

SQL Server 数据库经典译丛

SQL Server 2012 Reporting Services 高级教程

(第 2 版)

	Paul Turley	
	Robert Bruckner	
[美]	Thiago Silva	著
	Ken Withee	
	Grant Paisley	
	颜 炯	译

清华大学出版社

北 京

Paul Turley, Robert Bruckner, Thiago Silva, Ken Withee, Grant Paisley
Professional Microsoft SQL Server 2012 Reporting Services
EISBN: 978-1-118-10111-7
Copyright © 2012 by John Wiley & Sons, Inc., Indianapolis, Indiana
All Rights Reserved. This translation published under License.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2012-7862

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

SQL Server 2012 Reporting Services 高级教程(第 2 版) / (美) 特里(Turley, P.) 等著; 颜炯 译.
—北京: 清华大学出版社, 2014
(SQL Server 数据库经典译丛)
书名原文: Professional Microsoft SQL Server 2012 Reporting Services
ISBN 978-7-302-34268-7
I. ①S… II. ①特… ②颜… III. ①关系数据库系统—教材 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2013)第 249400 号

责任编辑: 王 军 韩宏志
装帧设计: 牛静敏
责任校对: 曹 阳
责任印制:

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>
地 址: 北京清华大学学研大厦 A 座 邮 编: 100084
社 总 机: 010-62770175 邮 购: 010-62786544
投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn
质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:

装 订 者:

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 44.25

字 数: 1188 千字

版 次: 2014 年 1 月第 2 版

印 次: 2014 年 1 月第 1 次印刷

印 数: 1~4000

定 价: 98.00 元

产品编号:

译者序

SQL Server 2012 Reporting Services 是微软公司开发的报表设计和企业级信息发布工具，从出现至今已经 9 年了。Reporting Services 原先的任务是为开发人员和 IT 专业人士实现一个一流的报表开发平台，而现在的 Reporting Services 已经能够帮助普通的业务人员开发一流的报表解决方案了。Reporting Services 针对商业智能和报表编写提供了大量工具，其功能之强大，令人惊叹不已。

本书基于作者在 SQL Server 2012 Reporting Services 及其他报表工具中积累的经验，从基础知识讲起，深入介绍了大量与报表创作和报表解决方案开发有关的知识，占用大量篇幅介绍如何使用 SQL Server 2012 Reporting Services 来解决多个不同领域中的业务问题。

这样一部综合性指南书籍既可为报表设计新手提供广泛的入门知识，引领他们成长为熟练的报表设计者，也为那些打算学习高级报表功能的程序设计专业人员提供了报表编程和报表集成知识，同时本书还详细介绍了报表服务器管理和安全性。本书的重点之一是针对 SQL Server 2012 Reporting Services 引入的多项 BI 创新(例如语义表格模型和 Power View 可视化工具)以及如何在报表中应用这些创新进行深入讲解。因此，本书不仅可作为普通报表设计人员的入门教程，也可以作为报表应用程序开发人员、系统工程师或 DBA、商业智能分析师、商业机构管理者的教程和参考手册。

本书不仅涵盖了 SQL Server 2012 Reporting Services 报表设计的全部内容，而且提供了大量报表和程序示例，不仅可作为培训和自学教程，还可作为报表程序员的程序开发参考手册。无论是初学者还是具有一定基础的读者，我们都建议下载本书附带的代码和报表示例，自己搭建一个小型教学平台，认真学习本书列举的示例，这样才能进一步深入理解作者的意图，体会到作者从多年实践中获取的真知灼见。

本书全部章节由颜炯翻译，参与本书翻译活动的还有肖国尊、李辉、胡季红、鲍春雷、王继云、高于博、刘鹏、朱宝庆、赵建军、胡海涛、龚琴、李忠鹏、郑亮亮、王佩佩、陈嘉耀。译者在翻译过程中，尽量保持原书的特色，并对书中的词句进行了仔细推敲和研究。但毕竟有少量技术是译者在自己的研究领域中所不曾遇到过的，所以疏漏和争议之处在所难免，如果读者在阅读过程中有任何意见或建议，请将反馈信息发送到 wkservice@vip.163.com，我将不胜感激。

颜 炯

作者简介

Paul Turley 是一位 SolidQ 讲师，也是一位 Microsoft MVP，同时还是一位解决方案架构师、教师和宣讲员。他常常在行业会议上发表演说，还编写了多部书籍，内容涵盖 BI、数据可视化和报表。他的博客是 SQLServerBIBlog.com。自 1991 年以来，他就在为微软、迪士尼、耐克、惠普这样的公司开发业务数据库解决方案。1996 年他成为微软认证教师，拥有多项行业认证证书，包括 MCTS and MCITP for BI、MCSD、MCDBA、MSF Practitioner 以及 IT Project+。

Paul 曾经独立或合作撰写了多部书籍和教程，内容涵盖数据库、商业智能以及应用程序开发技术。他的著作包括：*SQL Server Reporting Services Recipes for Designing Expert Reports*、*Professional SQL Server 2008 Reporting Services*、*Professional SQL Server 2005 Reporting Services*、*Professional SQL Server Reporting Services (SQL Server 2000)*、*Beginning T-SQL with SQL Server 2005 and 2008*、*Beginning Transact-SQL with SQL Server 2000 and 2005*、*Beginning SQL Server 2005 Administration*、*Beginning Access 2002 VBA*、*Data Warehousing with SQL Server 2000 Analysis Services* 以及 *Professional Access 2000 Programming*，以上书籍均由 Wrox 出版社出版。同时，他还作为主笔，撰写了 *SQL Server 2005 Integration Services Step by Step* 一书，由微软公司出版社出版。

Robert Bruckner 是 Microsoft SQL Server 部门一位主管软件架构师和开发人员。Robert 负责 SQL Server Reporting Services 的技术架构，包括 Power View 的技术架构。Robert 负责的核心领域之一是可扩展报表处理引擎的设计和开发，该引擎在 Reporting Services 和 Power View 中使用。Power View 是 Reporting Services 2012 提供的一项增强功能，它加强了终端用户支持，使终端用户能够方便地以交互方式查看数据，迅速获得深入的分析结果，使得数据分析研究成为一种乐趣！

Robert 于 2003 年加盟微软公司，此前他在 T-Mobile Austria，其间他作为一位系统架构师，从事研究、设计和实现数据库和商业智能系统的工作。Robert 还是奥地利维也纳科技大学的研究人员。Robert 以优异的成绩从维也纳科技大学计算机专业获得了硕士和博士学位，并且拥有数项专利。

如果你对搜索引擎感兴趣，那么可以在公共新闻组和 MSDN 论坛找到上千篇 Robert 发表的帖子，从而共享 Robert 在报表服务和 SQL Server 技术方面的见解、窍门、技巧和专家建议。Robert 在 SQL Server Reporting Services 和 Analysis Services 方面有多部合著作品。Robert 定期出现在行业会议上，同时还维护着一个很受欢迎的博客，其网址为 <http://blogs.msdn.com/robertbruckner>。闲暇时，Robert 喜欢从事山地自行车运行，还喜欢滑水和阅读。

Thiago Silva 是一位 MCPD，同时还担任 Credera 的架构师和咨询师。在 .NET 早期和 SQL Server 2000 时代，Thiago 就在从事 .NET 和 Reporting Services 的定制化开发工作。他是 Credera 公司(该公司位于达拉斯)的微软公司实践者，负责为包括世界 500 强在内的客户提供微软公司解决方案。最近 8 年来他一直从事咨询工作，此前，他是一位软件工程师和 Web 程序开发人员。Thiago 是本书前一版本，即 *Professional SQL Server 2008 Reporting Services* 一书的合著者，他还参与撰写 *Microsoft SQL Server Reporting Recipes for Designing Expert Reports* 一书。Thiago 凭借技术播客 “.NET Rocks!” 获得美誉，同时还是 DFW .NET 用户组合社团成员。他偶尔会在自己的技术博客 “Silvaware” 上发表文章，该博客网址为 <http://silvaware.net>。Thiago 拥有 Texas A&M University 信息系统专业信息及

运筹管理的 BBA 学位，并因使用 ASP.NET 2、ASP.NET 3.5 和 ASP.NET 4 进行 Web 开发而获得了 MCAD、MCPD 与 MCTS 头衔。

Ken Withee 是 Portal Integrators LLC 公司(www.portalint.com)的总裁，该公司是一个软件开发公司，专注于 SharePoint 平台开发世界级业务应用程序。他和妻子 Rosemarie 居住在华盛顿州的西雅图，并且是多部书籍的作者或合著者，这些书籍包括：*Microsoft Office 365 For Dummies* (Wiley, 2011)、*SharePoint 2010 Development For Dummies* (Wiley, 2011)、*Professional Microsoft Project Server 2010* (Wrox, 2012)、*Microsoft Business Intelligence For Dummies* (Wiley, 2010)、*Professional Microsoft SQL Server 2012 Reporting Services* (Wrox, 2012)与 *Professional Microsoft SQL Server 2008 Reporting Services* (Wrox, 2008)。Ken 还在多本刊物和杂志上发表了多部作品。

Ken 曾在旧金山州立大学师从 Edward Land 博士处获得了计算机科学理学硕士学位，其研究工作成果曾经发表在 LNCS 学报上，并且在 Phoenix 召开的 IASTED 会议上进行了宣读。他们的研究工作还曾在其他人机交互国际会议上进行了宣读。

Ken 拥有 12 年以上的专业计算机和管理经验，从事过多种技术。他是一位微软认证技术专家，并且在 SharePoint、SQL Server 和 .NET 领域通过了认证。

Grant Paisley 是一位 SQL Server MVP，并且是 Angry Koala 公司的创立者，该公司位于澳大利亚的悉尼，是一家微软商业智能咨询机构。Grant 是 SQL Server 悉尼用户组的负责人，还是一位国际知名的演讲家，曾在美国 TechEd 发表过演讲，还曾在澳大利亚发表过演讲，他甚至还在中国发表过演讲。他在商业智能方面的兴趣主要集中在数据可视化领域，并为此创建了 <http://reportsurfer.com>，这是一个报表网站。Grant 还是 *SQL Server 2008 Analysis Services with MD* 和 *Microsoft SQL Server Reporting Services Recipes* 的特约作者之一。如果你没有在舞台上看到他，那么你可能会在夏威夷水上风筝冲浪或在 Whistler 山地自行车赛上看到他。

致 谢

感谢我的幸福家庭，因为他们对我的工作给予了持之以恒的支持，偶尔还需要忍受我对书籍、论文、项目和会议的过分投入。感谢妻子 Sherri，因为她说过：“宝贝，我爱你，但是假如你将任何与会议有关的东西带回家，那你就去车库睡觉吧。”

感谢微软公司 Reporting Services 和 SQL Server BI 产品团队。Thierry、Sean、Carolyn、Lukasz、Ariel 与 Robert，还有许多其他的人，感谢你们在过去的 9 年中为书籍、项目与支持所展示的开放思维和提供的帮助。你们将这些美妙的技术提供给大众，使他们能够在世界范围内利用这些技术发布信息，并且取得了重要的成果，感谢你们，因为你们让我在你们的工作中发挥了一点点作用。感谢 Mark、Chuck、Denny、Carl 与 John，还要感谢 SQL CAT 团队。

感谢微软公司的 MVP 机构，因为这个机构为立志奉献的专业人士建立了一个难以置信的网络。感谢 SQL Server PASS 组织，因为它培养了一个令人尊敬和信服的社区。感谢令我感受到家人般温暖的 Arnie 和 Oregon SQL 团队。感谢所有的 PASS 章节指导者和 SQL Saturday 组织者。

感谢 SolidQ 全体成员建立了一个与众不同的优秀组织。我以自己能够与如此之多的值得信任的朋友和专业人士相提并论而感到自豪。

衷心感谢 Wiley 出版社的编辑和管理团队，特别是 Bob 和 Kelly。我真的无法理解你们到底是如何成功管理这些在闲暇时候撰写书籍的作者们的。感谢你们一直以来的耐心和坚持。感谢本书的其他作者和评审人员：Robert、Ken、Grant、Thiago、Joe、Chris、Nigel 与 Glyn，他们都花费了很多的夜晚和周末的时间不断地评审和重写。我们只需再进行一次修改就能够结束了，兄弟们！为了本书的杀青，感谢你们所有人。

我对那些编写了“其他”书籍的人们充满了深切的敬意，这些人是充满生机的行业的观察者和贡献者。感谢 Stacia、Teo 与 Brian，你们为行业建立了一个很高的标准。

— Paul Turley

Robert 希望对 Paul Turley 和 Bob Elliott 致以特别谢意，因为他们在这个项目中自始至终都提供了极好的合作氛围；他们从几位经验丰富的 Reporting Services 专家那里吸取经验，从部署在许多用户那里的 Reporting Services 中收集了证明为行之有效的最佳实践。此外，Robert 还希望对本书所有合著者表达诚挚的谢意，并感谢 Kelly Talbot 的卓越编辑成果，还要感谢技术评审人员，他们保证了内容的准确性。

— Robert Bruckner

感谢 Thierry D’Hers、Robert Bruckner 以及 Reporting Services 产品团队，感谢他们在写作本书过程中提供的指导和技术帮助。

感谢 Paul Turley 为我提供了机会参与编写本书的项目，感谢 Kelly Talbot 和 Bob Elliot 在写作和编辑过程中的支持和耐心。

最后，感谢我的妻子和孩子，他们对我写作本书所花费的漫长时间和耗费的周末给予了宽容。我感谢他们的爱心和鼓励，没有他们的爱心和鼓励，我不可能完成我的工作。

— Thiago Silva

我希望感谢我的祖母Tiny Withee，她今年已是 99 岁高龄，仍然十分健康。我还希望感谢我的妻子 Rosemarie Withee，母亲 Maggie Blair，父亲 Ken Withee，姐姐 Kate Henneinke，以及岳父和岳母 Alfonso Supetran和Lourdes Supetran，还要感谢全体家庭成员。特别真诚感谢合著者 Paul Turley、Robert Bruckner、Thiago Silva 还有 Grant Paisley。特别感谢 Bob Elliott、Kelly Talbot、Gayle Johnson、Joe Salvatore、Chris Albrektson、Nigel Sammy 以及 Wrox 出版社使本书问世的其他所有成员。

— Ken Withee

感谢 Angry Koala 团队，当我忙于其他工作的时候，Glyn Llewellyn 接手了我的工作，并帮我撰写了一些章节。感谢 Colin McGowan、David Lean、Geoff Orr、Mark Fitzpatrick、Lesley Llewellyn、Peter Orgill 与 Praveen Chand，他们在 BI 方面的专业水平和热情是我成功的支柱。感谢好友 Paul Turley，他为我提供了机会，并且在撰写本书过程中提供了支持。Kelly 和 Wrox 出版社的团队在本书出版过程中发挥了重要作用。最后还要感谢我挚爱的家庭成员 Sue、Megan、Lisa 与 Zoe(但我很少告诉他们这一点)。嗯，我差点忘了：“Megan 真是太棒了”——其实你们都很棒！

— Grant Paisley

序 言

十五年前，也就是 1997 年 1 月，微软公司收购了我的公司(Panorama Software 公司)。当时微软公司正在进入商业智能领域。这已经是很久以前的事了。实际上，这件事实在是过于久远，我都不能确定那个时候商业智能是不是被称为“BI”。回想那个时候，我们的任务和策略也很简单，那时我们提出了一个简单而有效的口号：“大众化的 OLAP”。这么多年过去，这项任务的主要内容并未发生变化。我们仅仅是关注了产品的另一个方面，即“大众化”。

早期我们关注的内容和任务就是确保我们的技术易于使用，不需要过多的咨询工作(相对于当时市场需求而言)，还有就是廉价。我们的思路是让尽可能多的公司使用这项技术。策略的另一方面是为解决方案提供一工作平台，并且围绕我们的 BI 产品构建一个健康的生态系统。

确实，我们很早就开始构建一个具有强大的功能、丰富的 API 集合和查询语言(MDX)的服务器，我们还构建了一个欣欣向荣的合作伙伴生态系统。

成果很快就完成了。SQL OLAP Server 成为当时世界上最为广泛使用的 OLAP 服务器，SQL OLAP Server 后来被称为 SQL Server Analysis Services。现在，这个产品仍然是世界上最为广泛使用的 OLAP 服务器。

因为我们发现市场接纳了 Analysis Services 服务器并反映良好，所以我们开始着手第二项工作。针对报表解决方案，我们提供了具有创新性的新产品，即 SQL Server Reporting Services。SQL Server Reporting Services 得到的广泛认可超出了我们的预期。一夜之间，在我们的用户群中，增加了上千万的报表用户，这些用户需要从自己的 IT 部门获取报表，并根据报表内容做出判断。就这样，我们在大众化方面前进了一大步，而大众化正是我们追求的目标。

随着越来越多的公司开始使用 BI 解决方案，用户人群的复杂性随之增长。我们的用户突然领悟到 BI 提供的强大功能与价值，因此他们对 BI 的要求也越来越高。然而，他们发现自己的 IT 部门并未满足其需求，这成为一个瓶颈。

我们发现，这个问题一方面成为进一步持续扩大提供“大众化的 BI”的障碍；另一方面，也成为进一步持续扩大提供“大众化的 BI”的机会，因此我们开发了一组工具。2010 年，我们发布了 PowerPivot，随后我们于 2012 年发布了 Power View。这些工具可以帮助信息工作者构建自己的 BI 解决方案，为此使用了我们所说的“受托管的自助式 BI(managed self-service BI)”。自助的概念很好理解：构建一个工具，任何一个访问数据的信息工作者都可以使用这个工具，这样他就可以构建自己的 BI 解决方案。托管的概念更为基础。即使我们希望为最终用户提供更为强大的工具，IT 部门也希望能够以一种受控且安全的方式完成这项工作。IT 部门希望确保能够备份所有关键数据和应用程序，保证数据持续更新，并且能够掌握都有哪些人使用了数据，以及这些人是如何使用数据的。

由于原先的任务是为开发人员和 IT 专业人士开发一个一流的平台，而现在的任务是为信息工作者开发一个一流的解决方案，因此这种变化极大地影响了如何设计和构建软件。我们需要简化提供的内容，减少用户必须掌握的技术知识。这需要我们在智能化的默认设置方面进行大量投入，这样用户就无须为学习产品而进行培训。因此，这就需要产品在交互性能方面提高到一个新的层次，产品需要具有即刻响应的能力，因为用户总是缺乏耐心。我们假设用户总是要使用自己的数据集进行

试验。这一点与一个专业的 BI 项目不同。专业的 BI 项目提供了一个具有良好定义的需求集合，开发人员必须满足这些需求。但是在自助式的场景下，用户只需使用数据进行试验，而后即可自行得到结论。这就让我们得到了一个最为重要的设计理念：趣味性。为了让用户花时间依靠自己获得有用的结论，唯一的方法就是他对数据感兴趣。

针对本书，我真正喜欢的一点就是它从 Reporting Services 产品线的角度，完美地反映了微软公司的 BI 发展历程，并且是从以托管(managed)和运营(operational)为特点的早期阶段开始，一直到最新的自助式报表，都有所介绍。本书是由构建了这项产品的专家编写的，并且是从客户角度出发进行编写的。他们在商业智能方面的工作年限累计总共超过 40 年。

我将本书称为 SQL Server Reporting Services 的百科全书。所有章节都根据最新的内容进行了更新，以确保其准确性，并且能够与最新版本的 SQL Server 2012 兼容。本书包括了一个新的内容，它说明了如何使用Power View和PowerPivot。你将学到组合使用这两个工具的最佳方法，从而可以在SharePoint 中使用 Reporting Services 提供的最新托管功能。

我希望你以阅读本书为乐，并且能够像我们(也就是产品团队)乐于构建 SQL Server Reporting Services 2012 那样，乐于使用 SQL Server Reporting Services 2012。

——Ariel Netz

Microsoft SQL Server Reporting Services 合作伙伴组计划经理

前言

Reporting Services 是一个报表设计和企业级信息发布工具，这个工具从出现至今已经有 9 年了。本书也随着这个产品一起不断更新。我们早在 2003 年这项产品发布之前就开始撰写 *Professional Microsoft SQL Server Reporting Services* 一书。在 Reporting Services 发布了 4 个新的版本之后，本书也已经出了 4 版。

基于我们在 SSRS 及其他报表工具中所获得的经验，我们学习了大量与报表创作和报表解决方案开发有关的知识。我们将在本书中分享这些经验。我们不仅要向你展示如何使用这项本来就是用于设计报表的技术和相关工具来设计报表，还将分享我们在使用这项技术和相关工具过程中学到的最佳实践和教训。作者花费了大量时间使用这些工具来解决多个不同行业和业务领域中遇到的业务问题。

本书读者对象

某些书籍是专门为初学者撰写的，某些书籍是为认真的开发人员和高级报表设计人员编写的。我们的观点是，在不牺牲少数人需求的前提下，要强调满足大多数人的需要。我们打算撰写一部能够满足广大读者需求的著作，同时也要满足报表设计人员、开发人员、管理人员和业务专家的特殊需求。

对本书而言，我们的目标是让本书成为一部综合性指南和一部极具价值的参考书。为了达到这个目标，我们将本书分为不同的部分，如果你愿意的话，每个不同部分都可以是一本书，只是规模较小。你可以根据自己的需要，针对某个部分花费较多的时间和精力，而将其他某些部分作为参考书来使用。本书是为报表设计新手撰写的，同时也适用于那些打算学习高级报表功能的读者。对应用程序开发人员而言，我们还介绍了报表编程和如何通过编程将报表集成到自定义的应用程序中。你还将学习到报表服务器管理以及与安全性的内容。

我们不仅知道都有哪些人士以各种不同方式使用 Reporting Services，还掌握了其他更多内容。下面各节描述了具有不同角色的人群，还描述了本书是如何满足不同人群的需求的。我们为这些代表不同人群的人取了名字，希望使示例更为真实。将这些示例作为指南，你不仅可以找到能够与你的工作具有最佳匹配度的角色，还可以知道你使用 Reporting Services 的方式。

报表设计人员

Mary 在一个为消费者提供服务的公司的财经部门工作。Mary 是一位熟练的计算机用户，掌握多项办公技能。她已经在此集团工作多年，可以轻松完成老板交给的工作。她熟悉公司的业务过程、财经报表流程，也熟悉票据和计费系统。她不是计算机天才，但了解文字处理、电子表格、电子邮件，还了解简单的数据库报表。Mary 好几年前就开始使用 Microsoft Access，根据从 HR 及客户计

费系统导出的数据，她可以使用向导创建一些简单报表。过了一段时间，她学习了如何编写查询语句和构建 Access 报表，并且可以在不使用向导的情况下，自行定义报表格式、进行分组和汇总。她已经使用过其他报表工具，并且能够基于公司数据仓库中的数据和某些运营数据库生成报表。她已经会使用图表和透视图来设计报表并分析销售趋势和盈利情况。她所在公司最近要对 Reporting Services 的使用进行标准化。

Mary 关注的核心内容是直接可以使用的报表系统，也就是说，她需要使用产品中提供的简便易用的工具来设计和部署报表，而且要尽可能的简单。她可能需要设计基于服务器的标准报表，其他用户需要通过公司的内部网络从一个中心报表服务器访问这些报表。她可能还需要根据管理员或其他高级设计人员创建的数据模型，创建自己特定的客户端报表。

Mary 对本书中的以下部分内容最感兴趣：

- 第 I 部分：入门
- 第 II 部分：报表设计
- 第 III 部分：设计商业智能报表
- 第 IV 部分：启用用户报表

应用程序开发人员

Joe 已经从事了数年数据库应用程序开发工作。几年前他开始使用 Microsoft .NET 编程工具，并且在公司的信息技术部门获得了一个编程职位。Joe 使用 Visual Basic .NET 和 C# 编程语言为公司设计了多个网站和门户。Joe 所创建的大多数报表都是从头开始编写的，就像开发自定义网页那样。他曾用过几种报表应用程序，希望为公司的定制业务应用程序添加报表功能。

就 Joe 看来，编写简单报表是其他人的事，他更为关注的是如何向一个解决方案中添加功能，而不是处理业务问题和与报表美观有关的质量内容。他关注的是使用程序代码和查询脚本来添加筛选、自定义格式化与条件逻辑功能。他还要设计自己的报表，从而使之能够嵌入到应用程序中，成为一个解决方案的集成部分。他还希望创建自定义的管理工具，从而能够完成自动化报表服务器日常维护工作。

Joe 知道，为将报表集成到不同的应用程序接口中，Reporting Services 提供了多种灵活的选项。他不仅需要将报表嵌入到一个自行开发的 Windows 桌面应用程序中，还需要将其嵌入到 Web 应用程序、SharePoint 门户以及移动设备应用程序中。

Joe 对以下部分内容最感兴趣：

- 第 II 部分：报表设计
- 第 III 部分：设计商业智能报表
- 第 V 部分：解决方案模式
- 第 VII 部分：Reporting Services 自定义编程

系统工程师或 DBA

Bob 是一位网络工程师和数据库管理员。与报表的美观和功能相比，他更关注公司服务器的安

全和稳定。他希望确保为报表管理人员、设计人员、开发人员和用户分配不同的角色，并保证报表服务器的安全。Bob 需要安装报表服务器和使用不同选项来配置报表服务器。他还需要安排维护任务计划，优化数据库及查询，并提供不间断的维护和灾难恢复：

Bob 认为以下部分内容对他来说最有意义：

- 第 I 部分：入门
- 第 V 部分：解决方案模式
- 第 VI 部分：管理 Reporting Services

业务领导

作为业务人员、公司经理或项目经理，你既可以是报表解决方案的消费者，也可以是开发工作的指导者。你可能采用了某个商业智能咨询公司的服务来构建一个决策支持系统，并用来协助运行公司的业务。你需要了解可以使用的选项，掌握产品和用来创建解决方案的技术概念。本书将帮助你理解这些特性，并将解释将其应用于实践所需的选项。解决方案的实现人员将向你询问业务需求和需要实现的功能。第 1 章和第 4 章可以作为很好的入门教材。你可能对本书第 III 部分“设计商业智能报表”很感兴趣，因为这部分内容能够为应用于业务分析的报表场景类型提供指导。第 V 部分，“解决方案模式”，讨论了如何定义和管理报表解决方案、业务需求和规范。这部分内容可以作为你、业务用户与报表设计人员之间的沟通媒介。

本书涵盖的内容

本书分为 7 个部分：

- 第 I 部分：入门
- 第 II 部分：报表设计
- 第 III 部分：设计商业智能报表
- 第 IV 部分：启用用户报表
- 第 V 部分：解决方案模式
- 第 VI 部分：管理 Reporting Services
- 第 VII 部分：Reporting Services 自定义编程

第 I 部分：入门

第 1 章“Reporting Services 介绍”描述了 Reporting Services 的功能和特性。你将学习 Reporting Services 的可扩展架构，正是这种可扩展架构使得 Reporting Services 能够对几乎现有的全部业务系统进行强大而又灵活的扩展。该章是后面学习设计、部署、管理商业智能和报表解决方案的基础，也可能是定制化商业智能和报表解决方案的基础。你将学习报表用户配置、报表应用程序和解决方案类型。该章还介绍了为 SQL Server 2012 新开发的报表设计工具。包括 SQL Server Data Tools 报表设计器和报表生成器。

第 2 章“Reporting Services 安装和架构”将帮助你安装 Reporting Services，还解释了一个完整的报表解决方案的架构。该章将一步步地指导你建立自己的开发环境和规划企业部署。该章还将帮助你理解 Reporting Services 使用的核心服务和技术。通过学习产品如何工作和组件如何交互，可为设计和维护一个可扩展解决方案打下更好的基础。

第 3 章“配置 SharePoint 集成”可以用来集成 Reporting Services 与 SharePoint。你将学习到如何安装和配置所有集成的组件。如果你不打算在 SharePoint 集成模式下使用 Reporting Services，而是打算在 Native 部署模式下使用 Reporting Services，那么可以略过该章。

第 II 部分：报表设计

设计报表既可以是一个通过运行向导即可轻松完成的过程，也可以是一个定义了高级特性的、高度复杂的开发过程。你将学习用报表如何处理和填充数据，还要学习如何使用参数和表达式来定义具有创新性的报表解决方案。

第 4 章“基本报表设计”首先介绍了基础知识，讲解了如何使用简单的设计工具创建基本报表。你将学习使用报表向导和常用的报表设计器特性构造基本报表所需的基础知识。你还要学习报表设计的基础构造块，包括报表项和报表布局属性。该章描述了用来创建标准报表的两种报表设计工具。报表生成器面向的用户是那些希望使用简单直观的报表工具设计基本报表的用户。SQL Server Data Tools 则可以帮助你管理报表项目，还可以帮助你在 Microsoft Visual Studio 开发环境中设计报表。

第 5 章“报表布局和格式化”关注不同的报表设计布局和用来组装报表的组件。该章描述了报表体、报表页眉、报表页脚和页面格式化属性。你将学习每一种报表项的功能，还要学习用于组织和展示数据的数据范围组件(data range component)。在学习了基础知识后，你将学习如何分组数据、列表与数据区(data region)，还有如何使用表格和矩阵报表，以及通过报表来定义钻取报表，你还将学习如何使用图表。你还要学习如何编写表达式和自定义代码对格式化进行扩展和应用业务逻辑，以及如何为移动设备设计报表。

第 6 章“设计数据访问”揭示了报表都是基于一个数据源设计的，Reporting Services 可以用来对来自多个不同数据源的数据进行展示。你将学习如何定义独立数据源和共享数据源，还要学习查询和数据集，以及如何使用参数对数据库和报表服务器中的数据进行筛选。你将学习新的参数功能，这些参数功能是在最新版本的产品中引入的。该章提供了 T-SQL 查询和存储过程的入门知识。你还要学习如何使用 Analysis Services 和 MDX 查询生成器构建报表。该章针对 Oracle PL/SQL、Sybase，以及 Access SQL 方言提供了查询示例。

第 7 章“高级报表设计”通过介绍如何创造性地使用数据分组和报表项组合将设计元素提升到一个新的层次。通过使用简单的编程代码，可以添加计算和条件格式化功能。无论你是否是一位应用程序开发人员还是一位报表设计人员，该章所包含的重要内容都可以帮助你设计报表，使报表不仅能够满足用户的需求，还能够利用引人入胜的报表特性提升报表的水平。

第 8 章“图表报表”既关注简单图表可视化，也关注高级图表可视化。你将学习到如何用多种图表表达方式对值进行分组和聚集。该章介绍了可以用来在不同场景下表示不同数据类型的图表。我们还将学习几种很罕见的图表功能和图表属性，如果你需要创建更高级的可视化报表，那么可以使用这些功能和属性。

第III部分：设计商业智能报表

第9章“BI语义模型”介绍了SQL Server 2012引入的一个新主题和一组新功能。该章用简单的术语解释了这些选项和技术，澄清了各种选择和最佳解决方案。

第10章“使用Analysis Services编写报表”讨论了基于多维数据集和数据库编写报表所具有的优势和存在的独特挑战。你将学到MDX查询设计的基础知识，还将学习如何在报表设计器中使用多维数据集数据。

第11章“OLAP报表高级技术”关注的是基于多维数据集数据编写报表所使用的设计技术和独特的最佳实践。你将学习如何使用参数、计算和聚集来创建MDX查询。你还要学习应用高级报表设计模式和技术来构建一个动态的多维数据集浏览器。

第IV部分：启用用户报表

第12章“表格模型”使用PowerPivot和SQL Server 2012中的企业级表格模型介绍了语义建模。你将学习表格模型的基本使用方法和应用方式，从而能够执行内存聚集，对来自不同数据源的数据进行建模，定义关系，使用DAX进行计算，还能使用层次结构和KPI。

第13章“使用Power View进行可视化分析”描述了Power View，这是一种简单而又强大的用户报表工具，它可以用一种语义模型对信息进行可视化展示。你将学到如何通过浏览信息来回答关键业务问题并从一组数据中获得结论。对于那些掌握了Power View工具同时又懂得一种语义模型和工具的核心机制的用户，可以创建令人信服的报表，并且能够通过形象并具有良好的交互性的界面很好地展示业务。

第14章“报表生成器解决方案策略”指导你如何使用报表生成器设计工具，该工具可以用来启用用户自助报表编写场景。在完成必要的准备和计划之后，报表生成器设计工具可以帮助用户自行完成报表构建工作。你将学习到教会用户使用这项工具为数据生成报表。你还将学习如何为用户提供支持，对用户进行培训使之能够使用这项工具回答问题，并使用共享数据源、共享数据集和报表部件等共享组件来构建自己的报表。

第V部分：解决方案模式

第15章“管理报表项目”将帮助你规划和管理不同规模的报表项目。你将学习到如何在只有一个或几个报表设计人员的项目中维护小型报表项目，还要学习如何在一个包含了开发者和报表设计者的团队项目中管理大型的正规报表项目。我们将讨论在SharePoint中使用报表生成器来管理业务用户报表的策略，还要讨论在集成版本控制和团队协作的情况下使用SQL Server Data Tools的策略。

第16章“报表解决方案模式和要点”介绍的设计模式来源于我们多年的解决方案设计。我们将理论和实践进行了区分，讨论了行业内其他人同样能够有效应用的最佳实践和技术。还讨论了如何根据用户和业务的需求来建立标准和实践。

第VI部分：管理Reporting Services

报表服务器管理是一项重要的任务，这项任务需要确保数据的安全，只有那些授权用户才能访问数据。通过对服务器端的报表进行配置并进行安全设置，可以优化性能并为相应的用户团体提供相应的数据。第13章和第14章介绍了如何使用必要的工具对报表服务器进行配置和管理。

第 17 章“内容管理”将指导你如何使用管理工具和 Reporting Services 功能来发布报表、管理报表执行和传递。你将学习如何创建自动化脚本和自定义解决方案来管理报表服务器的全部内容。你要从管理员的角度重新审视报表执行的步骤，并学习如何优化。你还要学习如何实现自动化的报表传递和服务管理。

第 18 章“将报表与 SharePoint 集成”指导你如何同时使用 Reporting Services 和 SharePoint 来创建业务解决方案。无论你是在 SharePoint 集成模式下配置 Reporting Services，还是在 Native 部署模式下配置 Reporting Services，你都可以通过 Web 部件嵌入报表内容。你还将学习到如何利用集成模式功能使报表编写成为业务团队协作文化中的一个无缝组成部分。

第 19 章“Native 模式服务器管理”是一部综合性的管理员指南。针对 Reporting Services，该章将探讨报表需求和部署场景的相关注意事项。你还要学习使用配置工具和应用程序，学习备份和恢复过程，以及针对可能出现的问题和性能优化，如何监视一个 Reporting Services 实例。

第Ⅶ部分：Reporting Services 自定义编程

在实际使用过程中，Reporting Services 所有的内置功能都可以自动化，并可以通过自定义程序代码执行。这些功能包括报表呈现和报表环境的核心服务，即数据访问、呈现格式、安全性以及传递(delivery)。

第 20 章“将报表集成到自定义应用程序中”指出，Reporting Services 是一个灵活的报表工具，可以很方便地集成到不同的应用程序中。该章你将学习如何使用 URL，通过文档和 Web 页面链接来访问报表，还将学习如何使用 Reporting Services Web 服务以可编程的方式呈现报表。你还要使用 ReportViewer 控件将报表嵌入到自定义的 Windows Forms 和 ASP.NET Web Forms 应用程序中。此外，你还要学习使用 SharePoint 的 Web 部件及其他技术在 Web 门户中显示报表。你可以使用编程方式进行呈现，还可以使用 URL 进行呈现，甚至可以使用 ReportViewer 控件来创建自定义的报表查看器和参数接口。该章提供了 C# 和 VB.NET 示例。

第 21 章“使用嵌入代码和引用代码”展示了为报表添加自定义逻辑的技术，这项技术需要使用应用程序代码，而这些代码是包含在一个报表中或包含在一个外部引用的程序集组件中的。无论使用哪种方法，你都可以使用编程技术添加自定义行为，并启用高级报表功能。

第 22 章“扩展 Reporting Services”是为那些认真的应用程序开发人员编写的。该章使用了面向对象编程技术，提供了 C# 和 VB.NET 代码示例。在该章，你将学到如何通过创建自定义的库和自定义的扩展来为报表和报表服务器的特性添加功能。这些扩展可以用来访问不同的数据源，以特定的格式呈现报表，对用户进行身份验证，还可以不使用产品提供的默认方法完成传递报表的工作。

附录

附录描述了 T-SQL 语法、变量、函数，还提供了 MDX 语言功能和参考。

为什么需要使用本书

大多数较新的业务级计算机都可以满足设计和运行 SQL Server 2012 和 Reporting Services 的软硬件要求。某些与 SharePoint 集成在一起的特性和功能需要更高的配置，包括软硬件。自定义编程示例需要安装任意一个版本的 Visual Studio 2010。

针对运行 SQL Server 2012, 微软公司在 MSDN 库中给出了软硬件要求, 具体请参考网址 <http://msdn.microsoft.com/en-us/library/ms143506.aspx> 上的内容。安装和配置一个 SharePoint BI 环境是一个复杂的过程, 需要特定的硬件支持。针对运行 SharePoint、PowerPivot、Power View 以及在 SharePoint 集成模式下运行 Reporting Services, 微软公司给出了运行环境要求, 具体请参考网址 <http://msdn.microsoft.com/en-us/library/ee210640.aspx> 上的内容。

- 为了使用本书介绍的全部高级功能, 推荐使用企业版和开发版。你可以从微软网站 www.microsoft.com/sql 下载评估版的 SQL Server。
- 使用标准 Reporting Services 报表的报表设计示例在任何一个版本的 SQL Server 2012 中都能够正常工作, 并且可以在满足最低配置要求的计算机上运行。
- 第 11 章和其他章节中使用的示例需要使用在多维存储模式下安装的 SQL Server Analysis Services。SQL Server Analysis Services 是 SQL Server 安装过程中的一个安装选项, 并且是在完整安装一个 SQL Server 的过程中一并安装的。
- 第 13 章和第 14 章介绍 SharePoint 集成和 Power View, 要求安装 SharePoint 2010 Enterprise 版。
- 在报表设计器外部执行的自定义编程示例需要一个单独安装的 Visual Studio 2010。第 20~22 章介绍了此项内容所需的材料。
- 在示例和练习中使用的示例数据库可以从 www.wrox.com 网站上随本书的示例项目一同下载。此外, 还可以获得其他资源。
- 完整的示例源代码可以从我们的 Web 网站 www.wrox.com 下载。针对编程示例, 下载时提供了 Visual Basic .NET 代码和 C# 代码。

为了模拟一个多服务器的服务器环境, 一种有效的方法就是使用虚拟机映像。为了测试和评估其开发的工具, 微软公司通过使用预配置服务器安装提供了虚拟映像。这些内容往往包含在使用时间上有限制的软件。如果需要了解有关支持材料, 请在 www.wrox.com 查看与此有关的最新信息。

本书的约定

为了帮助你更好地使用本书内容并掌握当前阅读内容, 本书使用了一些特殊的表示法。



带有一个警告符号的文本框用于说明重要的信息, 文本框内的信息与前后文本直接相关, 你务必牢记。



带有一个铅笔符号的文本框用于说明与当前讨论内容相关的注解、提示、线索、技巧和特别说明。

在本书中, 我们使用两种方式来表示代码:

本书大多数情况下都使用 monofont 字体表示代码示例。

表示代码示例时, 如果代码在上下文环境中特别重要, 那么将用加粗的字体强调这段代码的重要性。

源代码

当你在完成本书的示例过程中，既可以以手动当时输入所有代码，也可以使用与本书配套的源代码文件。书中所有的所有源代码都可以通过 www.wrox.com 或 www.tupwk.com.cn/downpage 下载。在访问该网站时只需要找到本书的标题(可以通过 Search 框或者标题列表来查找)，然后单击本书详细资料页面中的 Download Code 超链接来获取所有的源代码。其中包含在 Web 网站的代码用下面所示的图标来突出显示：



程序清单中包括了相应的文件名。如果仅仅是一个代码片段，那么在代码注释中将会看到下面所示的文件名：

代码片段文件名



提示：因为许多书可能有相类似的标题，所以最简单的方法是通过 ISBN 来进行查找；本书英文版的 ISBN 为 978-1-118-10111-7。

在下载了代码后，请用你所喜欢的解压缩工具进行解压。或者也可以访问 Wrox 主下载页面来查看本书或者其他 Wrox 书籍的代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件，我们会检查您的反馈信息，如果是正确的，我们将在本书的后续版本中采用。

要在网站上找到本书英文版的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

P2P.WROX.COM

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，

便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读使用协议，并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit 按钮。
- (4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。



不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 Subscribe to this Forum 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

目 录

第 I 部分 入门

第 1 章 Reporting Services 介绍.....	3	2.3.1 编写.....	38
1.1 哪些人需要使用 Reporting Services	4	2.3.2 管理.....	38
1.1.1 业务信息工作者.....	5	2.3.3 传递.....	39
1.1.2 业务管理者.....	5	2.4 Reporting Services 工具	39
1.1.3 软件开发者.....	5	2.4.1 报表设计器.....	39
1.1.4 系统管理员.....	5	2.4.2 Power View.....	39
1.2 仪表板、报表和应用程序.....	5	2.4.3 报表生成器.....	40
1.3 企业报表.....	8	2.4.4 第三方编写工具.....	40
1.4 解决方案类型	9	2.4.5 报表管理器.....	40
1.4.1 简单报表设计.....	10	2.4.6 SharePoint 库和 Web 部件.....	40
1.4.2 IT 专业人士设计的报表	10	2.4.7 Reporting Services 配置管理器	40
1.4.3 用户设计的报表.....	11	2.4.8 SQL Server 管理程序.....	41
1.4.4 基于服务器的报表	14	2.4.9 命令行工具.....	41
1.5 商业智能报表解决方案.....	15	2.4.10 HTML Viewer.....	41
1.5.1 报表数据源.....	16	2.4.11 Report Viewer 控件.....	41
1.5.2 复杂性和报表性能	18	2.4.12 Reporting Services Web 服务.....	42
1.6 自定义报表体验	18	2.5 Reporting Services Windows 服务.....	43
1.7 小结.....	19	2.5.1 HTTP.SYS 和 HTTP 侦听器.....	44
第 2 章 Reporting Services 安装和架构.....	20	2.5.2 安全子层.....	44
2.1 基本安装.....	21	2.5.3 报表管理器和 Web 服务	45
2.1.1 安装 Reporting Services.....	21	2.5.4 核心处理功能.....	45
2.1.2 安装 Reporting Services 示例和 SQL Server 示例数据库.....	33	2.5.5 服务管理.....	45
2.2 企业部署.....	34	2.5.6 WMI 和 RPC 接口.....	46
2.2.1 SQL Server 版本.....	34	2.6 Reporting Services 处理程序和扩展	47
2.2.2 命名实例	35	2.6.1 报表处理程序.....	48
2.2.3 拓扑结构.....	36	2.6.2 数据处理扩展.....	49
2.2.4 模式.....	36	2.6.3 报表项	50
2.2.5 安装选项	37	2.6.4 呈现扩展	50
2.2.6 命令行安装	37	2.6.5 计划和传递处理程序	52
2.3 报表生命周期	38	2.6.6 传递扩展	53
		2.7 Reporting Services 应用程序数据库	53
		2.7.1 ReportServer.....	53

2.7.2	ReportServerTempDB	54
2.8	小结	54
第 3 章	配置 SharePoint 集成	56
3.1	SharePoint 技术	57
3.1.1	SharePoint Foundation	57
3.1.2	SharePoint Server	57
3.1.3	SharePoint Web 部件	59
3.2	安装和配置	59
3.2.1	安装 SharePoint Server 2010	60
3.2.2	安装 Service Pack (SP) 1	62
3.2.3	安装和配置 PowerPivot for SharePoint	63
3.2.4	为 SharePoint 安装配置 Reporting Services	66
3.2.5	SharePoint 网站设置	71
3.3	架构	72
3.4	小结	74

第 II 部分 报表设计

第 4 章	基本报表设计	77
4.1	向导到底有什么好处	77
4.1.1	构造块	77
4.1.2	学习使用报表生成器	86
4.2	手动报表设计	91
4.2.1	构建一个矩阵	91
4.2.2	运行报表和保存报表	96
4.3	小结	97
第 5 章	报表布局和格式化	98
5.1	报表布局类型	99
5.1.1	表格报表	99
5.1.2	矩阵报表	100
5.1.3	列表报表	100
5.1.4	图表报表	101
5.1.5	仪表报表和仪表板	102
5.1.6	地图报表和空间报表	102
5.1.7	定义表格分组	102
5.1.8	交互式排序	106
5.1.9	对报表数据进行格式化	109

5.1.10	引入动态格式化	110
5.2	设计多列报表	110
5.3	设计仪表报表	110
5.3.1	浏览器兼容性	111
5.3.2	脱机查看	112
5.3.3	呈现格式的限制及考虑事项	112
5.4	小结	113
第 6 章	设计数据访问	114
6.1	编写商业智能报表	117
6.2	基于关系数据编写报表	118
6.2.1	数据和查询基础	118
6.2.2	数据源	119
6.2.3	T-SQL 查询设计	123
6.2.4	星型架构的数据仓库	124
6.3	筛选技术	129
6.4	使用其他数据源	145
6.4.1	Microsoft Access	146
6.4.2	在一个字符串表达式中构造一个 查询	148
6.4.3	Microsoft Excel	148
6.4.4	Oracle P/L SQL	149
6.4.5	Sybase Adaptive Server	150
6.4.6	联合使用数据源	150
6.5	最佳实践	151
6.5	小结	152
第 7 章	高级报表设计	153
7.1	页眉和页脚	154
7.1.1	聚集函数和总计	157
7.1.2	向一个表或矩阵报表中添加总计	158
7.2	创建报表模板	159
7.3	创建复合报表	160
7.3.1	剖析一个文本框	161
7.3.2	边距和缩进	161
7.3.3	嵌入式的格式化	162
7.3.4	设计主/明细报表	164
7.3.5	分组和数据集范围	164
7.4	设计子报表	170
7.5	在报表之间导航	176

7.5.1 创建文档结构图	176	10.6.1 创建多维数据集报表操作	237
7.5.2 链接和钻取报表	178	10.6.2 参数安全预防措施	237
7.6 为递归关系编写报表	180	10.7 最佳实践和规则	238
7.7 小结	184	10.8 小结	238
第 8 章 图表报表	185	第 11 章 OLAP 报表高级技术	239
8.1 图表类型	186	11.1 Cube Dynamic Rows 报表	239
8.1.1 柱状图	186	11.1.1 解剖 Cube Dynamic Rows 报表	240
8.1.2 分区图和折线图	188	11.1.2 对 CUBE DYNAMIC ROWS 报表的总结	248
8.1.3 饼图和圆环图	189	11.2 Cube Dynamic Rows 报表扩展	248
8.1.4 气泡图和股价图	190	11.2.1 修改 MDX 查询	248
8.1.5 图表类型总结	191	11.2.2 修改设计界面	249
8.1.6 解剖一个图表	193	11.3 Cube Restricting Rows 报表	250
8.2 创建一个多序列图表	194	11.4 Cube Metadata 报表	255
8.2.1 使用多个图表区	196	11.4.1 设计报表	255
8.2.2 有用的属性和设置	198	11.4.2 添加其他多维数据集元数据	258
8.3 小结	200	11.5 Cube Browser 报表	262
第 III 部分 设计商业智能报表		11.5.1 对报表的剖析	264
第 9 章 BI 语义模型	203	11.5.2 内幕	266
9.1 数据建模导论	203	11.5.3 最后的思考	278
9.2 BI 语义模型	204	11.6 小结	279
9.2.1 概念架构	207	第 IV 部分 启用用户报表	
9.2.2 BI 语义模型的混合特征	212	第 12 章 表格模型	283
9.3 小结	212	12.1 PowerPivot 介绍	283
第 10 章 使用 Analysis Services 编写 报表	214	12.1.1 PowerPivot for Excel	286
10.1 为什么需要在报表中使用 Analysis Services	214	12.1.2 安装和设置	287
10.2 在使用 Reporting Services 的过程 中使用 Analysis Services 数据	216	12.2 将数据导入 PowerPivot	287
10.3 使用多维表达式语言	217	12.3 PowerPivot 窗口	291
10.3.1 MDX: 简单还是复杂	217	12.3.1 “开始”选项卡	291
10.3.2 使用 MDX 查询设计器生成 查询	218	12.3.2 “设计”选项卡	292
10.4 添加非附加性度量	232	12.3.3 “高级”选项卡	293
10.5 MDX 属性和多维数据集 格式化	235	12.4 分析数据和丰富数据	294
10.6 对报表进行钻取	236	12.4.1 筛选和排序	294
		12.4.2 关联	295
		12.4.3 计算列	297
		12.4.4 度量	298
		12.4.5 浏览模型	300

12.5 小结.....	301	14.2.7 数据源和查询选项.....	370
第 13 章 使用 Power View 进行可视化分析	302	14.3 用户报表迁移策略.....	374
13.1 Power View 介绍.....	303	14.4 小结.....	376
13.2 Power View 架构.....	307	第 V 部分 解决方案模式	
13.3 为教程准备模型和连接.....	309	第 15 章 管理报表项目	379
13.3.1 部署 FAA 航班数据模型.....	309	15.1 解决方案和项目.....	379
13.3.2 为 FAA 航空图像创建一个 SharePoint 图像库.....	310	15.1.1 项目结构.....	380
13.3.3 将 FAA 工作簿直接发布到 PowerPivot 库.....	311	15.1.2 共享数据集和数据源.....	382
13.3.4 为 Power View 创建数据源连接.....	312	15.1.3 成功的关键因素.....	382
13.4 使用 Power View 进行可视化分析.....	315	15.1.4 报表规范.....	383
13.4.1 Power View 初步.....	315	15.1.5 报表模板.....	384
13.4.2 设计体验介绍.....	316	15.1.6 开发阶段.....	384
13.4.3 可视化和交互性.....	329	15.2 版本控制.....	385
13.4.4 在 Power View 中展示和导出.....	342	15.2.1 为什么使用版本控制.....	386
13.4.5 Power View 的窍门和技巧.....	347	15.2.2 设置版本控制.....	386
13.5 配置数据源连接.....	348	15.2.3 获得最新版本.....	386
13.5.1 连接到 PowerPivot 工作簿.....	349	15.2.4 查看报表历史.....	387
13.5.2 连接到 BISM 连接文件.....	349	15.2.5 恢复一个报表的先前版本.....	387
13.5.3 连接到 RSDS.....	351	15.2.6 设置签入/签出策略.....	387
13.5.4 身份验证场景.....	352	15.2.7 应用标签.....	387
13.6 Analysis Services 表格.....	357	15.3 对内容进行同步.....	387
13.6.1 安装 Analysis Services 表格.....	358	15.3.1 部署一个报表.....	387
13.6.2 将一个 PowerPivot 工作簿部署到 Analysis Services 表格中.....	360	15.3.2 部署一组报表.....	388
13.7 小结.....	361	15.3.3 检查生成错误.....	388
第 14 章 报表生成器解决方案策略	363	15.3.4 从一个部署中排除一个报表.....	388
14.1 报表生成器和语义模型历史.....	364	15.4 管理服务器内容.....	388
14.2 计划一个自助报表环境.....	366	15.4.1 检查部署位置.....	388
14.2.1 必须进行计划.....	366	15.4.2 管理 Native 模式下的内容.....	389
14.2.2 设计方法和适用场合.....	366	15.4.3 管理 SharePoint 中的内容.....	391
14.2.3 定义所有者.....	367	15.5 Azure 报表入门.....	391
14.2.4 数据治理.....	368	15.6 小结.....	395
14.2.5 数据源访问和安全性.....	368	第 16 章 报表解决方案模式和要点	396
14.2.6 用户教育.....	369	16.1 超级报表.....	397
		16.2 报表要点：基于基本技巧构建报表.....	400
		16.2.1 仪表板解决方案的数据源和数据集.....	400

16.2.2 KPI 记分卡	400	18.7 小结	476
16.2.3 交互式迷你图和图表	405	第 19 章 Native 模式服务器管理	477
16.2.4 具有导航和缩放功能的地图	412	19.1 安全	478
16.2.5 使用报表部件来组装一个 仪表盘	418	19.1.1 帐户管理	478
16.2.6 动态颜色和主题	420	19.1.2 系统级角色	482
16.2.7 具有动态列的表格报表	423	19.1.3 表面区域管理	483
16.3 小结	427	19.2 备份和恢复	483
第 VI 部分 管理 Reporting Services		19.2.1 应用程序数据库	484
第 17 章 内容管理	431	19.2.2 加密密钥	486
17.1 使用报表管理器	431	19.2.3 配置文件	487
17.2 内容管理活动	434	19.2.4 其他项	488
17.2.1 文件夹	435	19.3 监视	488
17.2.2 共享数据源	437	19.3.1 安装日志	488
17.2.3 报表模型	438	19.3.2 Windows 应用程序事件日志	489
17.2.4 报表	440	19.3.3 跟踪日志	489
17.2.5 报表资源	446	19.3.4 执行日志	492
17.2.6 共享计划	446	19.3.5 性能计数器	493
17.3 项级安全性	447	19.3.6 服务器管理报告	496
17.4 内容管理自动化	454	19.4 配置	496
17.4.1 RS 工具	454	19.4.1 内存管理	496
17.4.2 Reporting Services 脚本	456	19.4.2 URL 保留项	497
17.5 小结	457	19.4.3 电子邮件传递	499
第 18 章 将报表与 SharePoint 集成	459	19.4.4 呈现扩展	501
18.1 Native 模式下的 Web 部件	459	19.4.5 我的报表	502
18.1.1 安装 Native 模式的 Web 部件	460	19.5 小结	504
18.1.2 Report Viewer	461	第 VII 部分 Reporting Services 自定义 编程	
18.1.3 Report Explorer	462	第 20 章 将报表集成到自定义应用 程序中	507
18.2 将报表发布到 SharePoint	463	20.1 URL 访问	508
18.2.1 SharePoint Foundation 和 SharePoint Server Standard Edition	463	20.1.1 URL 语法	508
18.2.2 SharePoint Server Enterprise Edition	468	20.1.2 访问 Reporting Services 对象	509
18.3 报表管理	470	20.1.3 Reporting Services URL 参数	513
18.4 集成模式 Web 部件	472	20.1.4 通过 URL 传递报表信息	518
18.5 Native 模式和集成模式	474	20.2 通过编程进行呈现	520
18.6 报表模型	475	20.2.1 一般场景	521
		20.2.2 通过 Windows 呈现	522

20.2.3 将报表呈现到 Web	543	22.1.3 对数据处理扩展的详细研究	580
20.3 使用 ReportViewer 控件	550	22.2 创建一个自定义数据处理扩展	582
20.4 小结	559	22.2.1 场景	582
第 21 章 使用嵌入代码和引用代码	561	22.2.2 创建和设置项目	583
21.1 使用表达式生成器	563	22.2.3 创建 DataSetConnection 对象	585
21.2 计算字段	564	22.2.4 创建 DataSetParameter 类	595
21.3 条件表达式	566	22.2.5 实现 IDataParameter	596
21.4 IIF()是好帮手	567	22.2.6 创建 DataSetParameterCollection 类	598
21.5 使用自定义代码	569	22.2.7 创建 DataSetCommand 类	600
21.5.1 在报表中使用自定义代码	569	22.2.8 创建 DataSetDataReader 对象	616
21.5.2 使用自定义程序集	571	22.2.9 安装 DataSetDataProcessing 扩展	620
21.5.3 自定义程序集的安全性	573	22.2.10 测试 DataSetDataExtension	623
21.5.4 错误、报警及调试代码	573	22.3 小结	626
21.6 小结	574		
第 22 章 扩展 Reporting Services	575	第VIII部分 附录	
22.1 通过接口进行扩展	577	附录 A T-SQL 命令和语法参考	631
22.1.1 什么是接口	577	附录 B T-SQL 系统变量和函数	650
22.1.2 接口语言之间的区别	577	附录 C MDX 参考	664

第 I 部分

入 门

- 第 1 章 Reporting Services 介绍
- 第 2 章 Reporting Services 安装和架构
- 第 3 章 配置 SharePoint 集成



第 1 章

Reporting Services 介绍

本章内容

- 理解报表设计者角色和用来设计报表的工具
- 理解仪表板、报表和应用程序
- 研究商业智能解决方案
- 研究多维和表语义模型

你手里捧着这本书，希望判断本书能否帮你解决问题，或是能够教会你使用 Reporting Services 来创建报表的基本技能。如果你我是私下里讨论这个问题，那么我会问你，你到底想要什么？在我从事教学和拜访公司来设计报表和 BI 解决方案的过程中，在每一次课程或每一次咨询活动的开始，我都会问问学生或客户，他到底想要什么？需求是什么？你的报表到底要回答什么问题？存在什么问题？需要满足什么需求？为构建一个解决方案来达到目的，需要哪些条件？所以我问你，你到底想要什么？为什么你阅读与 Reporting Services 相关的书籍？你是不是必须解决某个具体问题？还是你只需获得一些基本的报表设计技能？你是否需要构建一个完整的报表解决方案？这些报表的用户都是谁？是部门工作人员、业务经理还是财经分析人员？也许你的用户是一家大型公司的CEO 或者是其他业务经理，他们希望了解公司是否正常运作。也许你需要为你自己的业务创建报表，以确保其盈利并达到目标。也许你需要销售产品，为此需要创建一个票据或一个 BI 仪表板来帮助管理一个跨国公司，这时，你创建的报表就是非常重要的。因此，你需要确保报表提供了精确的信息并且设计正确，同时使用了行业最佳实践。这可是一项重大职责。

无论你的需求是什么，我们都会介绍你所需要的技术内容，并对每个主题进行详细介绍。我专程与一些最为可信的同事进行了交流，以此来分享其经验。

本章针对这个强大的 Microsoft SQL Server 2012 报表工具和数据分析平台的概念和功能，从一个较高的层次进行了介绍。本章介绍了常用的报表场景，首先介绍了最常用的基本功能，然后介绍了比较高级的功能。在后续章节里，你可以深入研究这些功能，学会如何在自己的报表解决方案中使用这些功能。

SQL Server Reporting Services 已经成为事实上的行业标准报表工具，并成为其他报表工具的衡量标准。SQL Server Reporting Services 是一个基础，在这个基础构成的基石之上，你可以为业务用

户构建完整的报表、记分卡和仪表板解决方案。今天，SQL Server Reporting Services 能够完成从简单的独立数据报表一直到发布企业级基础报表，并将其集成到业务门户和自定义应用程序中的全部任务。

不久前，一家大型金融服务公司的信息技术(IT)部门希望了解他们正在使用的报表工具是不是当时市场上最好的产品。他们决定雇佣一家咨询公司来评估每一种主流的报表产品，并希望咨询公司能够向他们提交一份公正的分析报告。我有幸参加了这项工作。我们与客户一同工作，提出了 50 项评估标准。然后我与所有的主流产品生产商进行了沟通，安装了各个产品的评估版本并对各项功能进行了研究。与其他客户进行了交谈，并且和具体使用过不同产品的客户进行了交流。这项工作确实帮助我们从一个更为广阔的视角对此行业进行了观察，是一次有益的学习体验。市场上确实存在一些值得尊敬的产品，这些产品各有长处。但是我可以诚恳地指出，微软公司的产品是一个独一无二的特殊平台。作为一个咨询人员，作为微软公司的一个合同制员工，作为微软一个 SQL Server MVP，我有幸与 Reporting Services 产品团队共同工作了多年。这个团队视野宽阔，对他们自己开发的产品富有激情。我非常尊敬这些优秀的人才，他们一个接一个版本地不断开发和改进着 Reporting Services。

1.1 哪些人需要使用 Reporting Services

通过观察业务用户使用报表的方式，我们可将业务用户划分为几个大类。某些用户仅仅是报表的消费者。他们乐于使用为他们编写好的和发布的报表。其他用户偏好使用业务工具自行创建报表，他们会使用这些工具，还使用了这些工具完成其他任务，例如，他们使用 Excel 完成计划和财务分析。他们有可能只想通过浏览信息来发现趋势，并了解业务是如何按照他们的目标不断提升的。其他业务用户需要使用更复杂的工具来创建强大的报表。在大多数大型机构中，一个典型的信息技术部门的人员都包括三种角色：系统管理员、应用程序开发人员以及项目管理人员。一般而言，部门中其他人员都是为这三种角色提供支持的。那么，机构中的报表设计人员在哪里？老实说，我并没有一个简单的答案。实际上，设计业务报表的人员并不是来自同一个部门的 IT 专业人士。实际上，许多主要工作是创建报表的人来自业务团体的各个组成部分，而不一定是典型的计算机天才。

微软公司长期以来一直从事高技术产品的开发，这些产品吸引了技术团体。最近几年，微软公司开始更多地为那些穿西装打领带的群体开发产品，这些人谈论的内容包括业务绩效管理策略、市场共享，而不是远程过程调用和多态对象集成。如果你是一个业务类型的人士，你可能并不关心如何将自己的报表集成到自定义应用程序中和 Web 网站中，也不关心如何编写复杂的程序逻辑使报表更为丰富。我们有些人工作就是为了业务。你关心的可能是为你的熟练业务用户提供将报表部件从部件库拖放到报表中，从而能够以可视化的方式查看重要的关键指标度量，目的在于观察其销售的产品中，哪些产品销售情况良好。然而，为了提供这种体验，必须具有一定的专业技术水平。

在过去几年中，我研究了那些认为自己是报表设计者的人士。他们一般分为两大类，即业务类和技术类。对于那些技术天分比较差的人来说，报表工具必须具有更好的可用性，这是一项重要区别。下面的角色代表了报表工具的主要用户，并且描述了我们在行业内观察到的某些趋势，目前这种趋势仍然在不断演化。

1.1.1 业务信息工作者

属于这个角色的人员都具有比较强的计算机技能，但是他们并不花费时间来编写代码和使用编程工具。他们的主要兴趣是对信息进行探索并发现答案，而不是实际复杂的报表。如果你是一位信息工作者(information worker, IW)，那么你需要使用简便的工具来浏览数据，迅速创建简单的报表，并且无须掌握过多的技术知识。一般情况下，IW 通过创建报表来回答具体问题或来满足某项特定的需求，然后他们就会删除这个报表，或者将这个报表保存到个人空间以备重复使用。他们倾向于为每一项任务创建一个单独的报表，可能会(也可能不会)与其他具有类似需求的人员共享报表。这个群体是行业内规模最大、增长速度最快的报表工具用户群。

1.1.2 业务管理者

如果你是一位业务管理者，那么你的主要兴趣是你自己的业务领域。管理者需要使用报表来支持具体的过程，满足其分析需要，并帮助他们做出合理的决策。与信息工作者类似，管理者对报表执行的实现细节和技术不感兴趣，他们可能需要创建自己的报表来分析其团队或责任领域的生产效率。

1.1.3 软件开发者

为了使用高级报表特性，软件开发者要编写复杂的查询代码和自定义编程代码来处理业务规则，并定义报表条件格式化及行为。一般情况下，开发者熟悉报表设计环境，因为报表的设计环境与类似的编程工具是相似的。然而报表设计与应用程序开发不同。在某种程度上，与开发软件相比，设计报表更便捷。设计高级报表需要编写代码，甚至需要开发自定义组件。

1.1.4 系统管理员

如果你是一位系统管理员，一般情况下你最关心的是如何安装和维护服务器和基础设施，从而使报表解决方案能够正常工作与使用。管理员常常要花费时间和精力来管理安全和优化系统来提高工作效率。Reporting Services 提供了一个管理组件，在大规模实现中，这个组件特别重要。

在较小的组织机构中，一个人可能身兼系统管理员、开发者和报表设计者的角色。报表也可以用来协助使用监视系统和维护统计数据，从而减轻系统管理员的工作负担。

Reporting Services 使用不同的报表设计和数据浏览工具来满足信息工作者和技术专家的需求。报表生成器简单易用，功能单一，其界面类似于 Microsoft Office。SQL Server Data Tools(SSDT)是更为高级的报表设计器，是面向应用程序开发和其他技术专家的。Visual Studio 解决方案设计环境集成了 SSDT。

1.2 仪表板、报表和应用程序

从软件角度看，一个计算机系统可以用不同的方式向用户展示数据。大多数遗留报表系统都是运行在客户端桌面计算机上的。

就在最近，我们看到了报表设计出现了自助化趋势。人们开发了许多报表设计工具，这些工具试图在易用性和功能之间取得平衡。经过几年竞争，尘埃落定，那些最优秀的工具开始浮出水面。大多数业务管理者需要一个简单易用的仪表板界面，用于回答关键的业务问题，或者在无须设计报

表和编写查询代码的情况下，需要一个能够用来浏览简化的视图或语义模型的工具。微软公司为此采用了两种不同方法。首先，微软公司设计了一个现代化的报表设计工具，即报表生成器(Report Builder)，它可以方便那些技术背景较弱的用户使用传统的报表定义架构来设计强大的报表。其次，微软公司开发了一个独立的可视化工具，即 Power View，它使用了 Reporting Services 架构，但是提供了与传统报表不同的设计和用户体验。为了方便，我们在此不讨论 Power View 工具，并将其视为独立于其他 Reporting Services 功能的工具。你将在第 12 章和第 13 章学习表格模型(tabular model)和 Power View 工具。

早在好几年前，我们观察到应用程序出现了从基于客户端的处理模式转向基于 Web 服务器运行的趋势。基于 Web 服务器运行应用程序是一种能够有效地为大量人群提供系统访问功能的方法。Reporting Services 第一次发布时，人们只能通过基于服务器的解决方案访问 Reporting Services，报表交付基本上只能通过 Web 服务器完成，而这也是目前 SQL Server Reporting Services(SSRS)的主要使用方式。然而 SSRS 功能不仅限于此。Reporting Services 能够以多种模式运行，还可在多种应用程序中运行。如果我们观察过去 20 年来计算机系统是如何演化的，那么我们将看到：集中式的基于服务器的解决方案和基于客户端的应用程序都独具优势，并且在特性、功能、交互式用户体验和可扩展性等方面都达到了平衡。

必须指出的是，你可以在无须操心的情况下完成产品的安装、报表设计和使用等任务，同时，你也无须成为一名技术专家。本书后续章节将讨论如何以更为先进和更富有创造力的方式使用 Reporting Services。

混淆应用程序和报表之间的界限

利用 Reporting Services 可以将报表与应用程序集成在一起，从而使用户无法分清报表内容和应用程序界面之间的区别。为此只需要使用少许编程代码，即可对报表功能进行扩展，使其无论从外观还是行为都与应用程序相似。报表是从什么时候开始替代应用程序功能的？确切地说，报表、仪表板和计分板之间的区别是什么？他们之间的界限已经变得相当模糊。你的任务就是决定哪一项工具能够最好地满足你的需求。许多企业内部网的网站都运行在 Web 门户上，而不是运行在自定义编程的 Web 网站上。因此，Reporting Services 能够很自然地兼容任何 Web 门户环境。特别是，Reporting Services 本身就能够集成 Microsoft SharePoint Server。

令人激动的是，你现在已经拥有了一个能够完成一些令人难以置信的工作的工具。就像我们最喜欢的超级英雄叔叔所言：“权利越大，责任越大”。如果你的需求很简单，只需要设计简单的报表，那么对你而言，好消息就是使用 Reporting Services 也可以设计简单的报表。如果你是一位软件开发者，并打算使用这项功能强大的框架来探索这项给人以深刻印象的技术，那么欢迎你来到创建自定义报表的奇幻世界。

经过数年的产品体验，我得出了一个重要的教训。有人说，对一个锤子而言，世界上任何东西看起来都像是一个钉子。与此类似，对一个程序员来说，许多挑战看起来都像是一个编程的机会。在某些条件下，这种看法是正确的。但最有效的解决方案常常是直接使用一个产品中已有的功能，并且按照这项功能的设计目的来使用这项功能。就这一点，我常常与花费了几个小时就一个简单问题编写一个复杂解决方案的程序员进行交谈。我父亲常常建议我遵守著名的 KISS 法则：“保证简单易用(Keep it simple, stupid.)”。最后一部分是为了使这个词语容易记住，他是用最为慈爱的方式解释这一点的(至少我确信他是这个意思)。

1. 从一个应用程序启动报表

超链接和应用程序快捷方式可以很方便地添加到文档和自定义应用程序中。很多标准的报表查看环境都可以使用参数进行控制，而参数可以通过URL 传递给报表服务器。报表可以设计为能够提示用户输入参数值来筛选数据和修改报表格式及输出。这些参数还可以嵌入到一个URL 字符串中。这样，一个超链接可以使用一组数据来显示一个报表，其他超链接则可以用不同的数据显示同一个报表。参数甚至可以用来修改显示属性，例如字体大小和颜色，还可以用来隐藏和显示报表内容。

2. 用户交互

过去，许多报表仅仅是一个给出了带有汇总结果功能的数据列表。现在报表可以用来指导用户利用报表中的信息进行决策。报表元素(如文本标记、列名、图表数据点)都可以用导航来访问不同的报表章节，甚至可以用来导航访问新的报表。因为导航链接可以是数据驱动的，并且可以基于程序逻辑动态地创建，所以报表链接也可以用来导航到业务应用程序中。想想看，你现在可以使用报表启动程序并导航到文档库和在线内容！

当用户单击报表中的项和数据点时，报表的内容和布局可以根据背景中的参数值集合发生变化。汇总的标题也可以展开，从而能够显示更多细节，这样用户就可以下钻获取更具体的信息，如图 1-1 所示。

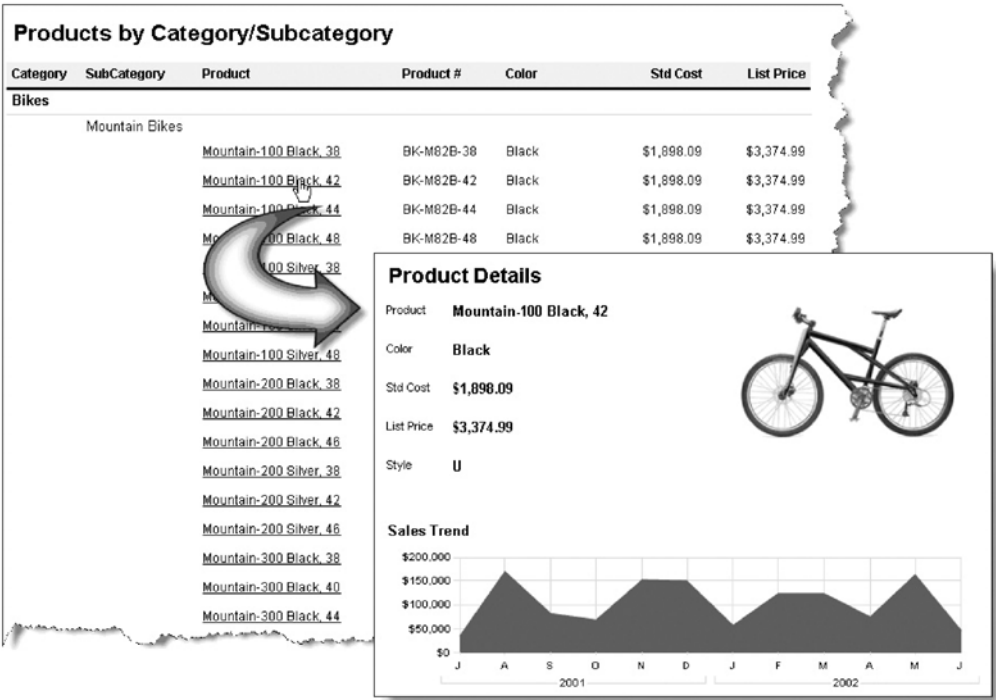


图 1-1

3. 将报表与应用程序集成

Reporting Services 很容易集成现代的 Web 应用程序，其原因之一是 Reporting Services 本身支持

超文本标记语言(Hypertext Markup Language, HTML),这是一种用来创建 Web 页面的标准标记语言。可以采用多种技术将报表集成到一个 Web 应用程序中,这些技术包括:

- 利用一个超链接将 Web 浏览器窗口导航到一个报表中
- 利用一个超链接,在一个单独的 Web 浏览器窗口中打开报表,为此需要使用报表显示控件,并且控件要具有浏览器功能
- 使用一个框架 IFrame 或一个 Web 控件 ReportViewer,将报表嵌入在一个页面中
- 使用服务器端自定义代码,通过编程方式将报表内容发送到一个 Active Server Page(ASP 或 ASPX)
- 通过编程方式将报表写入文件,方便用户从 Web 网站上下载这些报表
- 使用一个 Web 部件,将报表嵌入在一个 SharePoint Web 门户中
- 在 SharePoint 集成模式中,将报表与报表服务器完全集成在一起

为将报表集成到一个 Web 应用程序中或一个桌面应用程序中,可以采用多种创造性的方法。这些技术既可以是仅使用了少量 HTML 脚本的简单方法,也可以是复杂的自定义方法。如果这些方法仍不足以将报表嵌入到自定义的 Web 页面中,还可以使用编程方式将附加的内容嵌入报表。将报表与应用程序集成的方法很多,实际上你根本不需要考虑什么其他问题,你只需继续阅读本书即可!

SharePoint 集成模式完全可以让我们在 SharePoint 中进行报表设计和报表管理。如果你打算在自己的 SharePoint 门户中单独管理报表服务器,那么你仍然可以使用 SharePoint Web 部件在文件夹和报表之间导航,还可以查看驻留在 Reporting Services Native 模式下运行的报表服务器中的报表。

为了查看一个表单中的基于服务器的报表,可以使用 ReportViewer 控件或嵌入的 Web 浏览器。这些报表仍然是在报表服务器中进行管理的,同时它们也维护了由管理员定义的全部安全设置和配置选项。查询和数据访问仍然是在服务器上执行的。利用其他选项,可以将这些报表直接嵌入到客户端应用程序中。Windows Forms ReportViewer 控件可以充当一个轻量级的报表呈现引擎。这意味着:那些嵌入到一个自定义应用程序中的报表也可以在报表服务器上独立运行。

1.3 企业报表

向大众传递(deliver)报表需要一个合适的报表环境。Reporting Services 与桌面程序不同,桌面程序从源数据库获取数据,然后进行处理,但是 Reporting Services 需要处理查询,然后在报表服务器呈现报表。因为 Reporting Services 使用了 Windows 服务、基于共享服务器的组件以及 HTTP Web 服务,因此所有的处理工作都是在一个高效安全的环境中完成的。对 SQL Server 及其他企业级数据库来说,它们提供的标准的数据源连接都能够提高服务器资源的使用效率。简而言之,这意味着可以让大量用户在同一时刻运行这些报表,同时又保证了服务器资源的消耗最小化。

一个商业智能解决方案将来自多个数据源的数据集成到一个数据库仓库、数据市场或语义数据模型中。复杂的分析解决方案常常需要使用表格(tabular)或多维数据结构。如果你对术语“OLAP”、“表格”和“语义模型”感到陌生,那么一开始你会感到有些混乱。下面给出的速成教程可以帮助你分清楚这些内容。在 20 世纪 90 年代末期,微软公司随 SQL Server 7.0 发布了一个名为“OLAP Services”的产品。这是该公司第一项多维数据库技术,可以执行联机分析处理,以多维数据集和多维度方式存储数据,而不是以表的方式存储数据。在 SQL Server 2000 中,OLAP Services 演化为

Analysis Services。多维度(OLAP)数据库在磁盘中以预先分组和预先聚集的格式存储数据，因此针对报表和浏览需求而言，数据访问速度相当快。在 SQL Server 2012 中，微软公司将其分析数据库技术重新包装为“商业智能语义模型(Business Intelligence Semantic Models)”，即“BISM”。他们根据 BISM 提供了两种技术，即多维度(OLAP)和表格模型(tabular model)，可以实现在内存中保存和处理分析的功能。某些情况下，表格模型比较容易设计，效率可能更高，因此对生成报表和分析而言，速度更快。本书第 III 部分的章节将讲解 BI 数据源，并将介绍如何使用分析数据和语义模型来设计报表。

Reporting Services 报表服务器向外界提供功能的方式与标准的 ASP.NET Web 网站向用户提供服务的方式是相同的。报表既可以在公司防火墙内部进行访问，也可以在公司防火墙外部进行访问，但是针对后者，只有那些有权限的用户才能使用报表功能。在 SharePoint 集成模式中，用户只能通过文档库使用报表，并且报表是在 SharePoint 服务器环境内部进行管理并获得安全保障的。在 Native 模式或非集成服务器模式下，报表是通过一个名为报表管理器(Report Manager)的 Web 应用程序进行管理的。报表也可以在自行开发的 Web 应用程序中使用，而且几乎任何 Web 技术或开发工具都可以完成这项任务。

1.4 解决方案类型

Reporting Services 令人印象最为深刻的是：在一个业务环境中可以使用多种不同的方式实现报表。然而，为用户提供大量的选择并不一定能够解决问题。实际上，为用户提供过多的选择方案只会使用户思维混乱并感到恐惧。作为报表系统设计者，我们的任务是为用户提供合理的解决方案，这个解决方案应该简单、不会引起混乱，而且便于使用。Reporting Services 就是这样一个具有多面性的平台，当我们谈到“Reporting Services 报表”一词时，我们必须指明我们到底要表达什么含义。第 V 部分“解决方案模式”针对如何使用最佳实践和解决方案模式来创建和管理报表服务给出了指导性的建议。

针对大多数机构实现的最新 Reporting Services 使用了事实上的基于 Web 的报表管理器界面。当然，也可以将 Reporting Services 集成在一个公司的 SharePoint 网站中。为了满足其他具体业务需要，也可以将报表集成到自定义的应用程序或 Web 页面中，但是这种做法比较少见。实际上，将报表集成在多种自定义解决方案中是相对比较容易的。下面给出了一些可能需要集成报表的软件解决方案：

- 外购且可以马上使用的报表功能，这些功能是基于服务器的，可以使用报表设计人员创建的报表，并且可以部署到一个集中的 Web 服务器上。
- 集成在 Web 应用程序中的报表，可以在一个 Web 浏览器窗口中用 URL 链接打开。
- 使用 SharePoint Web 部件，将报表集成在 SharePoint Services 应用程序中。
- 自行构建的应用程序功能，可以使用程序代码呈现报表。报表可以在一个桌面应用程序或一个 Web 应用程序中打开，也可以保存到一个文件中以备将来查看。
- 使用 Power View 可视化工具，以交互式数据可视化的方式查看通过一个表格语义模型显示的数据。

1.4.1 简单报表设计

如果需要通过创建一个常用的报表类型来汇总或输出数据库中的数据，那么 Reporting Services 可以提供一些非常棒的工具，使得这项工作简单易行。例如，假定你保存了一份客户记录，还保存了这些客户购买的产品清单，现在你需要生成一个客户清单，这个清单包含了交易次数和客户消费总额。你可以使用报表向导生成一个表格报表，这个报表包含了上述信息。如果你需要比较针对每个客户的销售情况，比较方式分为按日比较和按时间段比较，那么就可以使用报表向导生成一个折线图报表，并利用该报表观察销售趋势。这种方法的主要优势在于：使用报表工具及其功能，很容易就可以创建常用的报表类型，无需用户了解过多的复杂信息，例如编写程序、编写查询语句以及构造表达式。

报表解决方案可以划分为信息技术人员创建的报表和业务用户创建的报表两大类。为了支持不同的用户组，人们开发了大量的工具。我们首先对此进行简要介绍，后面给出的表 1-1 将对当前版本的产品和先前版本的产品所提供的报表设计器选项进行总结。

管理一个完整的公司级 BI 解决方案可能是一项复杂且开销巨大的任务。幸运的是，如有必要，一个可行解决方案的全部组件都可以在一台单独的服务器上运行。中小型报表解决方案可以使用一个单独的通用数据库，这个数据库可以充当运营(operational)数据存储和报表数据结构。随着解决方案的不断成熟，这些数据库最终不可避免地要分离开来。一个使用来自运营数据库的数据并以固定时间周期填充数据的小规模数据集市，可以为报表编写提供一个比较简单的数据源，而且这个数据源不会与用户和应用程序争夺系统资源。

就短期使用而言，可以设计和部署简单的报表。稍做规划和管理，就可以设计出满足未来需求的报表。如果进行了合理的设计，报表还可以包含高级功能，这些高级功能不仅能够满足当前的简单需要，还可以满足未来更加复杂的需求。

本书第 II 部分第 5 章“报表设计”首先介绍了基本报表设计的基础内容，然后进一步展示了如何添加比较高级的功能。

1.4.2 IT 专业人士设计的报表

Reporting Services 首次发布之时，报表设计体验主要是针对程序员和应用程序开发人员进行优化的，他们对 Visual Studio 非常熟悉，而 Visual Studio 是用来帮助热爱技术的程序员创建自定义软件的。产品团队完成了第一代产品后，他们马上着手开发一组工具来满足业务用户自行设计报表的需求。这些功能是随着各个版本的产品发布逐步进入市场的。简单了解这段历史不仅有利于更好地了解当前使用的工具集是如何出现的，也有利于了解当前我们可以使用的功能是如何产生的。

对于那些热爱技术的人来说，Visual Studio 提供的报表设计体验(称为 SQL Server Data Tools，即 SSDT)，不仅功能强大，而且让他们感到熟悉(参见图 1-2)。



在先前各个版本的 SQL Server 中，基于 Visual Studio 的设计工具被称为“Business Intelligence Development Studio”(简称 BIDS)。如果我们之中某个 SSRS 怀旧者提起这个工具时使用了这个名称，那么你马上就会明白他指的是 SSDT。我本人也要花点时间来适应这种改变，之后我才会使用这个新名称。

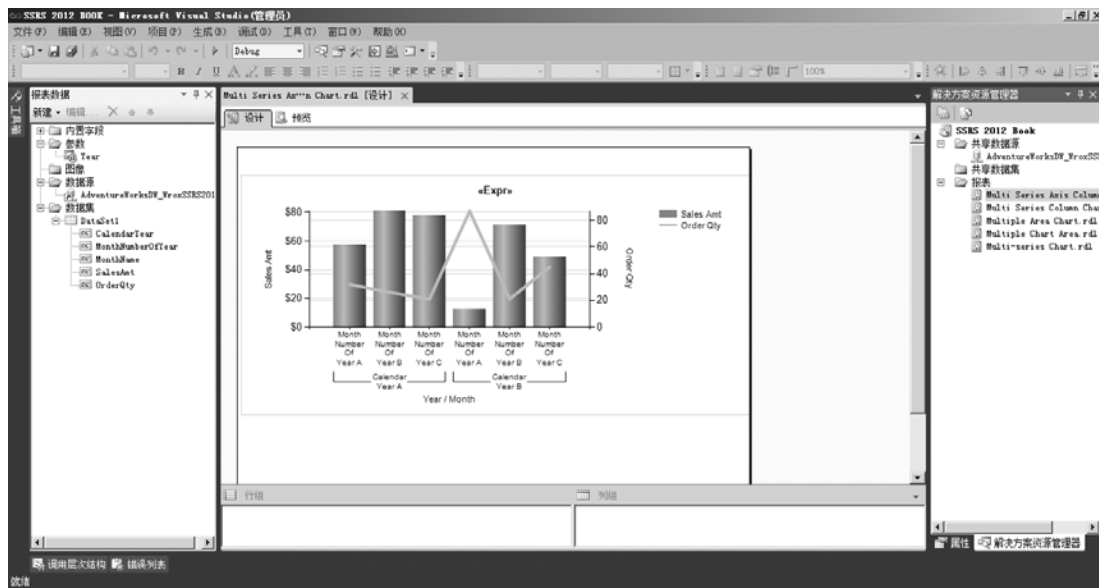


图 1-2

简单报表是很容易设计的，而高级功能则需要使用一系列工具，这些工具大多数应用程序开发者都可以运用自如。与其他 Visual Studio 解决方案类似，报表定义文件都被保存在文件夹中，而这些文件夹可以作为一个部署单元进行管理。

报表、数据源、共享数据集以及所有其他设计元素，都可以使用 SSDT shell 提供的集成版本控制功能进行管理。

许多技术专家同时使用报表生成器和 SSDT 来设计报表。这两种工具都提供了一个简单的报表向导来帮助人们设计常用报表。利用报表向导工具提供的向导提示和属性设置，可以比较方便地构造表格报表、分组报表、矩阵报表和图表报表等内容。

1.4.3 用户设计的报表

SQL Server 2012 为我们提供了两种自助式报表设计工具。长久以来，行业内许多人士都希望能够自助设计报表。在微软公司，为了满足创建完美易用的 BI 工具的需求，开发团队已经开发了多种不同的产品，每一种产品都具有独特的功能。Reporting Services 旗下提供了两种自助式报表工具，可以满足不同的用途。当前出现的报表生成器是基于成熟报表定义架构的，这种报表架构在八年来不断发展成熟。利用多种设计选项，报表生成器不仅可以设计简单的报表，也能设计复杂的报表。Power View 可视化报表工具引入了一种令人激动的动态数据浏览和管理体验。Power View 报表是基于一种表格语义数据模型的，并且提供了一种直观的基于 SharePoint 的交互式设计器，如图 1-3 所示。

SharePoint 环境提供了一种能够管理丰富内容并极具表现力的界面，可以满足多种报表和内容类型的显示。图 1-4 显示了 PowerPivot 库，其中给出了 Power View 报表和 PowerPivot 工作簿缩略图预览。

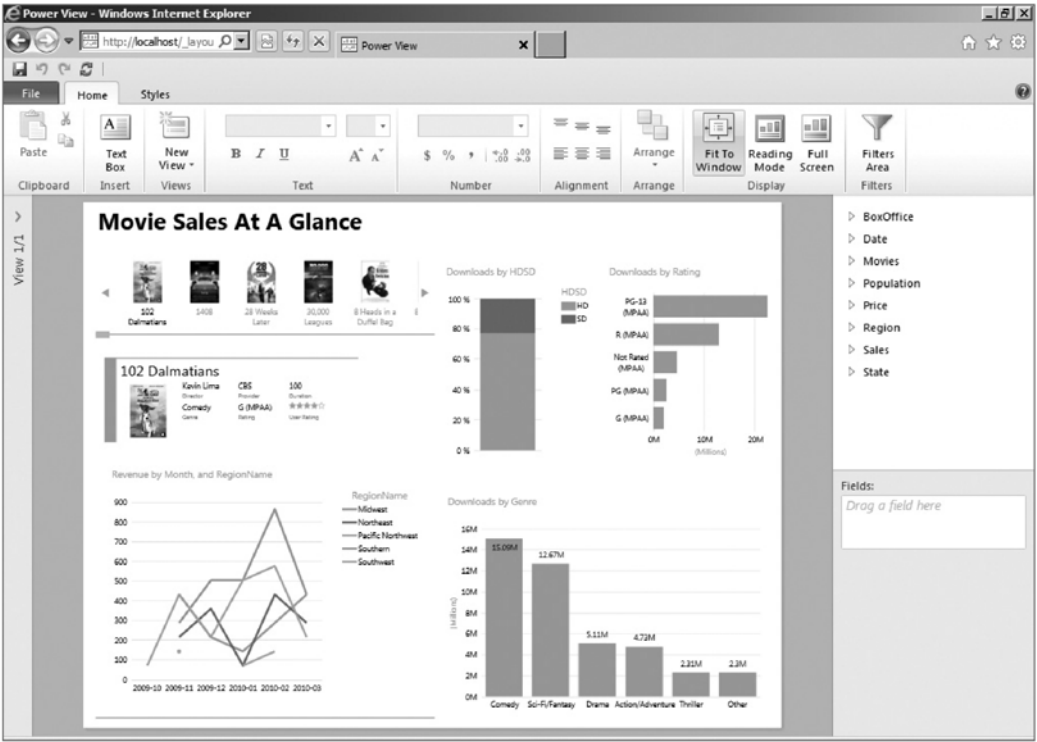


图 1-3

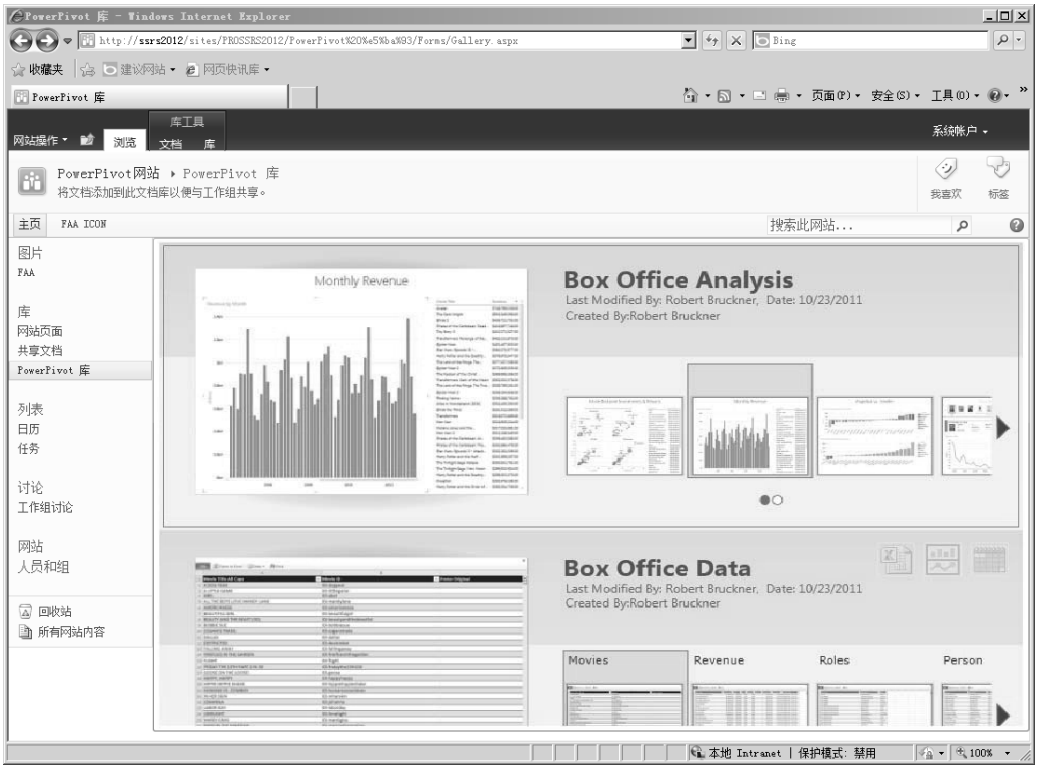


图 1-4

针对报表用户和打算在其机构内部使用这些功能并打算为此提供支持的人士，本书第 12 章和第 13 章将讲解 Power View 工具和表格模型。

SSRS 提供的第一代自助式报表设计工具是目前产品的开端，目前的产品已经比较健壮。Report Builder 1.0 是随着 SSRS 2005 引入的一种基本工具，可以生成功能有限的简单专用报表。在那个年代，Report Builder 1.0 是一项了不起的工具，可以帮助用户通过拖放来自一个语义数据模型的数据项和字段来生成简单的报表。目前，最新版本的报表生成器可以创建与 SSDT 交叉兼容的报表，并且创建的报表可以利用 SSDT 提供的先进功能。Report Builder 1.0 已经成为过去。如果你还在使用它，我强烈建议你改用新的工具集。

微软公司的 2008 产品族引入了 Report Builder 2.0，这项工具不仅适用于业务用户，也适用于技术专家。对一般用户而言，Report Builder 2.0 美观易用。微软公司针对过去各个版本所做的增量式产品改进使得报表生成器成为一种开箱即用的报表设计工具。用户不仅可以编写查询语句，也可以使用公司 IT 部门为他们准备好的数据源和数据集对象，这样他们只需拖放数据项或使用设计向导即可生成报表。在报表生成器中，每个报表都作为一个单独文档进行管理，每个文档都可以部署到报表服务器或 SharePoint 文档库的某个文件夹中。目前，报表生成器不再带有版本号，与先前版本的区别在于：安装该报表工具的 SQL Server 版本就已经确定了当前安装的报表生成器版本。图 1-5 描述了当前最新版本的报表生成器，该版本随 SQL Server 2012 一同安装，并且在设计视图中提供了一个地图报表。



图 1-5

表 1-1 对过去和现在的各个版本的报表设计工具进行了总结。

表 1-1 报表设计器和可视化选项

报表设计器	背 景	现 状
SQL Server Data Tools (SSDT)	实现了 Visual Studio shell，典型用户是 IT 专业人士，设计报表时，需要使用一个报表项目，并且是面向团队的。目前使用 Visual Studio 2010 shell	支持
Report Builder 1.0	随 SSRS 2005 发布，SSRS 2008 和 2008 R2 也对其提供了支持	有限支持，不鼓励继续使用
Report Builder 2.0	随 SSRS 2008 发布	SSRS 2008 支持
Report Builder 3.0	随 SSRS 2008 R2 发布	SSRS 2008 和 2008 R2 支持
Report Builder	随 SSRS 2012 发布	SSRS 2008 R2 和 2012 支持
Power View 可视化工具	随 SSRS 2012 发布 需要 SharePoint Enterprise 版的支持	交互式 Power View 报表比较独特，无法与报表生成器或 SSDT 交叉兼容

1.4.4 基于服务器的报表

报表既可以在一台服务器上运行，也可以在一台客户端计算机的某个独立应用程序中运行。请注意，Reporting Services 首先是针对基于服务器的报表设计进行开发和优化的。客户端选项被称为 Local Mode，只能在某些自定义编程的条件下使用，并且需要付出较多的努力才能实现，而且对专业水平具有一定要求。本章后续部分的讨论仅限于基于服务器的报表设计。

能够理解 SQL Server Reporting Services 和桌面报表工具(如 Microsoft Access)之间的区别是非常重要的。Reporting Services 并不是一个在任何桌面计算机上都能安装的应用程序。设计 Reporting Services 的目的在于为业务人员提供服务。Reporting Services 需要 Microsoft SQL Server 的支持，是一个专业的业务级关系数据库管理工具。同样，报表可以集成到 SharePoint Services 中，以便管理和提供安全保障，并且可以与其他共享的公司文档和资产一并进行管理。

基于上述原因和其他原因，Reporting Services 一般运行在一台服务器上，而不是运行在一台桌面计算机上。无论是在一个小规模业务环境中使用，还是在一个巨型公司中使用，Reporting Services 都表现出良好的可扩展性和适应性。它不仅能够为少数用户所使用，也能够为上千名用户使用，并且可以基于存储在多个不同数据库平台中的大量数据完成报表编写工作。但是，尽管 Reporting Services 是一个业务级的产品，这并不意味着设计报表是一项复杂而又困难的任务。我们不仅可以将业务报表驻留在服务器上，还可以使用 Azure Reporting 报表。Azure Reporting 是微软公司开发的基于云的报表服务器。

报表用户需要连接到网络上，从而与报表服务器相连，此处的网络也可以是因特网。当选中并查看报表服务器或 SharePoint 库中某个文件夹下的一个报表时，这个报表将在用户的 Web 浏览器中显示为一个 Web 页面。作为其他选项，还可以将同一个报表用多种不同的格式进行显示，包括 Word、Excel 和 Adobe PDF 等格式，还可以将一个报表显示为 PNG、JPEG、GIF 和 TIFF 等图像格式。可以将报表保存为上述文件格式以便离线查看、此外，报表还可以保存为其他格式。报表服务器可以将报表以电子邮件的方式定时自动传递，还可以将它保存为文件。这些功能作为标准功能，只需简单配置和很少的用户交互。

1.5 商业智能报表解决方案

一个商业智能(BI)解决方案是构建一个功能强大的业务报表平台的基础。根据需求和业务环境的不同,可能需要设计一个新的数据库。但是,分析业务数据并不意味着必须构建一个完整规模的 BI 解决方案。然而,如果你需要聚集大量数据才能根据关键度量和趋势来分析业务绩效,那么,针对事务处理而设计的关系数据库可能无法满足这项要求。在设计报表之前,充分理解这些核心概念并对 BI 进行投入常常能够减少开销,并帮助你创建一个可以为业务用户和上级领导长期所用的报表编写平台。

我们可以在微软公司提供的报表技术和 BI 技术(也就是 SQL Server 和 SharePoint)中进行选择。后续章节将对此进行讨论。简而言之,一个报表的数据源可以是用来管理事务数据的关系数据库,也可以是一个数据集市,还可以是存储在一个关系数据库管理系统(如 SQL Server 或 Oracle)中的一个数据仓库。针对大规模数据,为了取得更好的分析性能和分析灵活性,可将数据保存在一个语义模型中。这个模型既可以是一个多维度结构(常常称为一个多维数据集),也可以是一个在内存中执行聚集和计算的表格语义模型,图 1-6 解释了一个 BI 解决方案的相对复杂性和成熟度。

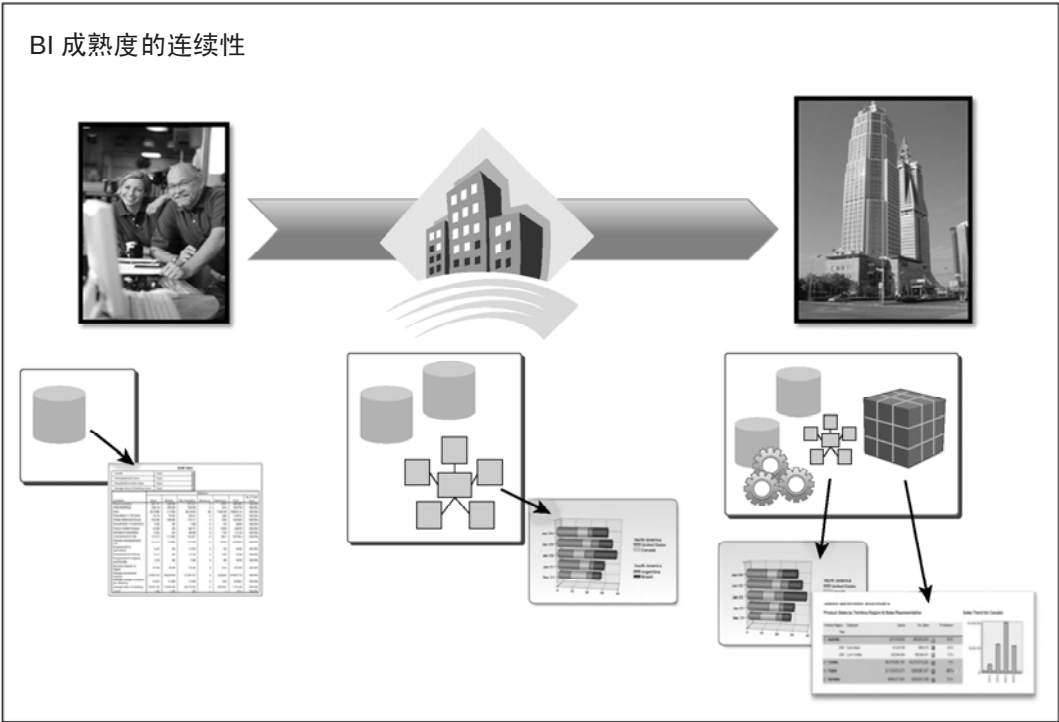


图 1-6

报表编写的工作量或 BI 解决方案的规模并不一定与业务规模成正比。某些情况下,小型业务需要管理大量数据;而在其他某些情况下,大型机构的需求却很简单。关键在于:随着数据量的增加,针对数据存储、管理和分析,必须提出一个最佳方案。

BI 解决方案可以帮助业务领导使用正确的工具,针对自己的业务积极地做出明智的决策。复杂的报表和分析结果可以使信息工作者和领导者具有更长远的眼光,能够看到历史业务数据之外的内

容。通过审查过去和当前的数据，可以找到趋势和模式。通过可靠的业务分析，可以预测未来的趋势，并且可以规划改进业务过程，以此做出明智的决策。

过去，业务工作需要运行报表来跟踪销售总额、票据、库存和生产等简单的事项。作为一项产业，我们已经进入了精于收集数据和存储数据的阶段。为了生成大多数业务报表，需要使用 GB 级甚至 TB 级数据。

我们在整体上并不擅长于将数据转换为有用的、可以作为行动指南的信息。现在，我们需要在全球范围内竞争。业务必须具有较高的效率、富有竞争力和适应性。与以前相比，大型公司需要更频繁地合并、学习、外包、缩减规模并重新校正他们的战略。今天的业务领导必须具有良好的适应性，时刻准备应对行业发展趋势和发展机遇。

在这样一个压力巨大的环境中，以前开发的静态报表应用程序必将被 BI 解决方案代替。BI 不仅仅具有“获取”数据的能力，它还以自助报表工具、仪表板和业务计分板的形式为领导者提供了高级智能功能。当重要事件发生时，或者当某个指标被突破时，它能够主动地提醒用户。

起先，简单报表应用程序可以使用来自一个或两个数据源的数据，但是最终报表应用程序可以基于多个数据源生成报表。可持续发展的 BI 解决方案是基于一致可靠的数据源设计的，这些数据源专门为报表编写进行了工程化设计。为此需要用数据转换软件包将来自多个数据源的数据转换到一个集中式的数据仓库中，然后对其进行处理，生成一个语义模型(OLAP 多维数据集或表格模型)。报表可以使用一个关系数据集市，也可以使用数据仓库或语义模型。可以创建多种类型的报表为业务领导提供支持，并对其重要业务处理决策提供支持。这些决策支持报表可以表现为图表、详情汇总、动态下钻和钻取报表、仪表板和业务计分板等形式。

1.5.1 报表数据源

每个报表至少具有一个数据源和查询，或者具有一个能够返回针对数据值的实体的引用，这个实体被称为数据集(dataset)。运营数据存储常常是最为复杂的数据库。某些整体交付的系统所用的数据库常常包含了上千张表。随着对数据库依赖程度的不断提高和数据驱动的计算机系统不断增多，大多数机构在以下三个方面有所突破：

- 每个数据库的复杂性不断增长，可以容纳更为复杂的处理内容。
- 数据容量不断增长。
- 不同数据库的数目不断增长，从而能够处理不同的业务数据，满足管理要求。

针对中等规模的公司，数据不仅具有复杂性的特点，而且 TB 级数据存储也颇为常见。与几年前同等性能的系统相比，目前存储空间的价格已经不算昂贵。跟踪订单、货运、客户查询、案例和客户都是非常有价值的工作，但是这些工作需要花费时间。记录上述活动意味着你必须保存大量数据来编写报表。将数据保存到数据库中是一项简单的工作，但是从数据中获得有用的智能数据可是一项挑战！最后，为了管理同一类型的数据，需要使用不同的系统，并且需要使用不同的管理方式。例如，一个在销售机构中用来跟踪销售趋势和潜在客户的客户关系管理系统与一个无须支持销售团队的订单管理系统是有区别的。使用这两类系统，都可以跟踪一个名为“客户”的对象，但是在这两类系统中，这个词的定义可能有所不同。“客户”一词，既可以是一个消费者，可以是一份合同，在另一个系统中，也可以是一个公司、领导者或供应商，在其他系统中，还有可能是分销商。大型公司还可能保存这样的记录：这些记录需要在不同系统之间进行复制，例如企业资源计划、人力资源管理、利润、供货商管理、会计和应收应付管理系统等。

某些情况下，大多数解决方案设计者认为，为了从所有这些运营数据源获得有价值的报表度量信息，他们必须将上述信息融入一个简化的集中式数据存储中，这个存储系统是专门设计来支持业务报表需求的。数据仓库系统(data warehouse system)是一个集中式的数据存储，用来将从这些专用的复杂数据源中提取的数据进行标准化。一般情况下，它使用了与运营数据存储相同的关系数据库技术，但是为了简化报表编写过程，数据仓库系统是一个受保护的只读环境。

数据仓库或数据集市都是使用关系数据库技术构造的简化数据结构，是专门针对报表编写而设计的。基于这些数据源编写的报表易于设计，与使用事务性运营数据存储所设计的报表相比，性能更高。许多业务不仅仅需要在报表中列出事务性记录，还需要计算总和。利用来自数据仓库的大规模数据，可以很方便地执行数据聚集，但是深入分析则需要特殊的数据存储技术以及更强大的数据和统计报表引擎。

分析数据源和自助式 BI

在此之前，报表与打印在一张纸上的事务记录没有什么区别，也被称为分类帐、日志与清单。随着人们越来越需要更为有用的信息，报表的复杂性也不断提高。目前，报表不再仅仅是打印在纸张上的记录。用户需要了解其业务的趋势和内部信息。动态报表可以帮助用户研究和调查其业务环境的趋势，用户不再只满足于查看静态清单。随着业务用户的复杂性不断提高，数据和报表介质的复杂度也在提高。为了了解一般的趋势，我们需要从历史研究未来。利用比较精确可靠的数据(包括过去的数据和现在的数据)，利用合适的报表模型，就可以预报和预测趋势及未来的活动。

随着对报表需求的增长，对更复杂的数据的需求也在不断增长。在一个具有一般数据报表需求和少数几个数据源的中型业务机构中，一个小型数据集市或数据仓库就足以满足一个完整 BI 报表解决方案的需求。然而，在一个大型的复杂业务环境中，为了满足深入的报表编写和分析的需求，必须使用一个包含完整的数据提取、转换和加载(ETL)解决方案，还需要数据仓库和商业智能语义模型的支持。语义模型可以帮助我们使用简单的、用户自助式的报表和数据浏览工具进行高级分析。

与 Analysis Services 中的多维数据集类似，表格模型可以用来聚集大规模业务数据。利用自助式报表工具，OLAP 中的多维数据集和模型更加有利于用户探索和浏览。与微软公司为 Excel 定制的 PowerPivot 附件相比，BISM 表格模型是基于相同的技术实现的，该技术可以在内存中完成聚集。可以使用 PowerPivot 附件来创建 PowerPivot 模型，并且可以将 PowerPivot 模型发布到 SharePoint 以便共享，而且可以充当报表数据源。通过使用透视表(pivot tables)和图表，我们可以用 Excel 来浏览这些模型。除了编写独立的报表和浏览多维数据集，标准的 Reporting Services 报表还可以基于多维数据集数据设计报表，为此需要使用一种特殊的查询语言，即多维表达式(Multidimensional Expressions, MDX)。

自助式 BI 解决方案为业务用户提供了数据分析的能力。进行有效的分析需要信息技术部门设计企业 BI 解决方案，而信息技术部门由于资源受限已经感到力不从心，更何况设计企业 BI 解决方案需要特殊的技能和广泛的计划。最近，自助式 BI 解决方案的革新已经填平了这道鸿沟，微软公司的 PowerPivot、表格模型和 Power View 就是这些革新的代表。表格语义模型可以帮助我们收集和聚集大量业务数据以便浏览和编写报表，这一点还带来一项额外的好处，那就是这个模型是完全在服务器上运行的，并且是安全的。表格模型技术实际上使用了 Analysis Services 存储引擎。整个模型被加载到内存中以完成聚集操作，结果返回相当迅速。

表格和多维语义模型都能够为高效的分析报表编写提供独一无二的支持, Reporting Service 可以使用任意一项内容作为报表数据源。

1.5.2 复杂性和报表性能

系统性能常常是一个高效的 BI 解决方案最为重要的指标之一。随着一个机构中的报表需求越来越复杂, 报表规模越来越大, 费用常常首先是用性能进行度量的。查询需要花费很长的时间执行, 并且需要与报表和数据库服务器的资源竞争。在这种情况下, IT 专业人士将会意识到一个简化数据库的价值, 并且认识到简化数据库的必要性。无论报表使用的是企业级数据仓库, 或者是一个部门级的数据集市, 抑或是一个简单的“报表结构”, 基本的概念一般都是相同的, 就是对数据库设计进行简化, 关注报表编写的需求。

前面已经提到, 某些性能需求和高级分析需求推动了解决方案的成熟, 使得解决方案采用了 OLAP 多维数据集。但是这并不意味着所有基于其他数据源设计的报表都要进行修改。很多报表在使用运营数据源或关系型数据仓库的情况下仍然能够很好地工作。但其他更复杂的报表就需要特殊的数据源(如 OLAP 多维数据集)才能很好地工作。

我还记得在项咨询任务中, 我开发的报表使用了复杂的财务公式, 为此使用了原始数据库结构作为报表数据源。当时编写的 T-SQL 查询语句颇为复杂, 而且难以调试。但是通过优化, 一个更为复杂的报表仅花费了 45 分钟就运行完毕, 而这个报表原先执行一次查询就需要 90 分钟之久, 因此, 这让客户大吃一惊。而当我们同样的数据转换为一种简化的数据集市结构之后, 运行同一个报表仅仅花费了不到 3 分钟的时间。利用 Analysis Services 的 OLAP 多维数据集, 同一个报表运行完毕仅需几秒钟。无需多言, 一旦用户发现运行一个报表只需几秒钟, 先前“可接受”的 45 分钟报表呈现时间就变得不可接受了!

这种情况下, 设计一个数据仓库或一个 OLAP 数据所需的费用就成为重点考虑的内容, 该费用必须低于糟糕的报表性能和不必要的长时间报表开发周期所增加的费用。SQL Server Analysis Services 是一种令人印象深刻并极具吸引力的技术, 即使对中小型业务也具有极大的价值。如果你没有使用过个工具集, 而且基本报表已经难以满足你的需求, 那么我们强烈建议你认真了解一下这个令人印象深刻的工具集, 你必将对它大加赞赏。

1.6 自定义报表体验

可以使用多种方式传递报表。报表不仅只有用户实时浏览一种交付方式。可以在服务器缓存中对报表自动进行呈现, 这样可以让用户迅速打开报表, 而且不会对数据源造成负担。可以根据预设的时间表, 通过电子邮件和共享文件来传递报表。使用数据驱动的订阅技术, 可以在用户非联机时段将报表向大量听众进行广播。每个用户都可以收到一份报表, 而且这些报表可以用不同的格式呈现的, 还可以是以不同的筛选数据进行呈现的。你将学习如何规划、管理和配置这些功能。

你还将学习如何优化、备份、恢复报表服务器数据库、Web 服务和 Windows 服务。你还要学习如何使用管理工具、配置文件和日志等对服务器环境进行定制, 并避免发生诊断问题。

1.7 小结

本章为本书的其他部分提供了一个语境，并介绍了相关的概念、技巧和功能。你将在后续章节中学习这些内容。

本书第 I 部分各章节通过介绍 Reporting Services 工具的安装和配置，将为你提供入门指导。你将学到如何在 Native 模式下安装 Reporting Services，还将学到如何在与 SharePoint 2010 集成的模式下安装 Reporting Services。

本书第 II 部分的 5 个章节的内容将指导你一步步地从一个报表设计新手成长为一位报表高手。在讨论基本报表设计时，你将使用向导和简单工具来创建典型的业务报表，如表格、矩阵和列表，以及简单的图表。你将学习到如何使用推荐的技术为每个报表设计布局和格式。你还将为 SQL Server 及其他数据源设计查询，并将学习基本的 T-SQL 来创建筛选数据集和参数化数据集。高级报表设计讨论了更具有吸引力的报表设计功能，还讨论了如何使用具有创造性的方法，通过使用动态格式化和操作，使报表更加具有活力。我们将用一章的篇幅研究图表报表设计，学习掌握先进的可视化图表功能，这项功能非常有用。

第 III 部分和第 IV 部分将探讨 SQL Server 2012 和 SharePoint 2010 引入的多项 BI 创新，包括：语义表格模型和 Power View 可视化工具。这两部分内容还将介绍如何在多维数据集和 MDX 查询基础上设计报表。无论你是在 SharePoint 模式还是在 Native 模式下使用 Reporting Services，你都将学习到如何使用报表生成器和自助式报表编写工具为业务用户提供支持，并通过便于使用的浏览和设计工具回答重要的业务问题。

SQL Server BI 和报表编写平台为你提供了许多工具和选项，其功能之强大，颇为惊人。第 V 部分将基于我们多年的解决方案设计工作和成功的报表项目，以指南的方式供你查阅。针对报表解决方案架构、设计和自定义编程技术，我们给出了一系列最佳实践。第 VI 部分将充分探讨报表服务器管理和内容管理。你将学习到如何将 Reporting Services 与 SharePoint 进行集成，还将学习到如何管理一个在 Native 模式下运行的报表服务器，以及如何对可靠性、安全和最优性能进行监视和优化。

第 VII 部分是为那些打算将报表集成到一个自定义解决方案中的程序员，以及能够在设计报表过程中使用自定义查询代码的高级报表设计人员编写的。你将学习到如何使用表达式和自定义函数来扩展报表能力，以及如何通过增加报表功能来扩展应用程序。

第 2 章

Reporting Services 安装和架构

本章内容

- 安装报表服务器
- 构建企业部署
- 使用工具管理报表生命周期
- 探索报表服务器架构
- 利用报表服务扩展

为了熟悉 Reporting Services，开发者和管理员常常要在一台个人计算机或一台开发服务器上执行一次基本安装。尽管基本安装掩盖了企业部署中的很多关键选项，但是基本安装仍然为我们提供了一个探索 Reporting Services 功能和安装过程本身的环境。这样一个环境对运行联机帮助文档和本书提供的练习和教程是一个理想的环境。

本章首先指导你如何完成 SQL Server 2012 Reporting Services 的基本安装，然后回顾与企业部署有关的重要考虑事项。

本章还将探讨如何实现和显示 Reporting Services 中的功能。对管理员和开发者来说，这些显示的信息是非常关键的。这些概念是后续章节的基础。

报表生命周期给出了 Reporting Services 工作的语境。本章将探讨与 Reporting Services 相关的各种应用程序和工具软件。然后，你将稍微深入地研究 Reporting Services 本身，为此将研究 Reporting Services Windows 服务架构，Reporting Services 组件以及支持数据库。在本章最后，你将牢固地掌握所有这些内容是如何作为一个整体来完成 Reporting Services 功能的。

本章包含以下内容：

- 基本安装
- 企业部署时需要考虑的事项
- 报表生命周期
- Reporting Services 工具
- Reporting Services Windows 服务
- Reporting Services 处理程序和扩展

- Reporting Services 应用程序数据库

2.1 基本安装

为理解 Reporting Services 的安装，必须理解 Reporting Services 的组件。在 SQL Server 2012 中，Reporting Services 具有两种模式：

- Native 模式
- SharePoint 集成模式

Reporting Services 的核心内容是一个 Windows 服务，该服务依赖于以一个 SQL Server 数据库引擎实例为宿主的两个数据库。请注意，在 SharePoint 集成模式中，SQL Server 2012 中的 Reporting Services 是作为一个 SharePoint 共享服务运行的。本章主要关注 Reporting Services 的 Native 模式安装。第3章将详细研究 SharePoint 集成模式安装和配置的细节和差异。

与 Reporting Services 服务进行交互需要通过应用程序完成，例如 Native 模式下的报表管理器，它以 Reporting Services 为宿主；另外在 SharePoint 模式下的 Reporting Services 外接程序也能完成这项任务；此外还有其他安装在客户系统上的应用程序，如 Business Intelligence Development Studio。本章将介绍这些应用程序和服务，还将介绍 Reporting Services 数据库。

当使用 Native 模式的基本安装时，服务器端和客户端的组件都被安装在一个单独系统中。Reporting Services 数据库也被安装在 SQL Server 数据库引擎的一个本地实例中。在无须依赖于其他系统的情况下，基本安装也常常被称为独立安装。

一般情况下，基本安装使用的是 SQL Server 软件的 Developer 版本或 Evaluation 版本。这两个版本都能够访问 Reporting Services 的全部功能。SQL Server 的 Evaluation 版是免费的，但是只能使用 180 天。Developer 版的价格非常低，但是仅限于在非生产环境中使用。这两个版本的 SQL Server 都可以通过微软网站下载，也可以通过订购服务或软件销售商获得。

通过使用这两个版本，能够让用户以比较低廉的价格甚至以免费方式使用 Reporting Services 的全部功能。此外，与其他版本的 SQL Server 相比，这两个版本的 SQL Server 还能够支持更多的操作系统，包括 Windows Server 2008、Windows Server 2008 R2，各种版本的 Windows 7 和 Windows Vista。

其他系统需求包括 1GB 内存和 1.4GHz(32 位或 64 位)处理器。如果将系统更新和 SQL Server 示例所需硬盘空间计入，基本安装还至少需要 6GB 硬盘空间。

以 Hyper-V 角色运行在 Windows Server 2008 SP2(含更高版本)的 Standard 版、Enterprise 版和 Datacenter 版上的虚拟机环境也支持 SQL Server 2012。

2.1.1 安装 Reporting Services

在执行 Reporting Services 安装前，有必要确保系统已经使用最新的 Service Pack 进行了升级更新。你还必须属于待安装计算机的操作系统本地 Administrators 组的成员，否则你必须使用属于本地 Administrators 组的成员账号的凭据运行安装程序。

开始安装时，首先要访问 SQL Server 2012 Evaluation 版或 Developer 版安装介质。安装介质可能是 DVD，也可能是本地驱动器或文件共享驱动器上的安装文件。安装介质必须能够从打算安装软件的系统中进行访问。执行安装介质根目录下的 SETUP.EXE 文件，就启动了安装程序。

首先,安装程序将检查计算机上是否存在 Microsoft .NET Framework 3.5 SP1 和 Windows Installer 4.5。如果没有找到 Microsoft .NET Framework 3.5 SP1 和 Windows Installer 4.5,那么安装程序将初始化安装这两个软件。

安装 Windows Installer 的过程(显示为一个修补程序)相当快,但是安装.NET Framework 所花费的时间要比安装对话框中显示的时间和安装进程要长得多。此处我们没有给出这些组件的安装步骤,但是其安装过程与一般的微软公司软件安装过程是一样的。当安装程序完成安装.NET Framework 或 Windows Installer 之后,系统可能需要重新启动。系统重新启动后,还需要重新启动 SQL Server 2012 安装程序。

图 2-1 显示了安装程序的 SQL Server 安装中心。安装中心被分为几个页面,每个页面所访问的文档和工具都对安装过程的不同方面提供了支持。



图 2-1

为了完成基本安装,请单击“SQL Server 安装中心”表单左方的对应链接。在图 2-2 所示的安装页面中,请选中“全新 SQL Server 独立安装或向现有安装添加功能”选项。这时将启动 SQL Server 安装向导。



图 2-2

SQL Server 安装向导执行的第一个步骤是把你所用的系统与一组“安装支持”规则进行比较。这些规则检查系统配置是否满足安装要求。分析结束后，向导会显示汇总信息。如果出现不满足安装要求的情况，将会显示规则列表，指出需要注意的事项。如果没有不满足安装要求的情况发生，就可以单击“显示详细信息”按钮，此时可以看到如图 2-3 所示的列表。



图 2-3

单击安装程序支持规则页面上的“查看详细报表”链接，此时将出现如图 2-4 所示的新窗口，这个窗口列出了详细检查结果，显示了全部警告和未能满足的规则。查看完毕后，可关闭此窗口。



图 2-4

在 SQL Server 安装向导的查看详细报表页面上，单击“确定”按钮，转入如图 2-5 所示的“产品密钥”页面。选择 SQL Server 免费版本，或者输入其他某个版本的序列号。选择 Evaluation 版本或输入 Developer 版本的序列号，继续下一步工作。



图 2-5

单击“下一步”按钮，进入如图 2-6 所示的“许可条款”页面。仔细阅读产品许可的各项内容。为了继续安装，选中“我接受许可条款”复选框。下面的功能使用情况复选框可以用来确定是否将那些与硬件和 SQL Server 组件使用有关的高级信息发送给微软公司，以便帮助微软公司改进产品。单击功能使用情况复选框下方的超链接，可以查阅隐私声明。与功能使用情况有关的内容包括：Reporting Services 是否已经安装以及操作系统是 32 位的还是 64 位的。与功能使用情况有关的数据收集量是非常小的，并且不再细分数据粒度，也不涉及某项功能区域使用次数等信息，而是只收集某项功能是否曾经使用过。



图 2-6

单击“下一步”按钮，进入如图 2-7 所示的“安装安装程序文件”页面。这个页面提示安装文件即将被安装到计算机中。安装过程结束后，向导将进入下一个页面。

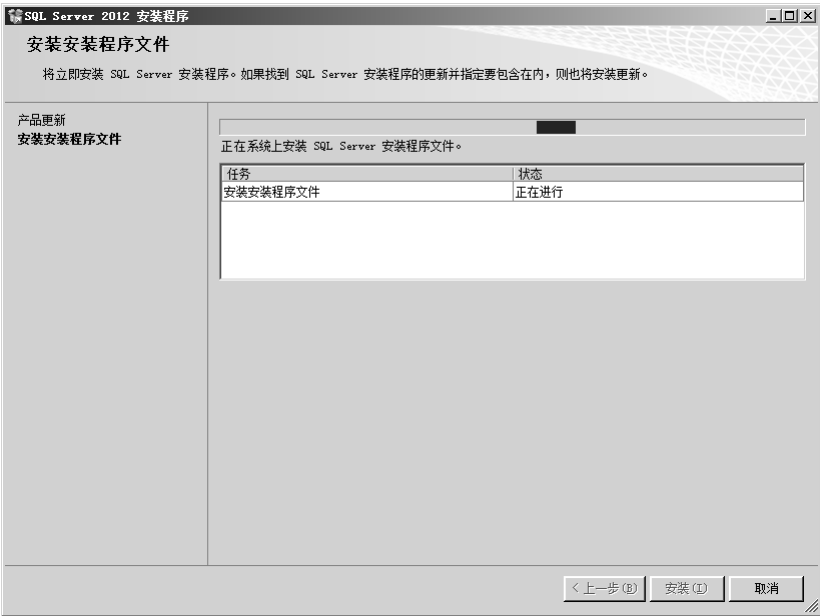


图 2-7

支持文件安装完成后，向导进入了另一个安装程序支持规则页面。在这个页面中，需要根据不同的规则集确认系统配置。与前面的页面一样，单击“查看详细报表”链接可以获取更多信息。在继续下一步安装之前，应该查看全部的警告信息和未满足条件的信息。

单击“下一步”按钮，进入“设置角色”页面。在这个页面中，请选中“SQL Server 功能安装”选项，如图 2-8 所示。



图 2-8

单击“下一步”按钮，进入“功能选择”页面，在这个页面中选中打算安装的 SQL Server 产品和功能，如图 2-9 所示。针对基本安装，请选中“Reporting Services 一本机”和“数据库引擎服务”。此外还要选中“SQL Server Data Tools”、“客户端工具连接”、“文档组件”以及管理工具(同时选中基本和完整)。如果打算安装其他组件，例如 Analysis Services，那么还要选中这些组件。



图 2-9

在“功能选择”页面中，可以修改安装共享组件的路径。基本安装一般都使用默认安装路径。如果必须改变安装路径，那么请单击安装位置旁边的按钮，然后选择一个合适的安装位置。

单击“下一步”按钮，进入如图 2-10 所示的“实例配置”页面。在此你可以发现在前一个页面中选中的数据库引擎的名称和 Reporting Services 实例。页面的下半部分列出了系统中已经安装的其他 SQL Server 实例。如果已经安装了一个默认实例，那么可以在这个默认实例上完成当前安装；否则必须提供一个合适的实例名。



图 2-10

为实例命名时，必须牢记实例名是区分大小写的，并且在系统中必须唯一。实例名不得超过 16 个字符，可以包含字母、数字、下划线(_)和美元符号(\$)。第一个字符必须是一个字母，而且实例名不能是联机丛书所给出的 174 个安装保留字中的内容。此外，建议不要用 235 个 ODBC 保留字中的内容对实例进行命名，联机丛书给出了这 235 个 ODBC 保留字。

在“实例配置”页面中，可以将路径修改为待安装的特定实例组件的路径。与前面一样，一般情况下，我们使用默认路径。如果必须修改路径，那么请单击路径下方的按钮，然后选择一个合适位置。



在“实例配置”页面中，除了可以输入实例名，还可以输入一个安装 ID。实例 ID 用来标识安装目录和 SQL Server 实例的注册表键。一般来说，除非有充分的理由，否则不需要修改实例 ID。

单击“下一步”按钮，进入如图 2-11 所示的“磁盘空间要求”页面。在这个页面中，可以查看安装各个组件所需的磁盘空间。



图 2-11

单击“下一步”进入“服务器配置”页面。这个页面包含两个选项卡：一个是“服务账户”，另一个是“排序规则”。默认选项卡为“服务账户”。

在如图 2-12 所示的“服务账户”选项卡中，可以选择用来安装每个服务的账户。针对基本安装，一般推荐使用本地服务账户，或使用针对数据库引擎和 Reporting Services Windows 服务(生成的)的网络服务账户。第 19 章将介绍如何在安装结束后修改服务账户。

在如图 2-13 所示的“排序规则”选项卡中，可以修改实例使用的排序规则(collation)。默认选择取决于本地操作系统的区域配置。与其他选项一样，除非理由充分，否则不推荐修改其默认值。



图 2-12



图 2-13

单击“下一步”按钮，进入“数据库引擎配置”页面。在这个页面中，可以配置安装 Reporting Services 的 SQL Server 数据库引擎实例。这个页面包含三个选项卡：“服务器配置”、“数据目录”以及 FILESTREAM。

在图 2-14 所示的“服务器配置”选项卡中，单击“添加当前用户”按钮，这时当前用户就被设置为数据库引擎实例的管理员。除非必要，否则这个选项卡中的其他选项均不需修改。



图 2-14

在如图 2-15 所示的“数据目录”选项卡中，可以修改数据库引擎实例使用的多个路径。再次说明，除非确有必要，否则这个选项卡中的其他选项均使用默认值。



图 2-15

在如图 2-16 所示的“FILESTREAM”选项卡中，选中“针对 Transact-SQL 访问启用 FILESTREAM”选项。针对本书的示例数据库，必须选中这个选项。随后我们将安装这个示例数据库。如果在安装过程中忘记设置这个选项，那么就必须使用 SQL Server 配置管理器修改 SQL Server 数据库引擎服务的 FILESTREAM 属性，为此需要使用 sp_configure 将配置选项 filestream access level 更新为一个适当的值。本步骤中其他与配置数据库引擎有关的细节请参考联机丛书。



图 2-16

单击“下一步”按钮，进入如图 2-17 所示的“Reporting Services 配置”页面。在这个页面中，可以从三个选项中选择一项内容。本章后半部分将详细讨论不同选项的含义。对大多数基本安装来说，应该选择“Reporting Services 本机模式”下面的“安装和配置”安装选项。后面的安装与使用均假定已选中这个安装选项。



图 2-17

单击“下一步”按钮，进入如图 2-18 所示的“错误报告”页面。在这个页面中用户可以根据需要参与反馈活动。尽管我们鼓励用户参与信息反馈，但是选择哪个选项并不影响安装。



图 2-18

单击“下一步”按钮，进入如图 2-19 所示的“安装配置规则”页面。这个页面用于在继续安装过程之前，根据当前选中的选项检查是否存在问题。与前面一样，单击链接“查看详细报告”将打开一个单独的报表。



图 2-19

单击“下一步”按钮，进入如图 2-20 所示的“准备安装”页面。认真查看选中的选项。如果你打算在其他系统中重复上述安装过程，那么可以复制页面底部列出的 INI 文件路径。将来如果需要执行基于命令行的安装，那么可以使用这个 INI 文件，具体请参考本章 2.2.6 节的描述。



图 2-20

单击“安装”按钮，软件随即开始执行安装过程。安装过程需较长的时间才能完成。在这段时间里，我们可以看到安装进度页面。安装过程结束后，可以看到如图 2-21 所示的页面，此时可以看到安装过程汇总信息。



图 2-21

单击“关闭”按钮，结束向导并返回安装中心。现在可以关闭安装中心，也可以使用安装中心的安装页面中的“搜索产品更新”选项来查看修补程序或服务包。

安装结束后，最后一步需要对安装进行验证。为此打开 Internet Explorer 浏览器并输入以下 URL：

- 如果安装了一个默认实例，那么输入 `http://localhost/reports`。
- 如果安装了一个命名实例，那么输入 `http://localhost/reports_instancename`，并用实例名替换 `instancename`。

如果是安装结束后第一次输入上述 URL，那么 IE 浏览器将花费一段时间才能解析上述 URL。最终 IE 将返回如图 2-22 所示的页面，这表明服务已经成功安装完毕。如果看不到如图 2-22 所示的新建文件夹、新建数据源、报表生成器、文件夹设置、上载文件和详细信息视图等选项卡，或者如果看到了用户权限不足的提示，那么请以 Administrator 身份重新加载 Internet Explorer。

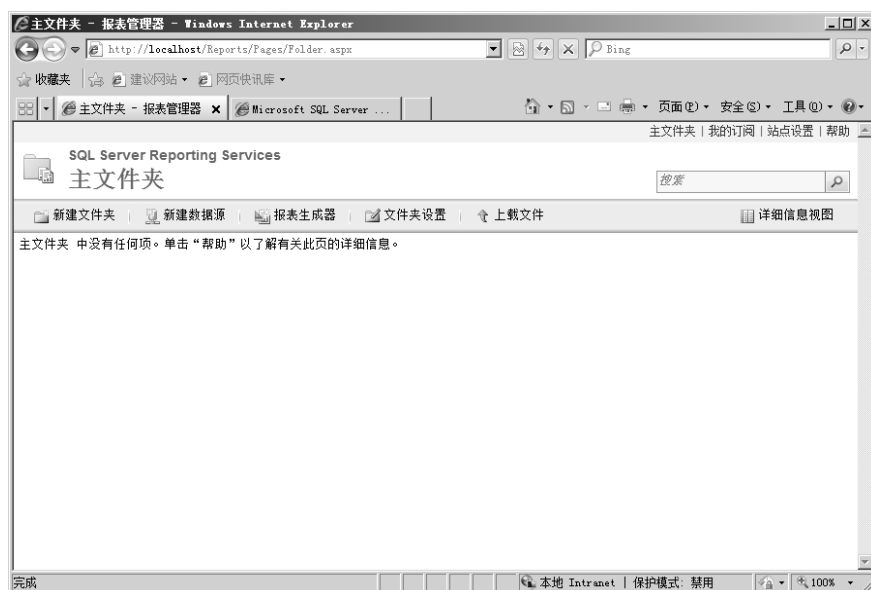


图 2-22

2.1.2 安装 Reporting Services 示例和 SQL Server 示例数据库

成功安装 Reporting Services 软件之后，现在需要安装 Reporting Services 示例和 SQL Server 示例数据库。本书自始至终都需要使用上述示例和数据库。此外，在联机丛书中，多个教程也使用了上述示例和数据库。

Reporting Services 示例包含多种应用程序、扩展、模型、报表以及脚本。这些示例可以帮助你学习 Reporting Services 中方方面面的内容。某些情况下，这些示例还可以充当开发生产性应用程序的起点。

本书给出的示例数据库主要包括了两个数据库，它们均与一个虚构的自行车生产商 Adventure Works Cycles 公司有关。数据库 AdventureWorks_WroxSSRS2012 是一个典型的事务系统结构的数据库，而数据库 AdventureWorksDW_WroxSSRS2012 则是一个典型的分析系统结构的数据库。这两个数据库可以帮助你同时熟悉运营报表和分析报表。

可以通过 Wrox 出版社网站 www.wrox.com 下载 Reporting Services 示例和示例数据库。如果你打算下载上述内容，那么针对示例数据库和 Reporting Services 示例，你必须根据当前使用的平台下载对应的安装文件。

在安装示例之前，请确保 SQL Server 数据库引擎和 Reporting Services 已经正常运行。然后启动下载完毕的 MSI 程序文件，根据指令完成示例安装。

2.2 企业部署

正如本章开始部分所言，基本安装忽略了 Reporting Services 企业部署中的许多重要内容。其目的在于避免一开始难度太大。但是如果需要规划如何在企业环境中安装、配置和发布 Reporting Services，你就必须认真考虑以下内容：

- SQL Server 版本
- 命名实例
- 拓扑结构
- 模式
- 安装选项
- 命令行安装

2.2.1 SQL Server 版本

SQL Server 2012 提供了多个版本，以下版本包含了 Reporting Services：

- Enterprise 版
- Business Intelligence 版
- Standard 版
- Developer 版
- Evaluation 版
- Web 版
- Express with Advanced Services 版

Enterprise 版、Business Intelligence 版、Standard 版和 Web 版都可以支持生产环境。其中，Enterprise 版和 Business Intelligence 版支持访问 Reporting Services 的全部功能，这些版本既可在 32 位平台上运行，也可在 64 位平台上运行。Standard 版和 Web 版只能访问一个受限的功能集。与 Enterprise 版和 Business Intelligence 版相比，Standard 版和 Web 版费用较低，比较适合规模较小的安装环境。

Developer 版和 Evaluation 版能够访问 Enterprise 版的全部功能。Developer 版价格低廉，专门用作开发和测试环境。Evaluation 版是免费的，但在 180 天试用期后将无法使用。与 Enterprise 版和 Standard 版相比，这两个版本能够支持更多类型的操作系统。

Web 版支持一个受限的功能集，甚至比 Standard 版支持的功能还要少，因此适合小规模部署或基于 Web 的部署。最后，Express with Advanced Services 版是一个功能高度受限的 SQL Server 版本，只能有限地支持 Reporting Services。这个版本是免费的，但是因为受限过多，因此除了少数非常特殊的场合，这个版本几乎不会使用。这个版本既支持 32 位平台，也支持 64 位平台。

如果你需要详细了解每个版本的系统需求和支持的功能，请参考联机丛书。针对上述各个版本的 Reporting Services，表 2-1 对核心功能和硬件需求进行了详细对比。关于操作系统的详细比较，请参考联机丛书。

表 2-1

支持的功能	Enterprise 版、Business Intelligence Evaluation 版和 Developer 版	Standard 版	Web 版	Eadvanced 版
Reporting Services Windows 服务	是	是	是	是
SharePoint 集成模式	是	是		
可扩展拓扑模式	是			
报表数据警告	是			
Power View	是			
基于角色的安全	是	是	支持但受限	支持但受限
自定义安全扩展	是	是	是	是
支持导出 Word、Excel、PDF 和图像	是	是	是	是
远程数据源和非关系数据源	是	是		
数据源、传递和数据呈现扩展	是	是		
报表传递	是	是		
报表历史、计划、订阅和缓存	是	是		
数据驱动的订阅	是			
SSDT 报表设计器	是	是	是	是
报表生成器	是	是		
报表管理器	是	是	是	是
服务器内存(最低要求)	1 GB	512 MB	512 MB	256 MB (32 位) 512 MB (64 位)
服务器内存(推荐值)	2+ GB	2+ GB	2+ GB	1+ GB
服务器内存(最大可用内存)	无限制	64 GB	64 GB	4 GB
支持的 CPU 架构	32 位 64 位	32 位 64 位	32 位 64 位	32 位 64 位
一个单独的 Reporting Services 实例所用的 CPU 内核最大数目	无限制	16 核	16 核	4 核
CPU 速度(最低要求)	1.4 GHz	1.4 GHz	1.4 GHz	1.4 GHz
CPU 速度(推荐值)	2.0+ GHz	2.0+ GHz	2.0+ GHz	2.0+ GHz

2.2.2 命名实例

在单独一台服务器上可以安装多个 Reporting Services 实例。每个 Reporting Services 实例都可以独立于其他实例运行，同时可以具有不同的版本。每一个实例都拥有自己的 Windows 服务与代码库，以及与其交互的 Reporting Services 数据库对(pair)。这些数据库可以运行在单独的 SQL Server 数据库引擎实例上，也可以运行在一个共享实例上，前提是每个数据库事先的命名是唯一的。

为了区分一台服务器上的不同 Reporting Services 实例，必须为每一个实例命名，这个名称在此系统中是唯一的，称为实例名(instance name)，而一个被命名实例称为一个命名实例(named instance)。

在一台服务器上，除了命名实例，还可以存在未命名的实例。这个实例也称为默认实例(default instance)。如果在一台服务器上只安装了一个实例，那么这个实例常常是一个默认实例。

可以在单独的一台服务器上运行多个实例，这些实例既可以全部是命名实例，也可以既有命名实例也有默认实例。在服务器硬件数量有限的情况下，可以采用多实例方法将 Reporting Services 从 SQL Server 2008 或 SQL Server 2008 R2 迁移到 SQL Server 2012 中，这是一种比较实用的方法。如果需要在部署中尽可能地减少许可证的数目，那么多实例就是一种非常好的方法。也就是说，我们推荐将一个单独的 Reporting Services 实例(既可以是一个命名实例，也可以是一个默认实例)部署在一台生产服务器上，这样可以获得最优的资源分配，还能够使总体稳定性达到最佳。

2.2.3 拓扑结构

拓扑结构(Topology)是指在为用户提供一种统一的服务功能访问界面的前提下，如何在服务器之间分配 Reporting Services 组件。拓扑结构关注的是 Reporting Services Windows 服务和 Reporting Services 数据库，而不是客户端工具。Reporting Services 支持两种拓扑类型，即标准拓扑(standard)结构和扩展拓扑结构(scale-outs)。

在标准拓扑中，Reporting Services Windows 服务被安装在一个系统中。Reporting Services 需要与一对 Reporting Services 数据库进行交互，这两个 Reporting Services 数据库既可以安装在本地系统中，也可以安装在远程系统中，并且是这个 Reporting Services 实例专用的数据库。本章开始部分所介绍的基本安装就是一个标准拓扑示例。

在 Native 模式下，当采用扩展拓扑时，Reporting Services Windows 服务的多个实例安装在多台服务器上。在 SharePoint 集成模式下，当采用扩展拓扑时，Reporting Services 服务则是作为一个共享服务运行在 SharePoint 场中的多个结点上。

对扩展拓扑结构来说，无论使用哪种模式，Reporting Services 实例都会共享一对 Reporting Services 数据库。通过共享数据库，扩展拓扑中的每一台运行 Reporting Services 服务的服务器(称为一个结点)都可以访问同样的内容，具有同样的安全配置。如果网络上提供了负载平衡的硬件或软件，那么对最终用户而言，扩展拓扑中的某些结点或全部结点都将展现为一个单独的资源，但是与标准部署相比，具有更为灵活的处理能力。可以将扩展拓扑结构中的其他结点配置为专门用于根据计划处理报表，这就减轻了环境中其他结点的处理负担。

如果需要在标准拓扑结构和扩展拓扑结构之间进行抉择，那么务必注意，只有 Enterprise 版才能支持扩展拓扑结构。为了安装扩展拓扑结构，需要在完成标准安装之后进行额外的配置。

最后，如果是为了提高可用性而安装扩展拓扑结构，那么就必须考虑在一个失效转移集群上实现 Reporting Services 数据库。请牢记：尽管 SQL Server 数据库引擎支持失效集群，而且 Reporting Services 能够与运行在一个集群上的数据库进行交互，但是 Reporting Services 服务本身并不具备任何集群功能。

2.2.4 模式

Reporting Services 既可以运行在 Native 模式下，也可以运行在 SharePoint 集成模式下。在 Native 模式下，Reporting Services 使用自己内部(或称“本机”)的功能管理自己的内容。

使用 Enterprise 版、Developer 版、Evaluation 版或 Standard 版进行部署的 Reporting Services 可以运行在 SharePoint 集成模式下。在 SharePoint 集成模式下，内容管理是通过 SharePoint 完成的，本机内容管理界面(如报表管理器)是不可访问的。第 3 章将详细讨论 SharePoint 集成模式。

对许多希望使用 SharePoint 作为其企业内容管理解决方案的组织机构来说, SharePoint 集成模式是非常有吸引力的。这一点对那些使用了 Power View 提供的分析和可视化新功能的组织机构来说, 尤其如此。然而, 集成模式存在一些限制, 例如, 该模式不支持链接报表。

对那些希望在 Native 模式下运行 Reporting Services, 同时又希望通过 SharePoint 展示 Reporting Services 内容的组织机构而言, Reporting Services Web 部件可以提供一种能够替代 SharePoint 集成模式的解决方案。第 18 章将会讨论相关的 Web 部件。

2.2.5 安装选项

在安装过程中, 可以看到三个 Reporting Services 配置选项。既可以在 Native 模式下使用默认配置安装 Reporting Services, 也可以在 SharePoint 集成模式下使用默认配置安装 Reporting Services, 还可以在一个名为 files only 的最小配置模式下安装 Reporting Services。

只有在一个安装过程中同时安装了 Reporting Services 和数据库引擎的情况下, 才能在 Native 模式下使用默认配置安装 Reporting Services。在完成了全部安装工作之后, 这些安装选项使 Reporting Services 处于工作状态。当然, Reporting Services 安装完毕后, 不是所有的 Reporting Services 功能都已经安装成功(例如, 此时尚未配置执行账户和电子邮件发送选项)。如果使用 SharePoint 集成模式安装 Reporting Services, 那么还必须执行附加的安装步骤才能完成配置工作。第 3 章将介绍与此有关的内容。

对企业级部署来说, “Native 模式-仅安装” 安装模式是最常用的选项。利用“仅安装”选项, 安装 Reporting Services 的过程仅仅是安装了服务器组件, 但是并未进行配置。安装结束后, 为了使服务正常工作, 需要使用 Reporting Services 配置工具来配置 Reporting Services 数据库, 并且需要为 Reporting Services Web 服务配置 URL 和 Report Manager。第 19 章将讨论这些内容。

2.2.6 命令行安装

对大规模企业部署来说, 建议使用基于脚本文件的命令行安装, 而不要使用本章前面介绍的界面驱动安装。通过在一个组织机构中使用一个通用的脚本安装文件, 可以确保在这个组织机构中安装的 Reporting Services 具有高度的一致性。

为了执行命令行安装, 请使用/CONFIGURATIONFILE 参数从命令行启动 SETUP.EXE 应用程序。/CONFIGURATIONFILE 参数要求提供包含了安装指令的脚本文件路径。可以使用一个文本编辑器创建这个脚本文件。脚本文件内容请参考联机丛书的有关内容, 也可以通过本章第一部分介绍的交互式安装过程创建脚本文件。如果通过一个交互式安装过程创建了这个脚本文件, 那么就必须检查参数设置, 并进行适当的修改。

下面给出了从命令行调用 SETUP.EXE 的方法:

```
SETUP.EXE /CONFIGURATIONFILE=c:\temp\configuration.ini
```

以下是 INI 文件的指令内容。使用基于文件的配置, 这些内容可以用来指导 SETUP 安装 Reporting Services 默认实例:



```
[SQLSERVER2012]
QUIET=True
ACTION=Install
FEATURES=RS
INSTANCENAME=MSSQLSERVER
RSINSTALLMODE=FilesOnlyMode
```

```
RSSVCACCOUNT="NT AUTHORITY\NETWORK SERVICE"  
RSSVCSTARTUPTYPE=MANUAL
```

代码片段 101117 codedownload0201.txt

2.3 报表生命周期

一般认为，报表生命周期包括三个阶段。在编写阶段，人们设计和开发报表；在管理阶段，最终用户可以访问报表；在传递阶段，报表被传递到用户手中。图 2-23 解释了这三个阶段，后续章节将讨论这三个阶段。

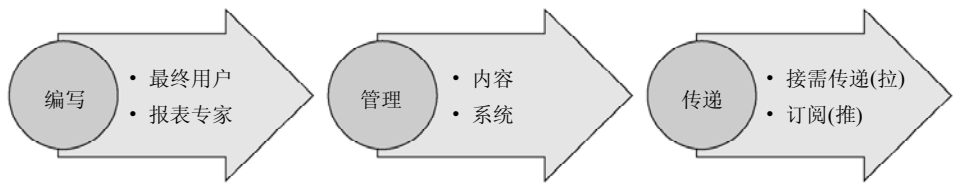


图 2-23

2.3.1 编写

报表生命周期的编写阶段(authoring phase)首先要通过正式和非正式的过程收集需求。这些需求决定了如何设计查询语句，用于为报表提供数据的查询。数据与图表、表格、矩阵和其他表达元素集成在一起，构成了基本的报表。然后，通过格式化和布局调整，生成一个报表草稿。随后，根据需求，验证这个报表草稿的准确性和一致性。只有通过了验证，才能将报表发布到一个集中的管理系统中以便最终用户使用。

报表编写是由两类工作人员进行处理的：

对最终用户作者来说，开发报表是他们的副业。在一个组织机构中，这些人往往属于非 IT 部门，他们倾向于使用技术水平要求不高但是用户友好程度较好的报表编写工具。这些工具能以便于解释的方式展示数据，并且能够与报表设计相结合。这些作者用一种比较简单的方式来设计报表布局并对报表进行格式化，甚至使用自动化任务完成上述工作。

报表专家关注报表开发，这也是他们的主要工作内容。这些专家通常在 IT 部门工作。报表专家需要精确控制查询和报表设计。他们使用的编写工具需要使用者具有比较好的技术背景，能够访问报表系统提供的全部功能。

当然，不是所有的报表编写人员都可以归纳到这两类人员。最终用户作者和和报表专家代表了报表编写人群谱系的两端。许多报表编写人员在学习时，会倾向于向某一端发展。为了满足所有报表编写人员的广泛需求，必须实现多种报表开发工具。

2.3.2 管理

在报表生命周期的管理阶段(management phase)，需要对发布的报表进行组织和安全管理，并且需要针对最终用户访问进行配置。这个阶段还要对多个报表和特殊功能所需的资源进行配置，例如订阅传递和缓存。这些活动一并称为内容管理(content management)，常常需要由报表编写者和管理员共同完成。

为了确保报表管理系统能够持续工作，需要对报表管理系统进行配置和不断维护。系统管理活动常常是管理员的工作领域，这个领域的工作具有排他性。

2.3.3 传递

完成了报表部署和报表配置工作后，最终用户就可以在报表生命周期的传递阶段(delivery phase)使用报表了。最终用户既可以根据需要来查看报表，也可以要求按照事先定义好的计划向他们传递报表。上述两种情况分别被称为报表传递的拉(pull)方法和推(push)方法。报表传递成功的关键在于灵活性。

2.4 Reporting Services 工具

Reporting Services 支持完整的报表生命周期，这项任务是基于 Reporting Services 提供的一组工具完成的，表 2-2 列出了这些工具。

表 2-2

编 写	管 理	传 递
报表设计器	报表管理器	报表管理器
Power View	SharePoint 报表库	Power View
报表生成器	Reporting Services 配置管理器	SharePoint 库和 Web 部件
第三方编写工具	SQL Server 管理应用程序	Report Viewer 控件
	命令行工具	Reporting Services Web 服务
		订阅

2.4.1 报表设计器

报表设计器提供了完整的报表开发功能，为报表专家精确控制报表提供了支持。这个应用程序可以通过 SQL Server Data Tools(SSDT)进行访问，SSDT 是一组专业设计工具，可在 Visual Studio 中使用。SSDT 是随 SQL Server 一同安装的，并且与已安装的 Visual Studio 集成在一起。如果 Visual Studio 2010 尚未安装，那么 SQL Server 将安装一个 Visual Studio shell，SSDT 可以从 Visual Studio shell 中运行。

报表设计器的界面中包含两个标签，即“设计”选项卡和“预览”选项卡。这两个选项卡支持查询开发、报表布局和格式化，以及验证。通过使用报表设计器，可以访问一组向导和对话框，这些向导和对话框能够支持高度定制化的复杂报表。后续章节将深入研究这些功能。

2.4.2 Power View

Power View 可以帮助最终用户编写具有高度交互性的报表和实现可视化效果(visualizations)，并且可以帮助最终用户深入研究数据。

Power View 是 SQL Server 2012 最新引入的 Reporting Services 报表工具。Power View 面向数据用户，帮助他们以可视化方式研究数据，方便地解答自己提出的问题。它还能够帮助他们在 SharePoint 环境中展示数据并与其他人进行协作。Power View 提供了一种瘦浏览器的体验，并且是基于 BI 语

义模型工作的。利用基于 Excel 的 Power Pivot 的建模工具，可以在客户端使用 BI 语义模型；还可以通过 SQL Server 2012 支持的 Analysis Services 表格项目，在服务器端使用 BI 语义模型。第 9 章将更为详细地讨论 BI 语义模型。

2.4.3 报表生成器

报表生成器是一个报表设计器，实现的功能类似于 SSDT 报表设计器。但是报表生成器具有 Microsoft Office 的观感。它可以单独下载，无须使用报表服务器，并且可以作为一个服务器安装的组成部分。

2.4.4 第三方编写工具

Reporting Services 报表是以 XML 文档格式保存的。Reporting Services 使用的 XML 被称为报表定义语言(Report Definition Language, RDL)。报表编写工具为报表开发提供了图形界面。因此，报表作者无须了解底层 RDL 文档的详情。

RDL 是一个开放标准，因此第三方组织开发了自己的报表编写工具，这些工具可以生成 RDL 文档。Reporting Services 可以使用这些 RDL 文档。

2.4.5 报表管理器

报表管理器主要是一个内容管理工具，支持用户通过一个直观的、基于文件夹的导航结构访问报表。因为报表管理器可以方便安全地访问报表，所以也常被当作一个报表传递应用程序。第 17 章将介绍报表管理器。

必须指出，只有当使用运行在 Native 模式下的 Reporting Services 实例时，才能使用报表管理器。对于运行在 SharePoint 集成模式下的实例，内容管理和报表显示功能是通过 SharePoint 提供的。如需了解更多信息，请参考本书第 3 章的内容。

2.4.6 SharePoint 库和 Web 部件

对于运行在 SharePoint 集成模式下的 Reporting Services 实例来说，报表和其他 Reporting Services 项为标准 SharePoint 库的组成部分，并且是作为 SharePoint 内容进行管理的。Report Viewer Web 部件是在 SharePoint 集成模式下安装 Reporting Services 的过程中安装的，可以通过 SharePoint 展示运行在 SharePoint 集成模式下的 Reporting Services 实例中的报表。

当通过 SharePoint 访问 Reporting Services 内容时，我们不仅能够访问运行在 SharePoint 集成模式下的实例，还可以使用一个较旧版本的 Reporting Services SharePoint 2.0 Web 部件展示运行在 Native 模式下的实例中的内容。利用 Report Explorer 2.0 和 Report Viewer 2.0 Web 部件，可以在一个 SharePoint 网站中展示由报表管理器进行管理的报表内容，还可以在一个 SharePoint 网站中展示在 Native 模式实例中呈现的报表内容。第 18 章将详细介绍如何集成报表与 SharePoint。

2.4.7 Reporting Services 配置管理器

Reporting Services 配置管理器可以帮助你访问关键的系统设置。此外，这个工具为某些管理任务提供了支持，例如创建 Reporting Services 应用程序数据库和备份与恢复加密密钥。第 19 章将讨论这些任务，还将介绍如何使用 Reporting Services 配置管理器来执行这些任务。

2.4.8 SQL Server 管理程序

因为 Reporting Services 是 SQL Server 产品族的成员之一，因此标准的 SQL Server 管理程序同样支持 Reporting Services。利用 SQL Server Management Studio，可以执行多种管理任务，包括管理共享计划和角色。为了配置 Reporting Services Windows 服务，需要使用 SQL Server 配置管理功能，当然，其功能与 Reporting Services 配置管理器存在重叠之处。第 17 章和第 19 章将讨论如何使用 SQL Server Management Studio 完成多种管理任务。

2.4.9 命令行工具

为将管理任务自动化，Reporting Services 提供了一组命令行工具。表 2-3 描述了每种工具及其默认安装位置。

表 2-3

工 具	说 明	默认保存位置
Rs.exe	执行 VB.NET 脚本，将管理任务自动化。这个工具可以用于没有运行在 SharePoint 集成模式下的 Reporting Services	<drive>\Program Files\Microsoft SQL Server\110\Tools\Binn\rs.exe
Rsconfig.exe	修改 Reporting Services 数据库的连接信息，并且可以在没有提供凭据的情况下，设置 Reporting Services 用来连接数据源的默认执行账户	<drive>\Program Files\Microsoft SQL Server\110\Tools\Binn\rsconfig.exe
Rskeymgmt.exe	管理 Reporting Services 使用的加密密钥。这个工具还可以用来将一个 Reporting Services 安装与其他 Reporting Services 安装联接起来，构成一个“扩展(scale-out)”部署	<drive>\Program Files\Microsoft SQL Server\110\Tools\Binn\rskeymgmt.exe

2.4.10 HTML Viewer

Web 浏览器(如 Microsoft Internet Explorer)是最为流行的 Reporting Services 报表查看工具。大多数情况下，当使用 HTML 格式呈现一个报表时，Reporting Services 可以通过加入 JavaScript 来提供一些交互式功能，包括工具栏、文档地图、固定表格页眉以及表格排序等。总之，所有这些基于脚本的功能都被称为 HTML Viewer。

为了保证与 HTML Viewer 兼容，建议使用最新版本的 Internet Explorer。目前，升级为最新服务包并启用了脚本的 Internet Explorer 8.0 和 Internet Explorer 9.0 均可保证正常运行 HTML Viewer 的全部功能。

其他 Web 浏览器也可以查看呈现为 HTML 的 Reporting Services 报表，如 Firefox、Chrome 和 Safari。如果打算向那些没有使用当前版本 Internet Explorer 的用户发布 Reporting Services 报表，就必须知道各种浏览器都为哪些功能提供了支持，详情请参阅联机丛书。

2.4.11 Report Viewer 控件

Report Viewer 控件可以用来在自定义应用程序中显示 Reporting Services 报表。Report Viewer 控件实际上包括了两个控件，一个用于 Web 应用程序，另一个用于 Windows Forms 应用程序，它们具有同样的功能。



请不要将 Report Viewer 与 SharePoint Report Viewer Web 部件相混淆。Report Viewer 2.0 Web 部件用于支持在 SharePoint 中显示 Reporting Services 的内容。

Report Viewer 控件可以在两种模式下运行，其中，默认模式为 Remote Processing 模式。在 Remote Processing 模式下，报表是由一个 Reporting Services 实例进行呈现的，并且是通过控件显示出来的。这种模式是优先选择的模式，因为这样既可以使用 Reporting Services 的完整功能集，还可以充分利用 Reporting Services 服务器的处理能力。

在无法使用 Reporting Services 服务器的情况下，或者如果数据必须直接通过客户端进行检索，那么 Report Viewer 控件还可以在 Local Processing 模式下运行。在 Local Processing 模式下，应用程序可以检索数据并将数据与报表定义进行配对，在无须 Reporting Services 服务器支持的情况下，在主机系统上生成一个呈现完毕的报表。当 Report Viewer 控件在 Local Processing 模式下执行时，无法使用 Reporting Services 的全部功能。

第 20 章将介绍如何通过 Report Viewer 控件将报表与自定义应用程序集成在一起。

2.4.12 Reporting Services Web 服务

为了满足特定应用程序的集成需求，Report Viewer 提供了一个 Web 服务，利用这个 Web 服务可以管理和交付报表。如表 2-4 所示，这个 Web 服务提供了一些端点(endpoint)，通过这些端点，可以访问多种可编程类。

表 2-4

端 点	说 明
ReportExecution2005	支持以可编程方式访问 Reporting Services 报表处理和呈现功能。在 Native 模式下和 SharePoint 集成模式下，这个端点均可使用，但是需要使用不同的 URL
ReportService2010	支持以可编程方式访问 Reporting Services 报表管理功能。在 Native 模式和 SharePoint 集成模式下，这个端点均可使用
ReportService Authentication	当 Reporting Services 运行在 SharePoint 集成模式下，并且 SharePoint 被配置为 Forms 身份验证时，这个端点为用户身份认证提供支持

第 20 章将讨论如何使用 Reporting Services Web 服务。联机丛书提供了每个端点的完整文档。

Reporting Services Web 服务提供的一项特殊功能是 URL 访问。利用这项功能，可以通过简单地调用一个 URL 来检索一个已呈现报表。通过在 URL 查询字符串中设置参数和呈现选项，可以获得不同内容的报表。第 20 章也将介绍 URL 访问。

订阅

利用订阅，可以按照事先定义的计划将报表发送到用户手中，也可以在某个事件发生(如数据发生更新)时，将报表发送到用户手中。Reporting Services 支持两类订阅：

标准订阅。标准订阅能够以某种具体的格式，使用事先定义的参数值呈现报表，然后将报表发送到一个预先设定的位置。这类订阅可以满足许多报表用户的需求。这种订阅方式为报表用户就如何、何时与何处查看报表提供了充分的自由。

数据驱动的订阅。这种订阅方式更为灵活，更适合将报表发送给大量具有不同需求的用户。为了建立这类订阅，需要引用一个自定义关系表，这个自定义关系表中保存了每个报表接收者的记录。其中每个记录可以指定报表呈现和传递选项，还指定了报表参数值。利用数据驱动的订阅，可以根据报表使用者的具体需求来裁剪一个订阅。

默认情况下，订阅传递仅限于电子邮件发送或文件共享发送。通过集成自定义传递扩展，订阅还可以支持其他传递选项。第 22 章将介绍自定义传递扩展。

2.5 Reporting Services Windows 服务

通过研究都有哪些作者、管理员及最终用户与 Reporting Services 交互，前面各节对应用程序进行了深入观察。本节将研究 Reporting Services 服务本身的基本架构。

在 Native 模式下，Reporting Services 服务是一个 Windows 服务。在 SharePoint 集成模式下，与以前的版本相比，SQL Server 2012 能够更为深入地将 Reporting Services 与 SharePoint 集成在一起。Reporting Services 服务是直接作为 SharePoint 的组成部分来运行的，并且是作为一个共享服务运行的。第 3 章对此将进行更为详细的介绍。

与服务的交互是通过 HTTP 和 WMI 接口完成的。HTTP 接口提供了对 Reporting Services 核心报表管理和交付功能的访问，WMI 接口提供了在 Native 模式下对服务管理功能的直接访问。在 SharePoint 集成模式下，SSRS 2012 将服务配置直接集成到了 SharePoint Configuration 页面中。外部配置文件和应用程序数据库都支持服务。图 2-24 描述了这些接口和功能。

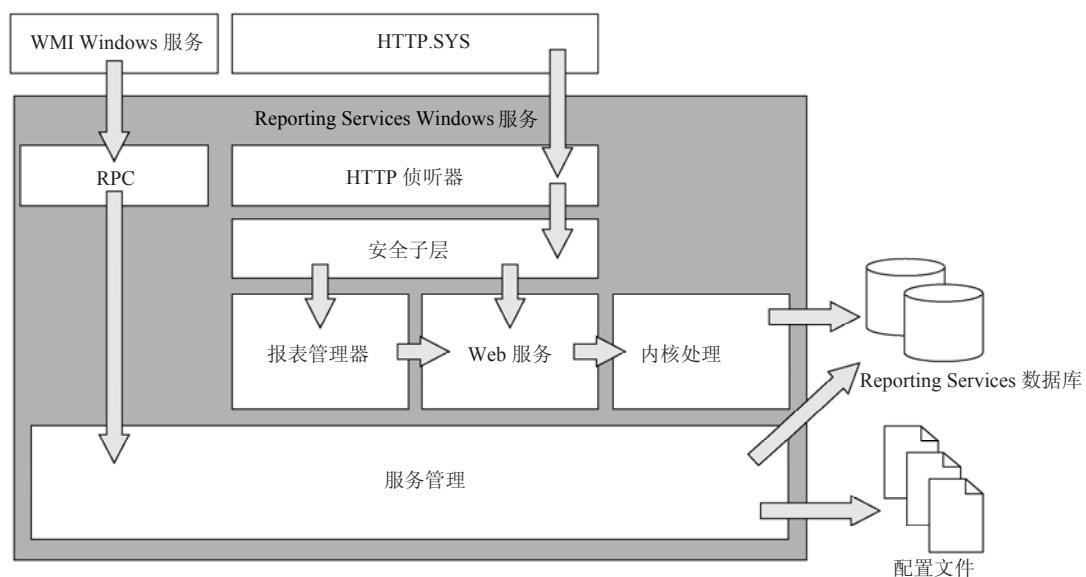


图 2-24

后面各节将研究 Reporting Services Windows 服务中以下方面的内容：

- HTTP.SYS 和 HTTP 侦听器
- 安全子层

- 报表管理器和 Web 服务
- 内核处理
- 服务管理
- 配置文件
- WMI 和 RPC 接口

2.5.1 HTTP.SYS 和 HTTP 侦听器

当一个 HTTP 请求被发送到 Native 模式下的 Reporting Services 服务器时, 请求首先是由服务器操作系统通过 HTTP.SYS 驱动程序接收的。HTTP.SYS 负责管理与请求的连接, 并且将 HTTP 通信路由到服务器适当的应用程序中。

Reporting Services 注册表中记录的 URL 提供了 HTTP.SYS 用来与 Reporting Services 路由连接通信的指令。Reporting Services Windows 服务中的 HTTP 侦听器功能从 HTTP.SYS 处收到重新路由的请求后, 由 Reporting Services Windows 服务中的报表管理器或 Web 服务应用程序接手处理工作。

Native 模式下的 Reporting Services 不需要使用微软公司的 Web 服务程序 Internet 信息服务(IIS)。这样就简化了 Reporting Services 的安装和管理。

尽管 Reporting Services 并不依赖于 IIS, 也不与 IIS 交互, 但是如果需要使用 IIS, 仍然可以在 Reporting Services 服务器上运行 IIS。只要二者记录的 URL 保留项(URL reservation)不存在冲突即可, Reporting Services 和 IIS 甚至可以通过同一个 TCP 端口进行通信。



唯一例外是 IIS 5.1 和 Reporting Services 无法在 32 位 Windows XP 平台上共享 TCP 端口。如果正好使用了这个配置, 那么就需要修改 URL 保留项才能使用不同的 TCP 端口号。可以按照第 19 章描述的方法, 用 Reporting Services 配置管理器修改 Reporting Services 保留项。

2.5.2 安全子层

收到请求后, HTTP 侦听器将把请求转发给 Reporting Services 安全子层。安全子层负责确定请求者的身份, 并且判断针对该请求, 请求者是否具有适当的权限。这些步骤称为身份验证(authentication)和授权(authorization)。

SharePoint 集成模式下的 Reporting Services 使用 SharePoint 网站身份验证机制, 详情请参考第 3 章内容。

Reporting Services 安全子层是通过一个名为安全扩展(security extension)的组件实现的。这个扩展处理身份验证和授权机制, 并且为 Reporting Services 提供了一个标准的接口集以便调用。Reporting Services 可以使用多种安全扩展, 但在部署 Reporting Services 时, 一次只能配置使用一种安全扩展。

Native 模式下的 Reporting Services 预先配置了 Windows 集成的安全扩展。这个扩展可以基于用户的 Windows 凭据对用户进行身份验证, 并且支持四种交换凭据的机制, 称为身份验证类型。

如果域支持 Kerberos 功能, 那么 Kerberos 是最佳身份验证机制。Kerberos 具有很高的安全性。如果启用了委派(delegation)和模拟(impersonation), 那么当最终用户查询一个外部数据源的时候, 可以使用 Kerberos 让 Reporting Services 模拟最终用户。

NTLM 使用了一种质询-响应(challenge-response)机制对最终用户进行身份验证。这种机制也是安全的,但是仅限于不支持模拟和委派的情况下使用。

Negotiate 身份验证类型是 Windows 集成的安全扩展的默认身份验证类型。使用此身份验证类型时,优先使用 Kerberos。如果无法使用 Kerberos,则使用 NTLM。

Basic 身份验证是最不安全的身份验证方式。当使用 Basic 身份验证时,在客户端和 Reporting Services 之间会以明文方式传递用户凭据。如果使用了 Basic 身份验证,那么最好考虑实现一个安全套接字层(Secure Socket Layer, SSL)证书,用来加密 HTTP 通信。

默认的 Windows 集成的安全扩展并不适合所有的场合。通过因特网传递报表,将 Reporting Services 功能与使用了自定义安全机制的应用程序集成在一起,它们是两种常见的使用场景。在这样的使用场景中,可能无法使用 Windows 集成的身份验证。在这类场合下,可以开发自定义安全扩展,并在 Reporting Services 中应用这些自定义安全扩展。第 22 章将讨论自定义扩展。

无论使用的是默认安全扩展还是自定义安全扩展,一旦获得了用户标识,就必须马上验证用户是否拥有执行一个请求操作的权限(实际事件过程序列是:用户通过了身份验证,请求则被直接或间接发送给 Web 服务,Web 服务随后回调安全扩展,完成授权)。与微软公司的其他产品类似,在 Reporting Services 中进行授权是基于角色的。创建角色时,同时需要将执行系统级和项级任务的权限分派给这个角色。随后需要为用户指派角色,从而确定一个用户是否被授权执行一个请求的任务。

2.5.3 报表管理器和 Web 服务

通过 HTTP 发送的所有请求最终都被发送到报表管理器或 Web 服务应用程序。2.4 节讨论了这些应用程序的功能。

就当前讨论内容而言,关键是要理解这两个 ASP.NET 应用程序(也就是报表管理器和 Reporting Services Web 服务)都是以 Reporting Services Windows 服务为宿主的(但是并不依赖于 IIS,前面已经讨论过)。这两个应用程序都是在自己的应用程序域中执行的,这样就使得 Windows 服务可以将这些应用程序当作独立的应用程序进行管理(尽管报表管理器的功能依赖于 Web 服务)。这样的好处是一个应用程序内部的问题可以被隔离在应用程序域的内部。在解决这个应用程序域中的问题实例时,Windows 服务可以重新启动一个新的应用程序域。

2.5.4 核心处理功能

Reporting Services 的核心处理功能(包括计划、订阅管理、传递和报表处理)都是由一组以 Reporting Services 服务为宿主的组件完成的。尽管这些组件并不依赖于 ASP.NET,但是这些组件是作为服务内部的独立应用程序域进行管理的。2.6 节将更详细地探讨这些组件。

2.5.5 服务管理

我们已经详细地介绍过 Reporting Services。为了确保资源可用且服务正常工作,Reporting Services 还实现了一组内部服务管理功能。尽管这些功能并非一个单独的实体,但是在总体上我们可以将其视为一个服务管理子层。

这个子层的一项关键功能是应用程序域管理。前面曾提到,报表管理器、Web 服务和核心处理功能都是以 Reporting Services Windows 服务为宿主的,并且使用了三个单独的应用程序域。这些应用程序域偶尔也会出现问题。服务管理子层的应用程序域管理功能可以监视出现的问题,并且将回收那些受到影响的程序域。这样就确保了 Reporting Services Windows 服务的整体稳定性。

这个子层另一项关键功能是内存管理。报表处理需要消耗大量内存。Reporting Services 服务监视内存使用情况，并且做出适当的反应。在必要的情况下，可以临时将大型请求从内存中移动到硬盘上，这样小型请求可以不受影响并继续执行。这些任务主要是在内存使用受限的情况下，通过动态内存分配和磁盘缓冲来完成。第 19 章将介绍 Reporting Services 的内存管理模型。

配置文件

Reporting Services 内部功能和外部功能都是由一组保存在配置文件中的参数进行控制的。配置文件(Configuration file)是一组事先定义好结构的 XML 文档，其中保存的信息规定了 Reporting Services Windows 服务中各个组件的行为。表 2-5 列出了最为关键的配置文件。

表 2-5

配 置 文 件	说 明	默 认 位 置
Reporting ServicesService.exe.config	这个配置文件中的信息影响了 Reporting Services Windows 服务的跟踪和日志	<drive>:\Program Files\Microsoft SQL Server\MSRS11.<instancename>\Reporting Services\ReportServer\Bin
RSReportServer.config	这个配置文件中的设置影响了 Reporting Services 的多个方面。对 Reporting Services 功能来说，这是 Reporting Services 的主要配置文件	<drive>:\Program Files\Microsoft SQL Server\MSRS11.<instancename>\Reporting Services\ReportServer
RSSrvPolicy.config	这个配置文件中的设置为 Reporting Services 扩展规范了代码访问安全策略	<drive>:\Program Files\Microsoft SQL Server\MSRS11.<instancename>\Reporting Services\ReportServer
RSMgrPolicy.config	这个配置文件中的设置为报表管理器规定了代码访问安全策略	<drive>:\Program Files\Microsoft SQL Server\MSRS11.<instancename>\Reporting Services\ReportManager

2.5.6 WMI 和 RPC 接口

微软公司开发的 Windows Management Instrumentation(WMI)技术能够以一种始终一致的方法管理运行在 Windows 平台上的设备和应用程序。Reporting Services Windows 服务通过本地 WMI Windows 服务注册了两个类，从而向 WMI 开放服务。WMI 服务可以向用于管理的应用程序提供这些类的属性和方法。

Reporting Services 注册的第一个类是 MSReportServer_Instance，它提供了 Reporting Services 安装的基本信息，包括版本、版本编号以及安装模式。

Reporting Services 注册的第二个类是 MSReportServer_ConfigurationSetting，它支持访问 RSReportServer.config 配置文件中的许多配置项。用于管理的界面(如 Reporting Services 配置管理器工具)都可以利用这个类完成自己的任务。



开发人员也可以利用这些 WMI 接口，当然也可以使用其他 WMI 接口。主要的难点在于如何了解 WMI 内部的命名空间组织方式。微软公司 Web 网站提供了 WMI Code Creator 工具。这是一个非常优秀的工具，可以帮助开发人员探索 WMI 命名空间和 WMI 向外界提供的属性和方法。

远程过程调用(remote procedure call, RPC)接口可以充当 WMI 和 Reporting Services 服务之间的一座桥梁。通过这座桥梁, WMI 服务接收到的所有针对注册类的调用, 都将被转发给 Reporting Services。

2.6 Reporting Services 处理程序和扩展

在 2.4 节中, 我们了解到 Reporting Services Windows 服务的内部情况。服务的核心处理功能是一个应用程序域进行介绍的, 其功能是通过一组组件完成的。本节将研究这些组件, 从而帮助我们理解 Reporting Services 如何发布自身的主要功能以及如何扩展这些功能有一个更为深入的理解。

在开始研究具体组件之前, 你必须知道扩展(extensions)和处理程序(processors)之间的区别。处理程序是 Reporting Services 组件架构中的协调器和促进器, 它们根据需要调用扩展, 并且为扩展提供数据交换机制(参见图 2-25)。尽管可以通过配置修改处理程序的行为, 但是自定义代码是不能对处理程序进行扩展的。

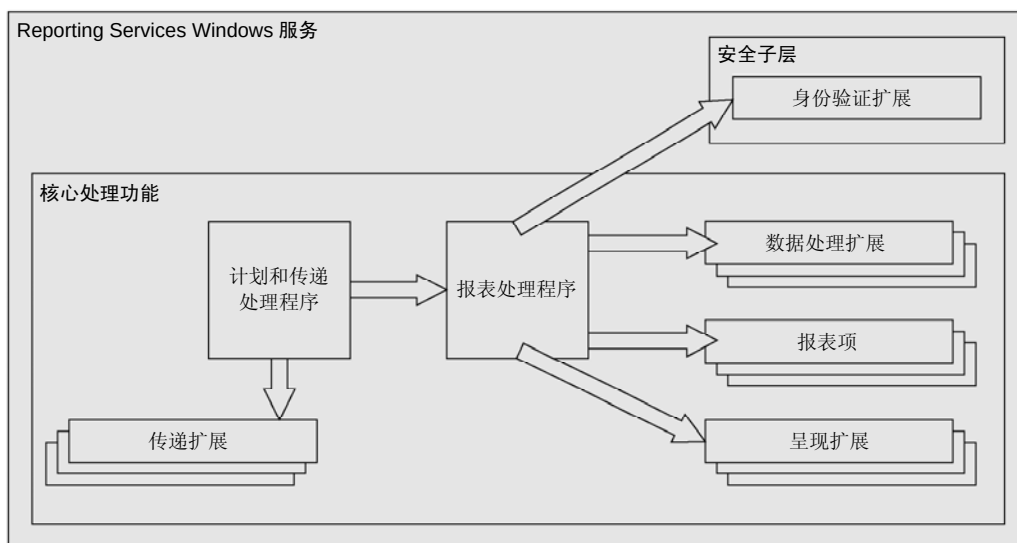


图 2-25

扩展是 Reporting Services 注册的组件, 可以提供具体的功能。扩展可以提供标准的接口, 完成 Reporting Services 指定的任务。

了解上述概念后, 我们再学习以下内容:

- 报表处理程序(Report Processor)
- 数据处理扩展
- 报表项
- 呈现扩展
- 计划和传递处理程序
- 传递扩展

安全扩展已经在本章 2.5.2 节中讨论过。

2.6.1 报表处理程序

报表处理程序可以通过组合数据和布局指令生成一个报表。当一个访问报表的请求到达之后，处理程序可以完成以下工作：

- (1) 调用安全扩展，对请求进行授权。
- (2) 从 Reporting Services 数据库中提取报表定义。
- (3) 将报表定义中的数据提取指令传递给数据处理扩展。
- (4) 将数据处理扩展返回的数据与布局指令进行组合。如果必要，还需要使用报表处理扩展，生成中间格式的报表。
- (5) 将中间格式报表传递给适当的呈现扩展，生成最终的报表。
- (6) 将最终生成的报表返回给请求者。

最终用户是不能查看中间格式报表的，但是中间格式报表可以用任何一种 Reporting Services 支持的格式进行呈现。为了减少生成一个最终报表所花费的时间和资源，可以将中间格式报表进行存储(缓存)以备复用。这样我们可以略过上面的第 2、3、4 步，因此可以更快地返回报表，而且消耗的资源也比较少。Reporting Services 支持三种格式的缓存：

- 报表会话缓存
- 报表执行缓存
- 快照

1. 报表会话缓存

当一位最终用户与 Reporting Services 连接时，马上可以建立一个会话。从最终用户发出的请求都是在一个特定会话的语境中发出的，直到这个会话超时。

在一个会话中，用户常常请求将同一个报表呈现多次，但是每次呈现的格式可能不同。Reporting Services 预见到了这一点，因此将中间格式的报表保存在自己的会话缓存中。缓存的中间格式报表副本记录了会话标识符，这样，当某个最终用户在会话中重复请求一个报表时，就可以利用这个缓存的报表。Reporting Services 提供的这项功能称为报表会话缓存(report session caching)，这项功能始终处于开启状态。

如果修改了报表的参数集，同时在数据集查询中使用了这些参数，那么在报表会话中就要为这些修改过的数据集创建一个附加缓存。

2. 报表执行缓存

为什么必须将缓存的报表与一个会话绑定在一起？为什么不能让所有请求同一个报表的用户共享缓存的报表呢？原因在于安全性。

报表是用获取自外部数据源的数据进行填充的。为了连接这些数据源，必须使用凭据。使用哪个凭据则取决于报表配置或报表使用的共享数据源。

如果获取数据使用的是请求者的凭据，那么报表可能只包含仅对该请求者才有效的数据。中间报表包含了这些数据，因此，如果缓存这些数据并对其他请求者开放这些数据，那么其他用户可能会看到他们本不应该看到的数据。

因此，只有那些不使用请求者凭据从外部数据源获取数据的报表才能被配置为使用报表执行缓

存。利用报表执行缓存，从一个报表请求生成的中间报表只能缓存一段时间，并且只能在缓存的内容过期前才能被其他用户用来呈现报表。

3. 快照

在同时使用了报表会话和报表执行缓存的情况下，最终用户请求一个报表，报表处理程序检查是否存在该报表的缓存。如果不存在该报表的缓存，报表处理程序必须组装中间格式报表，并将其保存到一个缓存中以便下一个请求使用，然后呈现请求的报表并生成最终报表。后面的请求可以利用缓存中的报表副本，但是第一个请求就没有这个优势，因此会导致不同的最终用户具有不同的用户体验。

为了解决这个问题，可以在最终用户发出请求之前，用快照来填充缓存。快照用的是同样的中间格式记录，可以满足与报表执行缓存一样的安全需求。

2.6.2 数据处理扩展

前面已经提到，报表处理程序从报表定义读取了数据检索指令，但是将与外部数据源建立连接并从外部数据源检索数据的工作交给了数据处理扩展。这些扩展向报表处理程序提供了一个数据读取接口，从而使数据可以从这些扩展流向报表处理程序，然后进入中间格式报表。可以在报表服务器上使用多数据源扩展，甚至可以在一个单独的报表中使用多数据源扩展。Reporting Services 包含了多个数据扩展，为以下数据源提供了支持：

- Microsoft SQL Server
- Microsoft SQL Server Analysis Services
- OLE DB 数据源
- ODBC 数据源
- Oracle
- XML 数据源
- SAP NetWeaver BI
- Hyperion Essbase
- Teradata

必须指出，SAP NetWeaver BI、Hyperion Essbase 和 Teradata 扩展需要单独安装客户端组件或.NET 数据提供程序。如果需要使用这些数据处理扩展，请参考联机丛书中的详细内容。如果需要访问其他数据源，那么可以实现一个自定义的数据处理扩展，并将其注册到 Reporting Services 中。第 22 章将研究这个主题。

另外，你也许需要使用一个标准的.NET 或 OLE DB 数据提供程序来获取所需的数据。刚才已经提到，数据处理扩展提供了一个标准数据读取接口。这个接口是基于.NET 规范的，所以这些接口与那些由某些 OLE DB 提供程序提供的接口相差不大。因此，许多.NET 和 OLE DB 数据提供程序都可以注册到 Reporting Services，并且可以把 Reporting Services 当作一个正式的数据处理扩展来使用。针对如何在 Reporting Services 中注册数据提供程序以使用户使用，联机丛书为此提供了详细资料。

2.6.3 报表项

报表处理程序可以生成报表、矩阵和图表，以及各种其他报表项。这些标准报表项可以满足大多数报表作者的需要。但是在某些时候，报表作者也需要使用其他报表项。在这些情况下，可以在 Reporting Services 中注册其他报表项。一般情况下，这些报表项是购自第三方提供商的，例如条码控件和图表控件。当然，也可以自行开发报表项。

无论是自行开发报表项还是外购报表项，报表项都包含了设计组件和运行时组件，并且需要分别注册到报表设计器和 Reporting Services 中。组件都必须提供标准接口，方便报表设计器和报表处理程序与其交互。

2.6.4 呈现扩展

报表处理程序生成(或者从缓存中获取)中间格式报表之后，中间格式报表将被交付给一个呈现扩展，进而被翻译为最终用户需要的格式。Reporting Services 提供了 7 种标准的呈现扩展。表 2-6 描述了这些呈现扩展。每种呈现扩展支持一种或多种报表格式。虽然 Reporting Services 支持自行开发的呈现扩展，但是微软公司并不鼓励自行开发呈现扩展。自行开发的呈现扩展一般需要很高的开发费用。

表 2-6

呈现扩展	支持的格式
HTML	HTML 4.0 (默认) MHTML
CSV	Excel-optimized CSV (默认) CSV-compliant CSV
XML	XML
Image	TIFF (默认) BMP EMF GIF JPEG PNG WMF
PDF	PDF 1.3
Excel	Excel (XLSX)
Word	Word (DOCX)

影响各种呈现扩展生成最终报表的参数被称为设备信息设置(device information settings)。每一种呈现扩展的默认设置是在 rsreportserver.config 文件中设置的。在一个具体请求中，可以根据需要，用精确的格式传送报表，为此可以重写这些设置。

必须指出，报表处理程序并不是仅仅简单地将中间格式报表传送给一个呈现扩展。处理程序和呈现扩展一起工作，呈现扩展随后会通过报表处理程序提供的呈现对象模型(Rendering Object Model, ROM)访问中间报表。

自从 Reporting Services 2008 发布以来，ROM 的基本结构就没有变化过。这样做有很多优点，

其中最重要的优点是可以在联机报表和打印报表之间保证一致性，减少了呈现过程中的内存消耗。

1. HTML 呈现扩展

HTML 易于访问，对于交互式报表来说是一种很好的格式。因此，对 Reporting Services 报表来说，HTML 4.0 是默认的呈现格式。

HTML 格式的不利之处在于 Web 页面始终不便于打印。HTML Viewer 是一种基于 JavaScript 的应用程序，已经嵌入在大多数用 HTML 呈现的报表中，本章一开始就已经讨论过这些报表。HTML Viewer 提供了客户端打印支持，解决了一些从 Web 浏览器打印报表时存在的问题。可以从 HTML Viewer 工具栏进行客户端打印。

可以令 HTML 呈现扩展返回 MIME HTML(MHTML)格式报表，这样可以不再使用默认的 HTML 4.0 格式。如果使用了 MHTML 格式，那么图像、样式表以及其他引用项都将嵌入在 HTML 文档中。这时，传送的报表无需再依赖外部资源。在某些场合下(如使用电子邮件向一个用户发送报表时)这个特性颇为有用。



不是所有的电子邮件产品都支持 MHTML，因此在选择 MHTML 发送电子邮件之前，请检查电子邮件用户群体是否支持 MHTML 格式。

2. CSV 呈现扩展

CSV 是指逗号分隔值(comma-separated values, CSV)。CSV 呈现扩展可以将一个报表的数据部分呈现为一种用逗号分隔的文本文件格式，便于电子表格和其他应用程序访问。

CSV 呈现扩展可以在两种模式下工作。其中，Excel 优化模式是默认模式。此时，报表中的每个数据域都被呈现为一个单独的逗号分隔值块。在 CSV 兼容模式下，CSV 扩展将生成一个单独的、统一的数据块，很多应用程序都可以访问这个数据块。

3. XML 呈现扩展

XML 是另一种常用的报表呈现格式。XML 呈现扩展(XML-rendering extension)可以将数据和布局信息集成到生成的 XML 文件中。

XML 呈现扩展提供的最为强大的功能之一是能够处理 XSLT 文档。如果需要将 XML 文档转换为其他基于文本格式的文档，如 HTML、CSV、XML 以及自定义文件格式，那么 XSLT 文档可以提供指令。Reporting Services 开发团队建议：在开发一个自定义的呈现扩展以满足某种特殊的呈现需求之前，可以首先尝试使用 XSLT 的 XML 呈现格式。

4. Image 呈现扩展

利用 Image 呈现扩展(image-rendering extension)，可以将报表发布为图像格式。Image 呈现扩展支持 7 种图像格式，默认格式为标签图像文件格式(Tagged Image File Format, TIFF)。许多传真程序都使用了 TIFF 格式作为其传输标准，许多组织机构在对文档归档时也使用了这种图像格式。

5. PDF 呈现扩展

Reporting Services 为 Adobe 便携文档格式(Portable Document Format, PDF)提供了一个呈现扩展。PDF 是因特网上最为流行的文档格式。它可以生成简洁、易读的文档，为打印提供了良好的支持。此外，PDF 文档不易修改。

尽管与 HTML 报表相比，PDF 报表交互性不佳(因为 HTML Viewer 可以处理 HTML 报表)，但是 PDF 报表支持文档地图。这项功能支持创建内容表，对大型报表来说，这项功能极具价值。为了查看 Reporting Services 生成的 PDF 文档，需要使用 Adobe Acrobat Reader 8.0 或更高的版本。这个软件可以从 Adobe 网站上免费下载。

当使用 PDF 显示报表时，PDF 呈现扩展尽可能地将报表所需的每种字体的子集嵌入在报表中。报表中使用的每一种字体都必须事先安装在报表服务器上。当报表服务器生成一个 PDF 格式的报表时，它需要使用保存在报表所引用字体中的信息，并在 PDF 文件中创建字符映射。如果引用的字体还没有安装在报表服务器上，那么生成的 PDF 文件可能就不包含正确的映射，因此在查看报表时，报表可能无法正常地显示。

6. Excel 呈现扩展

将报表呈现为 Excel，是 Reporting Services 支持的另一种呈现方式。如果最终用户需要针对数据执行进一步的分析，那么将报表呈现为 Excel 是非常有用的。

默认情况下，SQL Server 2012 中的 Reporting Services 生成的是 Excel OOXML(XLSX)格式。原先的 Excel 呈现器生成的是基于 BIFF8 格式的报表，这种格式是 Excel 97 或更高版本的 Excel 所使用的，现在仍然可以使用。但是如果需要使用这种格式，则必须在 rsReportServer.config 文件中进行配置。

将报表转换为 Excel 格式时，并不是所有的元素都能够很好地被转换为 Excel 格式。矩形、子报表、报表体和数据区域都将被呈现为一组 Excel 单元格。文本框、图像和图表必须呈现在 Excel 单元格中，因此有可能被合并在一起，是否合并则取决于报表其他部分的布局。

因此，如果必须用 Excel 呈现报表，那么在向最终用户发布 Excel 格式报表之前，最好对报表进行一次检查。在将报表呈现为 Excel 格式的过程中，针对每一种报表功能是如何处理的，Reporting Services 联机丛书提供了详细解答。

7. Word 呈现扩展

这个扩展可以用 Microsoft Word 格式来呈现报表，与使用 PDF 格式呈现报表相比，二者具有很多相同的优点，也存在很多相同的限制。与 PDF 不同的是，这个扩展呈现的 Word 格式报表比较便于最终用户进行编辑。

默认情况下，SQL Server 2012 中的 Reporting Services 生成的是 Word OOXML(DOCX)格式。原先的 Word 呈现器生成的是 Word 97 格式的报表，这种格式仍然可以使用，但是需要在 rsReportServer.config 文件中进行配置。

2.6.5 计划和传递处理程序

计划和传递处理程序主要的功能是将订阅报表的请求发送给报表处理程序，并且接收返回的报

表，并与传递扩展一起完成订阅传递的任务。这个处理程序还可以生成快照。

处理程序工作时，需要定期检查 Reporting Services 应用程序数据库中各个表中的内容。这些表是通过请求事件进行填充的。为了填充这些表，既可以通过编程执行 Reporting Services Web 服务的 FireEvent 方法，也可以通过 Reporting Services 配置的计划。计划本身是由 Reporting Services 创建的作业，但是计划是由 SQL Server SQL Agent Windows 服务执行的。创建一个计划时，Reporting Services 处理与作业创建和设置有关的任务。但是，使用计划会产生对 SQL Server SQL Agent Windows 服务的依赖。

2.6.6 传递扩展

计划和传递处理程序调用了传递扩展向订阅者发送报表。Reporting Services 为电子邮件和文件共享传递提供了传递扩展。如果在 SharePoint 集成模式下运行 Reporting Services，那么 Reporting Services 还支持 SharePoint 传递扩展，这个扩展可以用来将内容传递给一个 SharePoint 网站。

与本章所讨论的其他扩展一样，我们也可以对自定义传递进行组装和注册，以便 Reporting Services 使用。联机丛书提供了一个自定义传递扩展的代码，说明如何直接向一台打印机发送报表。

2.7 Reporting Services 应用程序数据库

本章常常提到 Reporting Services 应用程序数据库 ReportServer 和 ReportServerTempDB。这两个数据库保存了报表定义、快照、缓存和安全信息，还保存了其他很多内容。尽管我们强烈建议不要直接访问这些数据库，但是理解其在 Reporting Services 架构中的基本结构和角色还是非常重要的。



当 Reporting Services 运行在 SharePoint 集成模式下时,Reporting Services 在 SharePoint 内容和配置数据库中保存了内容和设置信息。这些数据库属于 SharePoint 应用程序的内容，所以在此不再讨论。与 Reporting Services 类似，建议不要直接访问这些数据库。

2.7.1 ReportServer

ReportServer 数据库保存了 Reporting Services 中的主要数据。这个数据库保存了全部报表定义、报表模型、数据源、计划、安全信息以及快照。因此，定期备份这个数据库是一项关键任务。第 19 章将讨论应用程序数据库的备份和恢复。

表 2-7 列出了 ReportServer 数据库中的一部分表，并描述了这些表的功能。

表 2-7		
功 能 范 围	表 名	表中保存的内容
资源	Catalog	报表定义、文件夹保存位置以及数据源信息
	DataSource	单个数据源信息
安全	Users	已授权用户的用户名和安全 ID(SID)信息
	Policies	一个引用列表。每个引用是针对一个不同的安全策略的引用

(续表)

功 能 范 围	表 名	表中保存的内容
安全	PolicyUserRole	一个用户与组、角色和策略的关联
	Roles	一个列表，列表中每个元素定义了角色和角色可以执行的任务
快照	SnapshotData	用于运行一个快照的信息，包括查询参数和快照依赖
	ChunkData	报表快照
	History	存储的快照和保存快照日期之间的引用
计划	Schedule	不同的报表执行和订阅传递计划的信息
	ReportSchedule	一个给定报表、该报表执行计划以及执行操作之间的关联
	Subscriptions	一个订阅列表，包括拥有者、参数以及传递扩展
	Notifications	订阅通知信息，如处理日期、最近一次运行时间以及传递扩展
	Event	事件通知的临时存储位置
	ActiveSubscriptions	订阅成功/失败信息
	RunningJobs	当前执行的计划处理程序
管理	ConfigurationInfo	Reporting Service 配置信息，该信息应该通过事先规定的接口进行管理，不要直接编辑该表中保存的数据
	Keys	用于数据加密的公钥和私钥列表
	ExecutionLogStorage	一个已执行的报表列表，以及与报表执行事件有关的关键元数据

2.7.2 ReportServerTempDB

ReportServerTempDB 数据库保存了临时性的 Reporting Services 信息，包括会话和缓存数据。

如果没有 ReportServerTempDB 数据库，那么 Reporting Services 是无法正常工作的。但是，我们并不需要备份 ReportServerTempDB 数据库，因为这个数据库中保存的所有数据都是临时性的。如果这个数据库中的内容丢失，那么也只需重新构建这个数据库。第 19 章将讨论如何重新构建这个数据库。

表 2-8 列出了 ReportServerTempDB 数据库中的一部分表，并给出了表的描述信息。

表 2-8

表 名	说 明
ChunkData	为会话缓存的报表和缓存的实例保存了报表定义和数据
ExecutionCache	保存了执行信息，包括缓存实例的超时信息
PersistedStream	为单独的一个用户保存了会话级呈现输出
SessionData	持久化一个用户的会话级信息，包括一个给定会话信息的报表路径和超时
SessionLock	临时存储，用来处理会话数据的锁定
SnapshotData	保存一个临时快照

2.8 小结

本章介绍了如何在 Native 模式下完成 Reporting Services 的基本安装，还介绍了如何运行 Reporting Services。完成本章内容后，你就可以通过学习本书内容来探索这个产品。尽管基本安装已经很有用，

但是基本安装忽略了很多企业部署需要考虑的问题。为了帮助你完成企业部署，后面介绍了与这些问题有关的知识。

本章还探讨了 Reporting Services Native 模式的架构。除此之外，还有以下内容：

- 将报表生命周期划分为三个阶段。报表是由终端用户和报表专家编写的，并且作为一个集中式的报表系统的组成部分进行管理，最终通过多种手段传递到最终用户的手中。
- 为了支持报表生命周期，Reporting Services 提供了多种应用程序，包括报表生成器、报表设计器、报表管理器、Reporting Services 配置工具、HTML Viewer、Reporting Services Web 服务以及订阅。此外，还有其他工具也为报表生命周期提供了支持。
- Windows 服务的结构，以及 Windows 服务用来完成实际报表功能的组件，包括处理程序、扩展以及数据库。

本章你所学到的知识将为你在后续章节中深入研究各项报表内容打下坚实基础。