jonathan Gennick, for all of their encouragement and guidance throughout this project. Through the hundreds of email messages and conversations, you have expertly moved this book along to completion and I look forward to getting to meet you all in person. It has truly been a pleasure working with each of you. Thank you!

To my brilliant and humble SQL Server DBA friend and cousin, Chad Tucker: without that kick in the butt many years ago, I would not be in Jacksonville, Florida. Without that nudge, I wouldn't have pursued public speaking, blogging, and eventually even authoring my first publishable work of art (this book). From the bottom of my heart to the tip of my tongue, thank you!

Another person that I must provide a special thanks to is my good friend Marc Munago. He probably doesn't know this, but he has been a great inspiration to me the last few years with his leadership traits and his ability to turn a penny into a dollar. He has been a true friend and mentor to me. We often have great "idea" sessions where we bounce goals, plans, and aspirations back and forth. Your support is very much appreciated and I thank you!

To my awesome neighbors, Eric and Cathy Eng! Eric is an intelligent entrepreneur, Engineer, and an amazing father. Cathy, being a long time writer, has helped me out several times; she owns and runs a company called Resume Rocketeer (`www.resumerocketeer.com`). Sherri and I always enjoy hanging out with you. Thank you both for all of the late night talks and friendship.

To Scott Gleason, a great friend, supporter, and a huge fan! I am thankful for all of your encouragement and your verbal marketing throughout this process, and the last several years for that matter! You are a great friend and I thank you from the bottom of my heart!

Let me not forget to thank Cricket Weaver and Patrick Barr, who allowed me to fumble through many proof of concepts and thorough testing of Reporting Services, while being a DBA consultant many moons ago! Thank you for the patience and leadership that you both showed when we ramped up on the product. Without that leadership, I probably would have thrown in the towel.

To all of the amazing employees at Acosta, especially the Data Warehouse and Business Intelligence (BI) team with whom I work. Not only are you are the most brilliant team of BI professionals that I have worked with over the years, you are also the most fun to work with. I truly enjoy coming into the office each and every day, as I look forward to what each day might bring. You are the crème de la crème of the Information Technology world. Thank you!

A very special thank you goes out to Tom and Mary Beth Ottke, who have been two of the most influential people in my life. You both are such intelligent, kind, loving, and wonderful people. You accepted me into your family many years ago and you helped me become the man that I am today. Without you, I probably wouldn't have even taken the first step in this journey. It is you who planted the seed that gave me belief—not only in myself, but in Him as well! Thank you for all that you have done for me.

To you, the reader of this book, I hope that we (the authors) will teach you about the latest edition of SQL Server Reporting Services. We thank you for spending your hard-earned money on our book and hope that you find it a valuable resource to have sitting at your fingertips.

Last, and certainly not least, I would like to thank my mother and father for everything you have done for me throughout the years. Your love, guidance, and words of encouragement have always been an inspiration to me. Even when you thought that I wasn't listening, your advice and wisdom were always well received. Without all of the leading by example and your hard work, I would not have followed through with this book.

–Brian K. McDonald

I would like to thank everyone who has helped make this book a reality. Rodney Landrum and Brian McDonald are both great collaborators and I couldn't ask for a better pair of co-authors. The staff at Apress has been excellent as well. With their support and guidance, we have been able to produce a work that we hope will help guide readers into the realm of SSRS. I would also like to thank all of the friends and family who have been supportive of the process along this long road.

–Shawn McGehee

I would like to thank Brian McDonald and Shawn McGehee for their contributions to this edition of the book and for tolerating me over the years. Through them, I have met many others who share our passions for SQL Server and community. This book is but one outlet of their service and dedication to these technologies and to the people like us who work and play with data every day. Well, except for holidays and most weekends. Cheers to them!

I would also like to acknowledge the efforts of the Apress staff, who patiently worked with us to mold this edition into a book we can be proud of.

–Rodney Landrum

Introduction

At its core, the process of designing reports hasn't changed substantially in the past 20 years. The report designer lays out report objects, which contain data from a known source of data, in a design application such as Reporting Services, Business Objects Reports, or Microsoft Access. He or she then tests report execution, verifies the accuracy of the results, and distributes the report to the target audience.

Sure, there are enough differences between design applications to mean that the designer must become familiar with each particular environment. However, there's enough crossover functionality to make this learning curve small. For example, the SUM function is the same in Business Objects Reports as it is in Microsoft Access as it is in Structured Query Language (SQL).

With Microsoft SQL Server 2012 Reporting Services (referred to as SSRS throughout the book), there is, again, only a marginal difference in the way reports are designed from one graphical report design application to another. So, if you do have previous reporting experience, your learning curve for SSRS should be relatively shallow. This is especially true if you come from a .NET environment, because the report designer application for SSRS 2012 is Visual Studio 2010 or the application included with SQL Server 2012, SQL Server Data Tools (SSDT), formerly known as Business Intelligence Development Studio (BIDS). We use BIDS and SSDT interchangeably throughout the book, with most references using BIDS. We have done this mainly because of the role that Reporting Services plays in the Business Intelligence stack of products with SQL Server, but also for readers who may be using prior versions of Reporting Services like SSRS 2008 R2.

Having said all this, several differences set SSRS apart from other reporting solutions:

- It provides a standard reporting platform based on Report Definition Language (RDL), which is the XML schema that dictates the common structure of all SSRS reports. This allows for report creation from any third-party application that supports the RDL schema.
- SSRS is an integral part of the SQL Server 2012 release.
- SSRS offers features out of the box that in other products would be expensive additions to a basic deployment. These features include subscription services, report caching, report history, and scheduling of report execution.
- SSRS can be extended with third party add-ons, custom code, and compiled DLL's.
- SSRS, being a Web-based solution, can be deployed across a variety of platforms.
- SSRS also allows for easy integration with Microsoft's Collaboration Software for the Enterprise: SharePoint 2010.

This book was written in parallel with a real SSRS deployment for a health-care application, so it covers almost every design and deployment consideration for SSRS, always from the standpoint of how to get the job done effectively. You'll find step-by-step guides, practical tips, and best practices, along with code samples that you'll be able to modify and use in your own SSRS applications.

Who This Book Is For

We coauthored the book with the intention of demonstrating how to use SSRS from multiple vantage points. As reporting architects and report developers, we go through the report design and deployment processes using standard SSRS tools such as Report Designer in BIDS and Report Manager. We also show how developers can extend SSRS by creating custom Windows and Web Forms applications.

Prerequisites

The core software that has been used in the examples throughout this book are:

- Microsoft SQL Server 2012
- Microsoft Visual Studio 2010 – used in chapters 7, 8, 9, and 10
- Microsoft SharePoint 2010 – used in chapter 12 with SSRS integration
- Microsoft SQL Server 2008 R2 – used in chapter 13 for ad hoc reporting using Report Models

Each of the aforementioned software is required if you, the reader, have the desire to follow along with the examples throughout the book. Most of the examples were built using SQL Server 2012, but with the exception of chapters 7, 8 and 9, they can be performed on SQL Server 2008 R2.

Downloading the Code

In this book, we use a subset of real databases designed for a health-care application that some of us developed over the years. You can find all of the supporting materials (databases, the data mart database, and cube file used in Chapter 12, the completed RDL files, queries, stored procedures, and .NET application projects, as well as full installation instructions) in the Source Code/Download section of the Apress Web site (`www.apress.com`). With so many other books with similar titles having existed over the years, it may be easier to find this book by using its ISBN number. The 13-digit industry standard ISBN number for this book is 978-1-4302-3810-2.

Contacting the Authors

Should you have any questions regarding any section in the book, please feel free to contact us via our email or twitter accounts. We would love to hear that you have purchased our book, so please feel free to tweet us. We sincerely hope that you get the enjoyment out of reading the book that we had in writing it for you.

Brian K. McDonald
`bmcdonald@sqlbigeek.com`
@briankmcdonald

Shawn McGehee
`shawnnwf@gmail.com`
@SQLShawn

Rodney Landrum
rodneylandrum@hotmail.com
@SQLBeat

CHAPTER 1

Introducing the Reporting Services Architecture

Microsoft's 2003 announcement that it was going to release SQL Server Reporting Services (SSRS) as a SQL Server 2000 add-on stirred up a frenzy of excitement. The product was originally slated for release with SQL Server 2005, so the early release was a welcome event for many. Our software development company decided to embrace SSRS early on and was fortunate to work with Microsoft during the beta phases. In January 2004, the month Microsoft's released SSRS to manufacturing (RTM), we deployed it immediately. We intended to migrate all of our existing reports (which we had developed on as many as five reporting applications and platforms over the previous ten years) to SSRS. We can sum up the reason for the seemingly rapid decision in one word: standardization.

Just as Microsoft wanted to create an industry standard with Report Definition Language (RDL), the Extensible Markup Language (XML) schema that dictates the common structure of all SSRS reports, we wanted to provide a standard reporting solution to our customers. Even in the first version of the product, SSRS delivered almost all the features we needed. Thanks to its extensibility via SSRS's Web service, we could programmatically add other features that weren't already directly supported. In addition, Microsoft was committed to enhancing SSRS for years to come. Some of the features released in the 2005 edition were client-side printing, interactive sorting capabilities, and an ability to define multivalued parameters. There was also a move forward in the self-service business intelligence (BI) arena with Microsoft's first ad hoc Report Builder ClickOnce application.

Microsoft's next release was SSRS 2008. The new release brought on many long-awaited enhancements to include modifications of its architecture, completely revamped report designer, and 2008 R2 brought us significant design updates to the built-in Report Manager application. With the vast updates implemented in the 2008 release, SSRS has taken its place as a key SQL Server component in Microsoft's business intelligence suite of products alongside SQL Server Integration Services (SSIS) and SQL Server Analysis Services (SSAS). Nobody could now think of Reporting Services as just an add-on.

The new features in SSRS 2008 and SSRS 2008 R2 pushed the technology one step further into becoming the reporting development environment of choice for programmers and designers, especially those already skilled with Visual Studio (VS) and Visual Basic .NET (VB.NET). As they were for its predecessors, SSRS 2005, SSRS 2008, and SSRS 2008 R2, the long-awaited features for SSRS 2012 are mostly driven by direct feedback from the user community. Throughout the book, we will demonstrate each of the new features released in 2008, 2008 R2, and 2012 as we show how to design professional reports, applications, and solutions built on Microsoft's BI initiatives. We will focus on SSRS as a whole, building on features from each version from 2000 to 2012; however, we will point out which features are new to SSRS 2008 R2 and SSRS 2012.

Understanding the Benefits of SSRS

Our company based its decision to migrate immediately to SSRS on the following perceived benefits for the company and for our customers:

Standard platform: As well as providing a standard realized with the RDL, our development teams had been using VS .NET as their main development environment. Because SSRS reports were currently developed within this platform, we wouldn't need to purchase additional development software. Our clients would need to purchase only a low-cost edition of a designer—VB.NET, for example—to gain the benefit of developing their own custom reports. In SQL Server 2005, Microsoft included the Business Intelligence Development Studio (BIDS) as a free, alternative report designer. This free development environment has been available with SQL Server ever since, but Microsoft has recently renamed it as SQL Server Data Tools (SSDT). Throughout this book, we will use BIDS and SSDT interchangeably. The BIDS environment runs in the shell of Visual Studio (`devenv.exe`) and, at the time of writing, is based on Visual Studio—VS 2008 for 2008 and 2008 R2, and VS 2010 for the latest release, SQL Server 2012. Anybody who learns to design reports with BIDS will have the advantage of a consistent interface when they move to the full version of Visual Studio, and will need no additional training.

Cost: SSRS is an integral part of SQL Server 2012 and is available in many editions, from Express Advanced to Enterprise in 2008 and even Datacenter edition in 2008 R2. However, because SQL Server 2012 has done away with Datacenter edition, the most feature-rich edition will once again be Enterprise. When you purchase SQL Server, you get SSRS as well. See a complete list of SQL Server 2012 features at `http://tinyurl.com/SQL2012Features`.

Web-enabled: Because SSRS is a Web-based reporting solution; a single deployed report is accessible to a variety of clients, from the browser to custom Windows Forms. In addition, because reports are primarily accessed via Hypertext Transfer Protocol (HTTP) or HTTP Secure (HTTPS), you can view reports from any location that has access to the SSRS Web server. Unless you have a thick client application that requires local reports to be deployed with the application, you can have one central repository for reports to be consumed across the organization.

Customizable: SSRS provides a .NET Web service as a front end, programmatically accessible to extend the delivery of reports beyond the browser. As .NET programmers, we knew we would want to build custom applications to render reports where we could control the look and feel of the report viewer. We show one such application in Chapter 7, which covers report rendering.

Subscriptions: SSRS subscription abilities gave a huge advantage for our company and our clients, as report delivery by e-mail or file-sharing, as well as off-peak processing, were now possible. We show how to set up two different kinds of subscriptions, standard and data-driven, in Chapter 8.

As you'll see, SSRS is a full reporting solution that encompasses many levels of professional expertise, from report design to database administration. In many organizations, especially small- to medium-sized ones, information technology (IT) professionals are asked to perform many jobs. They

write a query and design a report in the morning, perform database backups or restores in the afternoon, and update all the systems before heading home in the late hours of the evening. Sometimes even until the early hours of the next day! But we are sure that we're not the only ones that take such pride in our jobs and always striving to exceed the needs of the business.

Throughout each of our careers, we have all worn many hats in the companies which we have poured our time and devotion to over the years. We have been entrenched in every deployment phase from internal deployments to externally facing web application deployments to our clients, from simple implementations to advanced ones which extended Reporting Services capabilities. By developing efficient stored procedures, thoroughly testable security mechanisms, as well as building and maintaining well designed reports, we have witnessed the day-to-day operation of SSRS from many perspectives.

We have also been responsible for our company's overall strategy for building solutions to analyze and transform data gathered through both our own and third-party applications. To that end, an essential part of our jobs over the years has been to integrate SSRS into the overall BI strategy that incorporated the following:

- Disparate data sources such as Analysis Services Cubes and SQL Server relational databases
- Applications and tools such as Microsoft Excel and Business Scorecards
- Document management systems such as Microsoft SharePoint Portal Server

We'll dive into the details of such integration projects in Chapter 12, which is devoted to BI. We will also explore one of the key advancements of SSRS 2008 R2 and 2012, which is a tighter integration with SharePoint portal server, to the point that SSRS content can now be directly deployed, managed, and viewed all within SharePoint. We'll also show you how sections of reports can be created and served as web parts.

SSRS represents another world, not often seen by an administrator using standard management tools. This world is the domain of the software developer who can extend and control SSRS programmatically, building custom report viewers and deployment applications. In this book, as you work through each step of building a reporting solution for healthcare professionals, you'll see how an administrator can accomplish the task with built-in tools, as well as how a developer can create an application to provide enhanced functionality.

SQL Server 2008 R2 and 2012 Reporting Services Enhancements

There have been many major additions to Reporting Services since its initial release in 2005, but let's look at some of the most significant enhancements made to the SSRS technology in SQL Server.

Report Builder/Data Modeler

The Report Builder application, a feature introduced in SSRS 2005, is a local ad hoc report-designing application intended for use more by report consumers than by report developers. An administrator familiar with the source data creates the business logic and underlying data structures as a data model. With the Report Builder application, the user can create and publish reports based on available models. Microsoft designed Report Builder 2.0, released in SSRS 2008, for Microsoft Ribbon technology, much like Microsoft Word and Microsoft Excel, and it was a significant improvement on Report Builder 1.0. Each enhancement provided a richer development environment and additional content sources, such as Oracle and Analysis Services Cubes. As if that wasn't enough, Report Builder 3.0 made its first appearance with SSRS 2008 R2, with its new data visualization report items and cached result sets.

Chapter 13 tells you how to build and deploy a data model, as well as how to create reports with the Report Builder 1.0, Report Builder 2.0, and Report Builder 3.0 applications.

SSRS 2012 Integration with Microsoft Office SharePoint

While SharePoint integration was available with the use of SharePoint controls in previous versions of SSRS, SSRS 2012 takes the integration several steps further. By using SSRS 2012 in SharePoint Integration Mode, users can deploy, manage, and deliver reports and report objects, like web parts, data sources, and models, even entire dashboards or portals, all within the SharePoint environment. In addition, the deployed reports inherit the native features of SharePoint, such as workflow capabilities and the ability to check in and check out reports, and report change notification. We will demonstrate this tighter integration with SharePoint in Chapter 12.

Tablix Report Properties

As the name suggests, the Tablix properties first seen in SSRS 2008 combine two existing report controls, Table and Matrix. This combination gives developers a more flexible tool when creating reports. The availability of multiple columns and rows blends the static nature of the Table control with the dynamic nature of the Matrix. Reports can now accommodate multiple parallel rows and column members at each level, independent of each other but using the same aggregate calculations. In previous editions of this book, we provided workarounds to combining tables and matrices by embedding one within the other. In Chapter 4, we will explore the true power of the new Tablix control properties for the List, Table, and Matrix controls.

Enhanced Charting and Report Item Visualizations

From the beginning, SSRS offered charts and visualizations natively in reports. These charts, while versatile, were somewhat limited in scope. Much, if not all, of the functionality in the charting aspects of previous versions of SSRS could be easily duplicated in Microsoft Excel. In fact, the charting was almost identical. SSRS 2008 provided several charting and graphical data-visualization enhancements vital for the sound BI reporting solution of which SSRS is a pivotal component. New charting elements such as range, polar, radar, funnel, and pyramid are available, as well as many new "gauges" delivered with the acquisition of Dundas reporting controls for SSRS.

SSRS 2008 R2 included several eagerly-awaited report item visualizations to enable the creation of a more sophisticated dashboard look and feel. One is the Map control that can display data from a geospatial data result set or an Environmental Systems Research Institute, Inc. (ESRI) shape file. Other great additions include Data Bars, Sparklines, and Indicators. We will explore several of these new visualizations as we incorporate them into reports in Chapter 5.

Enhanced Performance and Memory Management

Microsoft reengineered the report engine in SSRS 2008 to lessen the memory footprint for reports at the server level, speeding delivery of reports to end-user applications. This enhancement also resolves the contention that arose when long-running, large reports and smaller, non-memory-bound reports processed simultaneously.

Embeddable SSRS Controls

The ability to embed controls in custom applications makes it easier for developers to integrate SSRS into their projects. Since the release of SQL Server 2005, the Visual Studio environment has included distributable controls that you can use for Windows Forms development and ASP.NET Web Forms development. These controls provide additional benefits to developers, such as the ability to render reports while disconnected from the SSRS. We will cover updated SSRS controls in Chapter 9.

HTML Text Formatting

Aside from the change from dual to single service architecture and the ability to export to Microsoft Word format, text formatting is probably one of the most significant advancements of SSRS 2008. In previous versions of SSRS, in-line formatting of textual content, for example for a form letter, was not possible. For example, if you wanted to have a single textbox contain some text in regular font, but wanted to bold or italicize other sections of the text, you wouldn't be able to do it. Textbox report items in SSRS 2008, SSRS 2008 R2, and SSRS 2012 allow for normal and rich-text modes and allow formatting in the same way as a word processor does. You can create a placeholder to allow a limited subset of some HTML and style tags. The text formatting can combine both literal text and data source text for mail merge and template reports. We will demonstrate the full use of this feature by creating a custom form letter style report in Chapter 6.

Microsoft Word Rendering

Since the first version of SSRS, you could export any report to Microsoft Excel. While this was an important capability, not being able to export to other Microsoft Office formats, such as Word, was a limitation. Developers often want to create reports using the rich text formats found in today's modern word processors. By combining SSRS's ability to design custom reports from multiple data sources with Word's ability to provide rich formatting, SSRS 2008 overcomes significant limitations of its predecessors. Another limitation was that report users could not export into 2007 formats. Excel 2003 has a limitation of 65,536 rows and 256 columns, but one of SSRS 2012's new rendering enhancements enables us to export to Word and Excel 2007-2010 formats, so we can now store 1,048,576 records and 16,384 columns on one sheet of an Excel workbook.

Report Parts

If you're like us, you have probably wanted to create little reusable objects that could be incorporated in more than one report. Until SSRS 2008 R2, you could do this only by creating reports that could be embedded into other reports as subreports. Now, you can publish individual sections of reports, like a Tablix containing the top 10 employees by sales totals or a Sparkline showing the customer complaints trend for the current year. Any report item like these can be deployed to a ReportServer or SharePoint server and reused by end users using ad hoc tools like Report Builder or SharePoint. Furthermore, report developers can use these report parts to reduce duplicate efforts for reports that need the data represented in the same fashion. A very useful feature is that if the report part is modified by a user with appropriate permissions, the consumers of that report part are notified about the update and they can choose to refresh their report or to leave it as it was.

Lookup Functions

Before 2008 R2, the need often arose to find a value in a different data region in the same report. Since the data in both of the tables could be linked by a common field, one workaround was simply to gather the data in the source query by joining the tables together. Often, this may perform better on the backend, but sometimes you need to look up values in another data region, so Microsoft has added three lookup functions to Reporting Services: Lookup, LookupSet and MultiLookup.

Shared Datasets

Before moving on from the upgrades released with SSRS 2008 R2 and 2012, let's briefly talk about shared datasets. This feature, added in 2008 R2, allows you to create a dataset that can be consumed among other reports. Imagine you have created a project with 50 reports, about 10 of them with a parameter for all of the countries throughout the world. Thinking of manageability, you designed this dataset to be loaded by making a call to a stored procedure. In previous releases of SSRS, you would have needed to create a dataset for every report that needed this parameter, so any change that affected them all would have had to be made in them all, one at a time. From 2008 R2 onward, we can create a shared dataset and use it across reports. A change in that single dataset updates all the reports that need that change.

SSRS and Business Intelligence

SSRS is just one component of Microsoft's BI platform. We'll now cover other new features and enhancements since SQL Server 2008 that will form an integral part of your overall reporting solution.

Business Intelligence Development Studio and SQL Server Data Tools

Business Intelligence Development Studio (BIDS) is a limited version of Visual Studio 2008, included with the SQL Server 2008 base installation. In SQL Server 2012, the report designer takes on a new name, SQL Server Data Tools (SSDT), and we now have the Visual Studio 2010 shell rather than VS 2008. With SSDT and BIDS, developers can create entire projects for each of the supported components of SQL Server 2012, including SSIS, SSAS, and of course SSRS. We will use SSDT throughout the book (except in Chapter 13, where we use Report Builder to show you how to design and deploy SSRS reports and Analysis Services projects). Note that SSDT and BIDS both use the `devenv` executable and as such, can be used interchangeably.

SQL Server Management Studio (SSMS)

With the release of SQL Server 2008, Microsoft continued to build on its management platform with SQL Server Management Studio (SSMS). Microsoft has taken a big step toward consolidating, in a single environment, many of the tools that in previous versions of SQL Server would have been executed individually. SSMS replaces Enterprise Manager and Query Analyzer, offering a much more elaborate set of tools for creating and managing SQL Server objects and queries. In addition to managing SQL Server and Analysis Services servers, administrators can use SSMS to manage instances of their SSRS reporting servers. We have heard in the SQL Server community that Management Studio will run in the Visual Studio shell, but for now, anyway, it still runs with `ssms.exe`. However, SSMS users will now be able to undock windows and have them on multiple monitors as Visual Studio developers have done for some years.

Throughout the book, we will show you how to use both SSMS and Report Manager for various tasks. For example, we will show you how to use SSMS to test query performance and the browser-based Report Manager to view published reports, set security permissions, and create subscriptions. Although the two applications share functionality for managing SSRS, Report Manager is often preferable to SSMS because it can perform many more administrative tasks and does not require a local installation. You can access Report Manager from a browser anywhere on your network, but you would need access to the installed SQL Server 2012 client tools to use SSMS.

SSRS Architecture

You've probably heard the expression that the devil is in the details. You'll be drilling into those details throughout the book, right down to the data packets that SSRS constructs, as you explore each aspect of SSRS from design to security. For now, let's pull back to a broader vantage point—the 10,000-foot view if you will—and look at the three main components that work together to make SSRS a true multi-tier application: the client, the report server, and the SQL Server report databases. Figure 1-1 shows the conceptual breakdown of the three component pieces.

The data source and the SSRS databases, `ReportServer` and `ReportServerTempDB`, are separate entities. The data source is the origin of the data that will populate the reports, while the report server databases store metadata and execution information about the reports. Both the data source and the report server databases can physically be located on the same SQL Server, assuming the data source is a SQL Server database. The data source can be any supported data provider, such as SQL Server, Oracle, Lightweight Directory Access Protocol (LDAP), Microsoft SharePoint List, SQL Azure and Analysis Services. It's possible to configure a single server to act as both the SSRS report server web service and report server database as well as the data source server. However, this isn't recommended unless you have a small user base. We'll show you how to monitor the performance of the SSRS configuration and build a small Web farm, post-installation, in Chapter 10.

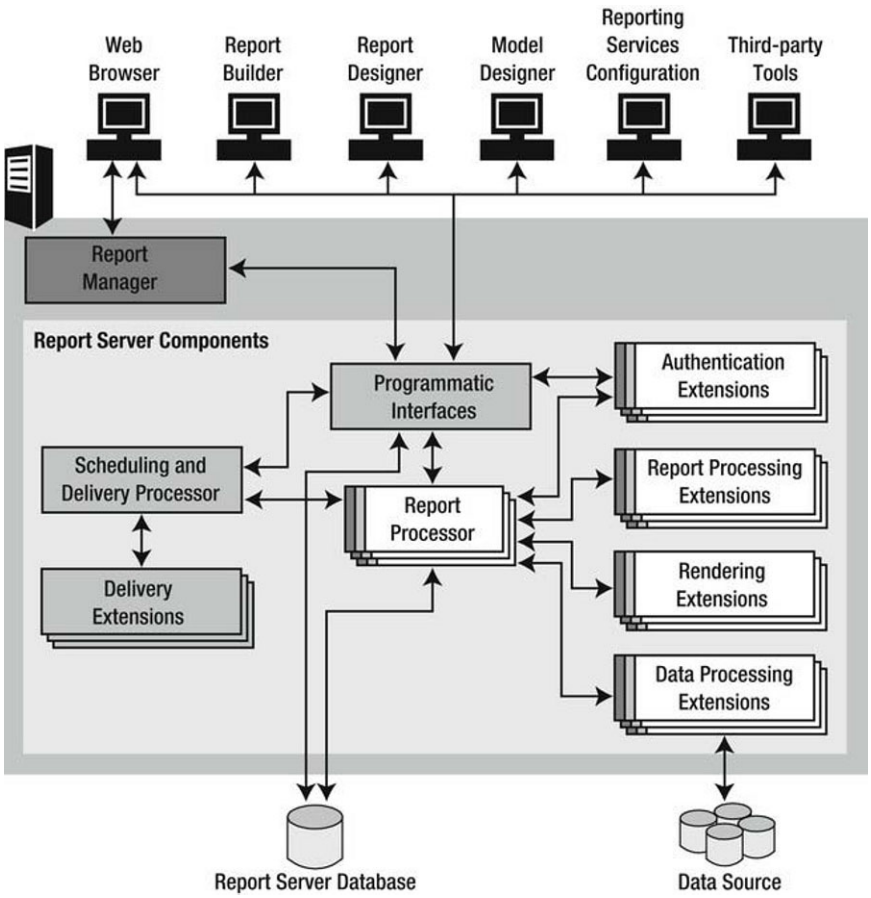


Figure 1-1. *SSRS components*

SSRS Databases

SSRS is added as an option during the SQL Server installation process. The SSRS native installation creates two databases that are used to store report metadata and manage performance:

`ReportServer`: This is the primary database that stores all the information about reports that was originally provided from the RDL files used to create and publish the reports to the `ReportServer` database. In addition to report properties (such as data sources) and report parameters, `ReportServer` also stores folder hierarchy and report execution log information.

`ReportServerTempDB`: This database houses cached copies of reports that you can use to increase performance for many simultaneous users. By caching reports using a nonvolatile storage mechanism, you make sure they remain available to users even if the report server is restarted.