

the **1**st **COMPREHENSIVE AND AFFORDABLE** Guide Series!
FREE Bonus Material at www.rationalpress.com

The Rational Guide To:

SQL Server Reporting Services

Anthony T. Mann

In this guide you'll learn how to...

- ✓ *Use Microsoft's new reporting platform*
- ✓ *Empower users to quickly analyze data*
- ✓ *Customize reports*



*Rational Guides for a
Fast-Paced World™*

Product Unit Manager
SQL Server Reporting Services
Microsoft Corporation

To order this book or to find additional details, go to www.rationalpress.com

PUBLISHED BY

Rational Press - An imprint of the Mann Publishing Group
208 Post Road, Suite 102
Greenland, NH 03840, USA
www.rationalpress.com
www.mannpublishing.com
+1 (603) 601-0325

Copyright © 2004 by Mann Publishing Group.

All rights reserved. No part of the contents of this book may be reproduced in any form or by any means without the written permission of the publisher. For questions about rights and permissions, send e-mail to permissions@mannpublishing.com.

ISBN: 0-9726888-9-7

Library of Congress Control Number (LCCN): 2004090129

Printed and bound in the United States of America.

Trademarks

Mann Publishing, Mann Publishing Group, Agility Press, Rational Press, Inc.Press, Farmhouse Press, The Rational Guide To, Rational Guides, ExecuGuide, AdminExpert, From the Source, the Mann Publishing Group logo, the Agility Press logo, the Rational Press logo, the Inc.Press logo, Timely Business Books, Rational Guides for a Fast-Paced World, and Custom Corporate Publications are all trademarks or registered trademarks of Mann Publishing Incorporated.

Disclaimer of Warranty

While the publisher and author(s) have taken care to ensure accuracy of the contents of this book, they make no representation or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties or merchantability or fitness for a specific purpose. The advice, strategies, or steps contained herein may not be suitable for your situation. You should consult with a professional where appropriate before utilizing the advice, strategies, or steps contained herein. Neither the publisher nor author(s) shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

Credits

Author:	Anthony T. Mann
Technical Editor:	Jason Carlson
Copy Editor:	Jeff Edman
Series Concept:	Anthony T. Mann
Cover Design:	Marcelo Paiva
Book Layout:	Marcelo Paiva

All Mann Publishing Group books may be purchased at bulk discounts.
Visit us on the web at www.mannpublishing.com or call (877) 877-MANN.

Table of Contents



Part I - Introduction

- Chapter 1 An Overview of SQL Server Reporting Services
- Chapter 2 SQL Server Reporting Services Architecture
- Chapter 3 Designing Effective Reports

Part II - Authoring Reports

- Chapter 4 Data Sources and Data Sets
- Chapter 5 The Report Design Environment
- Chapter 6 Customizing Reports
- Chapter 7 Adding Interactivity

Part III - Managing Reports

- Chapter 8 Report Processing and Management
- Chapter 9 Report Security
- Chapter 10 Server Monitoring
- Chapter 11 Deploying Reports

Part IV - Delivering and Using Reports

- Chapter 12 Report Subscriptions
- Chapter 13 Internet Explorer
- Chapter 14 SharePoint Technologies
- Chapter 15 Microsoft Office

Chapter 1

SAMPLE CHAPTER - COPYRIGHTED MATERIAL

An Overview of SQL Server Reporting Services

SQL Server Reporting Services is a set of innovative components and services that are part of Microsoft's overall Business Intelligence strategy. *Business Intelligence* is, quite simply, the ability to use technology to make better business decisions. These business decisions stem from an understanding of the vast amounts of data existing not only in large companies, but in small to medium-sized companies as well. In the past, many software vendors, including Microsoft, have targeted large companies for their Business Intelligence software. But that's about to change...

Prior approaches to Business Intelligence by leading vendors typically meant that only the largest companies could afford to implement these powerful solutions. For those companies that had the capital to invest in Business Intelligence, only select employees (typically analysts) were empowered to effectively sift through mountains of data to spot trends and patterns. Microsoft's new Business Intelligence strategy is to extend the technology in such a way that knowledge workers, analysts, middle managers, executives, and operations people alike can have access to the data they need to make better business decisions. This strategy is now being targeted at companies of all sizes. However, allowing all employees to use Business Intelligence tools doesn't mean that everyone can view sensitive data that doesn't pertain to them. The strategy is simply to empower users to have the tools they need to make better decisions. Issues relating to reporting security are covered in Chapter 9.

SAMPLE CHAPTER - COPYRIGHTED MATERIAL

To set the stage, you should know a little about prior reporting solutions. In the past, reporting technologies had at least these problems:

- ▶ Reports were difficult to create.
- ▶ Reports were not accessible from other applications.
- ▶ Customization was limited.
- ▶ Long software development cycles made custom reports nearly impossible to attain - especially for small to medium-sized companies.
- ▶ Reports and corresponding data were usually not secure.

SQL Server Reporting Services is a comprehensive reporting platform whereby reports are stored on a centralized web server (or set of servers). Because reports are centralized, users run reports from one place. Having centralized reports also means that report deployment is quite simplified. In addition, because this platform sits on top of the Microsoft .NET Framework, the door is open to unlimited integration possibilities. In fact, all SQL Server Reporting Services is exposed as a set of Application Programming Interfaces (APIs) via Web Services. *Web Services* is an open mechanism that allows applications to integrate by using standard Internet technologies, such HTTP, SOAP, UDDI, and WSDL. Web Services are not just a set of Microsoft standards - they are industry standards.

SQL Server Reporting Services is made available as an add-on product to Microsoft SQL Server, beginning with SQL Server 2000. For more information on licensing and availability, visit the Microsoft web site at www.microsoft.com/sql.

So what's the big deal about SQL Server Reporting Services? Other reporting products have some cool features. Other reporting products run over the web. Some reporting products allow customization and yet others allow programming. All of this is true, but no other reporting solution offers such a comprehensive platform to achieve a best-of-breed solution. What's more is that SQL Server Reporting Services compiles reports into a .NET assembly. A *.NET assembly* is a unit of executable code that is managed by the Microsoft .NET framework to provide optimized services during the process of code execution. The .NET framework is a core component in all Microsoft technologies.



SAMPLE CHAPTER - COPYRIGHTED MATERIAL

Let's See "The Goods!"

Before diving into what reporting services can do, it's important for you to see what a sample report looks like. Since SQL Server Reporting Services make reports centrally available on a web server, running a report from a PC can be done in a web browser such as Microsoft Internet Explorer. When you install SQL Server Reporting Services, a sample Visual Studio .NET project (called **SampleReports** by default) is also installed. Figure 1.1 shows what the sample report, called **Company Sales**, looks like at runtime in Internet Explorer.

The screenshot shows a web browser window titled "Report Manager - Microsoft Internet Explorer". The address bar shows a URL pointing to a report named "Company Sales". The report content includes a header for "Adventure Works 2002 - 2003 Sales" and a table with columns for years (2002, 2003) and quarters (Q1, Q2, Q3, Q4). The table is organized into a tree view where the "Clothing" category is expanded to show sub-categories like Bib-Short, Cap, Gloves, Jersey, Shorts, Socks, Tights, and Vest. The "Bike" and "Accessory" categories are collapsed.

		2002				2003			
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Component		\$246,313	\$497,370	\$2,411,201	\$1,469,913	\$703,539	\$1,500,025	\$5,370,651	\$3,640,919
+ Clothing				\$73,000	\$30,313	\$24,081	\$47,321	\$13,713	\$13,584
-	Bib-Short								
-	Cap	\$1,000	\$1,613	\$4,439	\$5,790	\$4,363	\$3,214	\$46,124	\$34,720
-	Gloves	\$742		\$61,223	\$44,899	\$30,000	\$57,940	\$68,963	\$78,673
-	Jersey	\$10,814	\$19,145	\$68,240	\$40,617	\$24,700	\$54,948	\$388,726	\$300,466
-	Shorts	\$485		\$30,917	\$26,675	\$14,884	\$23,240	\$140,248	\$112,682
-	Socks	\$1,396	\$2,112			\$742	\$2,227	\$30,572	\$26,133
-	Tights	\$742		\$74,930	\$61,904	\$33,055	\$56,269	\$24,535	\$20,207
-	Vest			\$2,215	\$960		\$2,430	\$102,305	\$94,784
+ Bike		\$4,688,606	\$5,334,309	\$8,160,133	\$6,888,508	\$5,758,684	\$6,565,628	\$11,798,192	\$10,249,706
+ Accessory		\$5,399	\$11,648	\$49,778	\$40,860	\$33,066	\$56,700	\$559,179	\$522,980

Figure 1.1 : Sample Company Sales Report for Adventure Works.

Figure 1.1 shows a sample report that is quite interactive. In other words, the sample **Company Sales** report allows you to drill up and down (by clicking the + and - links) to see more or less data detail. However, reports can be interactive in other ways as well. For example, you can search for specific data in the results, or even configure the report to supply filter criteria to limit data displayed on the report. An example of a filtered report would be one that displays sales only for the **Clothing** category. Adding interactivity to your reports is covered in Chapter 7.



SAMPLE CHAPTER - COPYRIGHTED MATERIAL

Reporting Lifecycle

To provide true value, a SQL Server Reporting Services report is taken through an entire process, known as the reporting lifecycle. The *reporting lifecycle* is comprised of three separate activities:

- ▶ **Report Authoring** — The process of defining the report itself, report properties, user interactivity, and “look-and-feel.” Report Authoring is covered in Part II of this book.
- ▶ **Report Management** — Activities centered on the administration of published report. Such activities include determining when reports are refreshed, who has access to those reports, and more. Report Management is covered in Part III of this book.
- ▶ **Report Delivery** — Activities focused on delivering reports to the end-users. You can specify the trigger for the delivery process (such as an event or user action) and the device on which the report shall be rendered (such as a PC browser, PDA, cell phone, or other mobile device). Report Delivery is covered in Part IV of this book.

Report Definition

The definition of each report is specified in XML and is stored in a file with an RDL file extension (which stands for Report Definition Language). RDL files can be generated using Visual Studio .NET 2003, or even your favorite text editor. After all, RDL files are simply XML, which is a text-based language. Therefore, any tool which can read or write XML will work fine. On the other hand, Visual Studio .NET 2003 has additional capabilities to allow you to design, debug, and test your reports from within a single integrated environment, so it’s a good idea to use this tool to create and edit RDL files.

Defining a report generally includes these activities:

- ▶ Specifying one or more data sources that contain the data. SQL Server Reporting Services can access information from data sources, such as Microsoft SQL Server, OLE DB, ODBC, and Oracle. Data sources are covered in Chapter 4.



SAMPLE CHAPTER - COPYRIGHTED MATERIAL

- ▶ Indicating which queries or stored procedures are used to retrieve data from a data source. The results of these queries are made available in a structure called a data set, which is also covered in Chapter 4.
- ▶ Placing graphical objects and elements on the report to show data, such as charts, graphs, and tables. There's even a special type of object, called a matrix. A *matrix* is a special object that is similar to a pivot table, which allows columns and rows to expand dynamically. Defining reports using these graphical elements is discussed in Chapter 5.
- ▶ Defining report options and criteria, such as parameters, sorting, grouping, and filtering. Customizing reports is shown in Chapter 6.
- ▶ Specifying security options, such as which users can access reports and what data they can see in those reports. Security of reports and data is covered in Chapter 9.

In Visual Studio .NET a single report is part of a larger project containing a set of reports that are developed together. Therefore, a reporting project likely contains multiple reports. One or more projects are contained within a solution. This follows the same concept as any other development project within Visual Studio .NET. To illustrate the file-to-project relationship, see Figure 1.2, which shows the same **Company Sales** report shown in Figure 1.1, but in the design-time environment of Visual Studio .NET 2003.

SAMPLE CHAPTER - COPYRIGHTED MATERIAL

is the limit! All fields, report options, features, security settings, and more can be accessed and programmed using a rich set of objects available at runtime on the report server. You can even determine programmatically which server in a Web farm the report is running on. A *Web farm* is a series of servers that work together to spread incoming HTTP requests so that no single server reaches its processing limit. There are also statistics about the reports that you can access and analyze, such as how long your reports take to run, which reports are run most often, and so forth. Programmability of SQL Server Reporting Services is shown in Chapter 6.

Integration

Because of its native support for Web Services and related technologies, SQL Server Reporting Services supports tight integration with these commonly-used Microsoft products:

- ▶ Microsoft Internet Explorer
- ▶ Microsoft Office
- ▶ Microsoft SharePoint Portal Server
- ▶ Microsoft Windows SharePoint Services

SQL Server Reporting Services is powerful enough to develop reporting solutions that integrate with other popular Microsoft products right out of the box. Imagine a scenario where you create a reporting portal for use by your company to empower all knowledge workers to run reports from within the context of the portal or dashboard. For instance, you could configure a SharePoint Portal Server to show sales professionals new reports of their own personal sales every Monday morning. SharePoint Portal Server would cooperate with the reporting server to ensure that only the currently logged-in user could see the sensitive data. The possibilities are limitless! Integration with other products is discussed in Part IV of this book.

SAMPLE CHAPTER - COPYRIGHTED MATERIAL

Deployment

Deployment with SQL Server Reporting Services is quite straightforward, since just as every .NET application is simple to deploy. Once the RDL files that comprise your reporting project(s) are written, they are compiled and deployed on the target reporting server. Deployment is covered in Chapter 11.

Security

The type of authentication supported by a requesting device dictates how the reporting server needs to handle security. Because every company handles security in different ways, multiple authentication schemes are available:

- ▶ **Basic Authentication** — Authenticates a user, but passwords are transmitted in clear text across a network. Credentials are not automatically encrypted.
- ▶ **NTLM** — Securely authenticates a user, following the encryption algorithm designed for Windows NT.

Tech Tip:

NTLM is formerly known as Challenge/Response authentication.

- ▶ **Passport** — Global centralized secure authentication scheme handled by Microsoft's Passport service.
- ▶ **Kerberos** — Secure authentication mechanism provided in Windows 2000 and later operating systems. Kerberos is harder to hack than NTLM and has support for strong passwords. This authentication method is recommended over NTLM when possible.

SQL Server Reporting Services follows a role-based security model. *Role-based security* is a model whereby a user login is associated with one or more roles (which usually mimic job functions). A role is associated with one or more privileges within the reporting system. Therefore, a single user login will likely



SAMPLE CHAPTER - COPYRIGHTED MATERIAL

have multiple privileges. For example, the system administrator role is allowed all privileges and can therefore perform all actions. On the other hand, a sales person might only have the privilege to view his/her own sales reports, but not a co-worker's sales reports. You can use the pre-installed security roles or create your own. Security and roles are covered further in Chapter 9. The following roles are automatically configured when you install SQL Server Reporting Services:

- ▶ **Browser** — Allows a user to view, but not change reports.
- ▶ **Content Manager** — Allows a user to manage the content of published reports.
- ▶ **My Reports** — Allows a user to manage all aspects of reports located in their personal **My Reports** folder.
- ▶ **Publisher** — Allows a user to publish reports to the report server.

Delivery

Delivery refers to the ability of SQL Server Reporting Services to render a report in a particular format and send it to the target location. Reports can be generated in any of the following formats:

- ▶ **HTML** (for web pages and other HTML compatible programs, both HTML 3.2 and 4.0)
- ▶ **HTML with the Office Web Components**
- ▶ **Web Archive file**
- ▶ **CSV** (comma separated values, or any delimiter)
- ▶ **TIFF File** (as a printable snapshot image, or any other image format, BMP, JPG, PNG, WMF)
- ▶ **Excel Document** (if using Office XP or later)
- ▶ **XML** (eXtensible Markup Language)
- ▶ **PDF File** (for using Adobe Acrobat)

SAMPLE CHAPTER - COPYRIGHTED MATERIAL

In addition to the output format of the reports, you can also indicate the timing and location of the report delivery in a subscription. In other words, subscriptions can be generated under the following conditions:

- ▶ **On Demand** — Reports are rendered when the user requests a report to be run. Large reports might take considerable time to run, so you have the option of configuring reports to use a cached (or snapshot) copy of the report. Using cached data speeds up report querying because the report does not have to process data every time a user requests the report.
- ▶ **Simple Subscription** — Reports are generated upon certain events (such as data changing) or time schedules (such as every Sunday night) and pushed to a client device or computer from the reporting server.
- ▶ **Data-driven Subscription** — Reports are generated automatically, just as in a push subscription, but also allows a query to be written and applied to the delivery of a report at run-time. This allows you to create subscriptions to reports based on parameters or recipients that are not known at the time the subscription is created.

Reports can be delivered to a list of recipients via e-mail or placed on a network file share. Additional delivery options can be customized by writing your own delivery extensions. Subscriptions are covered in Chapter 12.

SAMPLE CHAPTER - COPYRIGHTED MATERIAL

Editions

SQL Server Reporting Services is available from Microsoft in these editions:

- ▶ **Standard** — Includes basic features, such as:
 - Report rendering and delivery
 - Simple (push) subscriptions
 - Role-based security
 - Exporting to multiple file formats
 - Support for up to 2GB RAM
 - Support for up to 4 processors
- ▶ **Enterprise** — Includes all Standard Edition features, plus:
 - Data-driven subscriptions
 - Web farm support
 - Custom authentication
 - Support for more than 2GB RAM
 - Support for more than 4 processors
- ▶ **Developer** — Same as the Enterprise Edition, but will also install on Windows XP. The Developer Edition cannot be deployed onto a production server.
- ▶ **Evaluation** — Same as the Enterprise Edition, but will expire after the trial period ends.

SAMPLE CHAPTER - COPYRIGHTED MATERIAL

What Can SQL Server Reporting Services Do?

Starting with SQL Server 2000, SQL Server Reporting Services extends Microsoft's Business Intelligence platform to allow control over most aspects of your reports, such as:

- ▶ Report Definition
- ▶ Programmability
- ▶ Integration
- ▶ Deployment
- ▶ Security
- ▶ Delivery

Summary

SQL Server Reporting Services is a revolutionary new technology that allows you to gain complete control over all your reporting needs. You can feel confident that all of your data is stored securely and will be safe from unauthorized users. Finally, reports can be rendered in a variety of popular formats and displayed on PCs, PDAs, cell phones, and other mobile devices.

The rest of this book explores these topics in greater depth. You'll learn how to create, manage, deploy, and use reports created with SQL Server Reporting Services. If you're ready to dive in, let's begin...

