



Professional SharePoint 2013 Development

SharePoint 2013 开发高级教程 (第4版)

[美]

Reza Alirezaei
Matt Ranlett
Brian Wilson
Paul J. Swider
蒲 成 李文强

Brendon Schwartz
Scot Hillier
Jeff Fried

著

译

清华大学出版社

SharePoint 2013

开发高级教程

(第 4 版)

Reza Alirezai
Brendon Schwartz
Matt Ranlett
[美] Scot Hillier 著
Brian Wilson
Jeff Fried
Paul J. Swider
蒲 成 李文强 译

清华大学出版社

北 京

Reza Alirezaei, Brendon Schwartz, Matt Ranlett, Scot Hillier, Brian Wilson, Jeff Fried, Paul J. Swider
Professional SharePoint 2013 Development
EISBN : 978-1-118-49582-7
Copyright © 2013 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: 01-2013-8906

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

SharePoint 2013 开发高级教程(第 4 版)/(美)阿里瑞萨(Alirezaei, R.) 等著; 蒲成, 李文强 译.
—北京: 清华大学出版社, 2014
书名原文: Professional SharePoint 2013 Development
ISBN 978-7-302-38015-3

. S... . 阿... 蒲... 李... . 企业管理—应用软件—教材 . F270.7

中国版本图书馆 CIP 数据核字(2014)第 216010 号

责任编辑: 王 军 韩宏志
装帧设计: 牛艳敏
责任校对: 邱晓玉
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: 46 字 数: 1119 千字

版 次: 2014 年 10 月第 1 版

印 次: 2014 年 10 月第 1 次印刷

印 数: 1~3000

定 价: 98.00 元

产品编号:

译者序

作为目前最强大的企业级应用平台之一,SharePoint 已逐渐成为全世界范围内企业建立高效协作应用的首选基础性平台。简单来说,SharePoint 是一个基于浏览器来提供协作、内容管理以及扩展功能的企业信息化解决方案平台,其最新版本为 SharePoint 2013。自 2001 年诞生至今,SharePoint 已经从最初的企业门户建设发展成为集企业门户、信息分享、社交功能、工作流、商务智能甚至是云端应用等诸多功能为一体的集成化平台。

现在,使用 SharePoint 2013 来配置和开发应用,基本可以满足所有的企业日常业务处理。为了迎合 APP 及云技术的潮流趋势,SharePoint 2013 进行了一次华丽转身。SharePoint 2013 在管理方面也进一步改进,除了支持云端部署外,还提供全新的应用程序构建框架,协助用户将云端上提供的各类 APP 与服务整合到 SharePoint 2013 中。

本书由处于 SharePoint 领域前沿的一些最著名的开发人员和资深咨询顾问所撰写,详尽介绍了 SharePoint 2013 的所有更新内容,深入剖析 SharePoint 开发方方面面的内容。本书的内容完整而全面,从新特性和概念入手,结合微软所提供的所有开发与配置工具并辅以针对性较强的示例讲解,呈现出 SharePoint 2013 各项新功能的配置、开发、部署及应用的全貌。

不过,因为本书主要面向开发人员,且内容中涉及大量开发术语、Web 技术以及开发工具,所以如果读者在阅读本书之前已经具备一些 Web 开发和开发工具的知识,那么在理解本书的内容时将会事半功倍。此外,本书各章节介绍的主要功能,几乎都提供相关的资源,比如,源代码、官方及第三方资料网站、权威人士的博客等,如果读者能够善用这些资源,相信能够快速提升和积累 SharePoint 开发和应用层面的技能及经验。

SharePoint 的特性决定了本书内容的广度和深度,虽然译者有多年的 Web 开发及 SharePoint 开发配置部署经验,但在翻译这样一本内容面如此博大精深的书籍时仍然觉得有些许力不从心,在翻译过程中,译者字斟句酌,力争让读者能够准确地理解原文的精髓。如有翻译不当之处,敬请读者谅解。读者可将意见和建议发送到 wkservice@vip.163.com。

本书全部章节由蒲成、李文强翻译,参与翻译工作的还有申成龙、杨帆、王滨、李鹏、贫书谦、赵栋、林超、陈世佳、王佳。

在此要特别感谢清华大学出版社的编辑和微软(中国)有限公司解决方案部区域经理刘浩,他们在本书翻译过程中为译者提供了巨大的帮助,没有他们的热情付出,本书将难以顺利付梓。

作者简介

Reza Alirezaei

Reza Alirezaei 是 Development Horizon 公司的创办人和总裁，该公司总部位于加拿大多伦多，是微软的金牌合作伙伴。Reza 自 2006 年开始就一直是 SharePoint 领域的微软最有价值专家(MVP)。除了从事咨询工作，Reza 在企业培训和 SharePoint 社区的演讲方面拥有 10 年的经验。他著有一些 SharePoint 书籍、论文和在线文章。Reza 的博客地址是 <http://blogs.devhorizon.com/reza>，还可以向其邮箱 reza@devhorizon.com 发送电子邮件来联系他。

Brendon Schwartz

Brendon Schwartz 是 SharePoint 专家，拥有 10 年 SharePoint 经验，自该产品早期就开始从事此方面工作了。作为协作、社交计算、企业内容管理、业务过程管理和应用程序开发领域的专家，Brendon 出席过各种研讨会并构建过具有创新性的产品解决方案。他为许多客户做过项目管理，包括《财富 500 强》企业，比如，美国电话电报公司、可口可乐公司、AutoTrader.com 和家得宝(The Home Depot)公司。在带领团队为可口可乐公司构建一个首批由微软托管的解决方案期间，他协助制定了 Office 365 的未来发展。Brendon 荣获 ASP.NET 和 SharePoint 领域的微软 MVP。作为 SharePoint 的热心作者，他创作了数量众多的文章和书籍，其中包括 *Professional SharePoint 2007 Development* 和 *Social Computing with Microsoft SharePoint 2007*。Brendon 非常享受与妻子、儿子和幼女在一起的时光。

Matt Ranlett

Matt Ranlett 是一名 SharePoint Server MVP，他在 Slalom 咨询公司工作并担任解决方案架构师和咨询实习经理。当他没有参与组织或出席用户群相关的活动时，Matt 会撰写和编辑 SharePoint 方面的白皮书、杂志文章和书籍。

Scot Hillier

Scot Hillier 是一名独立咨询顾问，并且是微软 SharePoint MVP，专注于为使用 SharePoint、Office 及相关技术的信息工作者创建解决方案。他是 18 本微软技术书籍的作者/合著者，其中包括 *Inside SharePoint 2013* 和 *App Development in SharePoint 2013*。Scot 的一部分时间用来针对 SharePoint 项目提供咨询服务，另一部分时间用来出席像微软 SharePoint 大会这样的 SharePoint 活动并演讲，还通过关键路径培训为 SharePoint 开发人员提供培训。Scot 早先曾是一名美国海军潜艇军官，毕业于弗吉尼亚军事学院。可以向其邮箱 scot@scothillier.net 发送电子邮件来联系他。

Brian Wilson

Brian Wilson 是一名 SharePoint 解决方案和信息架构师, 并且是 WiredLight 公司的董事, 这是一家提供 SharePoint 解决方案的公司。Brian 拥有 15 年经验(其中包括四年在微软咨询服务部的 SharePoint 和信息工作者团队中担任高级顾问), 他承担过微软一些大型客户的架构设计和解决方案开发工作, 这些客户的企业环境使用了 SharePoint 技术。从 SharePoint 的第一个版本开始, 他就在美国、欧洲、亚洲和南非的许多行业中参与了多种前沿的 SharePoint 客户项目。Brian 在 WiredLight 公司为客户提供创新型的设计和开发的专业知识, 该公司专注于提供 SharePoint 咨询、产品和解决方案。当他有闲暇时间的时候, 喜欢滑雪、潜水以及观看橄榄球的精彩球赛。想要了解关于 Brian 的更多信息, 请访问 <http://www.wiredlight.net/> 或他的 LinkedIn 个人资料 <http://uk.linkedin.com/in/bkvwilson>。

Jeff Fried

Jeff Fried 是 BA Insight 公司的 CTO, 专注于用 SharePoint 进行基于搜索的应用程序的开发。Jeff 经常就该领域进行演讲并撰文, 拥有 15 项专利, 已经撰写了超过 50 篇技术论文, 并在下一代搜索引擎、网络和联络中心领域引领了开创性产品的创建工作。

Paul J. Swider

Paul Swider 是一名国际演讲家、培训师和自由顾问。此外, 他还是 Charleston SharePoint Users Group 的发起人以及来自查尔斯顿港附近一个堤礁岛的成功企业家, 在故乡, Paul 喜欢驾着小船追逐潮汐和海风。Paul 参与了许多社会团体和慈善事业, 他是 Sharing the Point 的创始会员, 这是一个在新兴市场提供免费 SharePoint 培训机会的国际合作行动组织。17 年的软件咨询经验以及许多的微软认证和演讲资格证书, 使得 Paul 成为 SharePoint 社区的一名权威人士。作为一名微软认证培训教师(MCT), Paul 为成千上万名 SharePoint 管理员、开发人员和架构师提供过培训和咨询。

技术编辑简介

Andy Au

Andy Au 是 Development Horizon 咨询公司的团队领导者，该公司总部位于加拿大多伦多，是 SharePoint 领域的微软金牌认证合作伙伴。Andy 拥有超过 8 年的工作经验，其中在 SharePoint 2003、2007 和 2010 方面有 6 年的工作经验。在其职业生涯中，Andy 多次受雇于各种解决方案的管理层和开发团队，且他拥有微软认证技术专家(MCTS)、微软认证信息技术专家(MCITP)以及微软认证开发专家(MCPD)的认证。闲暇时间，Andy 喜欢观看体育比赛并与家庭和朋友一起休闲放松。

Mehrdad Mehrjoo

Mehrdad Mehrjoo 在其过去 7 年的职业生涯中一直专注于 SharePoint 方面的工作。Mehrdad 的知名度非常高，因为他能够将几乎所有的软件与 SharePoint 集成在一起工作。他现在受雇于 Development Horizon 公司，是一名高级 SharePoint 咨询顾问，负责领导底层基础架构设计和开发实践。他是一名公认 MCTS、MCITP 以及 MCPD。Mehrdad 十分享受与妻儿在一起的时光。

Siavash Fathi

Siavash Fathi 是 Development Horizon 公司的一名高级 SharePoint 咨询顾问。Siavash 对于 SharePoint 应用程序充满了热情并专注于客户端和远程编程。除了 SharePoint 之外，Siavash 还喜欢研究机器人技术和人工智能(AI)，并发表了几篇关于这些领域的论文。Siavash 拥有 MCTS、MCITP、MCPD 以及敏捷大师认证(CSM)的认证。

致 谢

撰写一本书从来都不是一项轻松的任务。在本书的编写过程中有许多人投入了大量的精力，虽然本书封面没有列出他们的姓名，但也需要感谢他们。我要谢谢我的太太 Heidi，是她使得这一切成为可能。你的牺牲精神和愿意独自照顾家庭的奉献精神，一直以来都让我觉得你非常了不起。尽管我们的孩子还太小，还认识不到爸爸在周六和周日外出工作是不合理的，我还是非常感激可爱的孩子们。如果一个人没有良好的后援，那他就是不完整的，所以我要感谢与我一起共事过的每一个人，是他们给予了我变得更好的动力。首先，要感谢 Wrox 公司我们团队的所有成员，没有他们，本书不可能完成——Mary、Kelly 以及所有的编辑，你们非常棒！我要谢谢我的朋友 Aaron Richards，特别是他的创新性思维和对解决方案以及 InfoPath 永不停歇地追寻。还要感谢向我伸出援手的技术支持团队，如 Andrew Connell、Douglas McDowell、Robert Bogue 和 Doug Ware。当我面临很棘手的问题和需要一篇阐述某些技术的博文时，Doug Ware 从没让我失望过。谢谢所有的 SharePoint MVP，你们是最伟大的朋友，也是最好的 MVP 社区。还要谢谢这些年来一直鼓励我的好朋友 Matt Ranlett、Jeremy Howard、Jerry Pattenau 与 Chris Haldeman。十分感谢 SharePoint 团队，他们总是乐于为我提供答案。最后，我想要感谢曾经帮助我成长、学习和让我保持良好发展方向的人，其中包括我的父母、Doug Turnure、Aaron Cutlip、Dan Attis 以及其他无法一一列举的人。

——Brendon Schwartz

首先我需要感谢我的家庭，你们给了我最大的包容，让我可以花费无法尽数的时间来研究、写作和编辑本书的内容。还要感谢 Wrox 天才般的编辑团队和审稿人，是你们不辞辛劳的工作才让本书顺利面世。

——Matt Ranlett

非常感谢 Bill Baer 和 Vesa Juvonen，你们为本书的各章内容做出了巨大贡献。你们的建议、专业知识和贡献帮助编写团队创作出了一本高质量的 SharePoint 2013 书籍。谢谢你们！向 SharePoint 和 Office 产品团队，向 SharePoint 社区中的每一个人：向我们站在其肩膀上的那些巨人致敬！

——Brian Wilson

感谢我的太太和家庭给予我的支持，感谢我的 SharePoint 朋友们给予我的启迪，感谢本书的编辑为这本书付出的努力以及持续推进。

——Jeff Fried

前言

如果你已经拥有一些使用 SharePoint 平台及其辅助技术的经验,大概已经了解 SharePoint 是一个通用平台,用于构建能够满足广泛业务需求的解决方案。总体而言,由于 SharePoint 变得越来越重要且用途越来越广泛,因此多年来微软对于该平台投入了大量资源。现今,这一伟大产品的最新版本是 SharePoint 2013!

SharePoint 2013 中为核心平台引入了许多新功能和改进功能。不过,最激动人心的新增内容是新的应用程序模型,它允许开发人员构建应用程序并对其进行隔离式部署,而只需要少量或不需要平台上其他软件的依赖选项,应用程序安装在该平台上且无须自定义代码就能在 SharePoint 服务器上运行。

在 SharePoint 2013 中,更加强调的是云端编程和标准化 Web 技术,如 JavaScript 和 HTML。从这方面看,可以说 SharePoint 2013 平台的其他主要变化就是为了支持新的应用程序开发模型。

本书读者对象

本书适合那些有兴趣在 SharePoint 2013 上或在 Office 365 中的 SharePoint Online 上进行应用程序开发的读者。尽管一些知识要求具备 SharePoint 经验,但书中的示例都非常详尽,而且如果你已经具备 Web 开发和开发工具的知识,那么你将很容易就能理解。

本书内容

SharePoint 2013,正如其之前的版本一样,是一个大型产品,所以本书的内容也很完整和全面。因此,在你开始详细阅读本书之前,此处先让你了解一些关于每章的必要内容以及该章的学习目标。

本书开篇概述 SharePoint 的架构,随后第 1 章和第 2 章介绍与开发人员相关的新功能。第 3 章和第 4 章将讲解 SharePoint 2013 中的工具化体验,以及一些关于 SharePoint 2013 中的应用程序生命周期管理(ALM)的重要信息。

鉴于应用程序模型在 SharePoint 2013 中的重要性,第 4 章~第 7 章将着重介绍应用程序开发和云。

第 8 章~第 15 章会讲解 SharePoint 2013 的一些重要领域,其中许多都得到了重要更新。这部分内容还包括第 11 章。由于在 SharePoint 套件中,InfoPath 的未来具有一些不确

定性，且一些客户已经进行了投入，该章自然会阐述一些内容，以帮助开发人员更好地决定如何向前推进。

第 16 章 ~ 第 19 章关注的重点是商务智能(BI)与 SharePoint 的集成。相较于 SharePoint 2010 ,SharePoint 2013 中的 BI 提供了更好的分析、制作报表、制作仪表板和可视化的能力。这通过更好的 BI 工具以及在 SharePoint、微软 Office 应用程序和 SQL Server 2012 之间更紧密的集成来实现。

本书的内容安排

本书的内容安排自然而然地建立在你逐步深入阅读本书内容所能学到的技能的基础上。在初步介绍和基本平台介绍的章节内容之后，本书内容将转向 SharePoint 平台的高级功能部分。每一章都建立在本书之前章节内容所介绍的知识的基础上，所以你应该按照先后次序通读这些章节，或者至少在阅读本书后续章节之前阅读“前言”和简要介绍 SharePoint 平台的章节。

学习本书的前提条件

要最大限度地理解本书，以及本书面向开发人员的原因，你需要一个开发环境。有两种方式来准备开发环境。

- ？ 构建你自己的开发计算机。构建一台完善开发计算机的最佳资源，是 MSDN 的文档，请参考 <http://msdn.microsoft.com/en-us/library/ee554869.aspx>。
- ？ 使用 Office 365 开发人员网站。这会缩短你的设置时间并能够让你在 5 分钟内就入门。这种设置你需要做的，仅仅是在 <http://msdn.microsoft.com/en-us/library/fp179924.aspx> 注册一个开发人员网站。

如果你选择使用 Office 365 开发人员网站，请注意你还可以为你的开发人员网站部署 Napa，它是一个 Office 365 特有的开发工具。它允许你构建 SharePoint 托管的应用程序，而无需在你的开发计算机上安装 Visual Studio 2012 和 Office Developer Tools for Visual Studio 2012。

最后，最重要的一点是，微软已经为 SharePoint 2013 发布了许多示例代码，这些代码几乎覆盖了 SharePoint 开发的方方面面。可在 <http://msdn.microsoft.com/en-us/library/jj901637.aspx> 网页找到这些示例代码。

本书约定

为了帮助你最大限度地理解文字内容并提醒你注意正在介绍的内容，本书使用了大量的约定。



警告：此处内容表示非常重要、不可遗忘的信息，这些信息与其周围的内容密切相关。



提示：提示、贴士、暗示、技巧以及当前内容的延伸介绍会以此形式进行补充。

代码有两种不同的表示方式：

- ？ 我们使用非高亮的 monofont 字体类型来表示大多数代码示例。
- ？ 我们使用粗体来强调在当前上下文中特别重要的代码，或者表示对比之前代码段的变化。

源代码

在读者学习本书中的示例时，可以手工输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/> 或 www.tupwk.com.cn/downpage 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。



提示：由于许多图书的标题都很类似，因此按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 978-1-118-49582-7。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

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，就必须先加入该论坛。

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。如果要想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 1 章 | SharePoint 2013 架构概览 | 1 |
| 1.1 | 从架构角度看有哪些新内容 | 1 |
| 1.2 | 内部部署服务器场架构 | 2 |
| 1.2.1 | Web 服务器层 | 3 |
| 1.2.2 | 应用服务器层 | 3 |
| 1.2.3 | 数据库服务器层 | 4 |
| 1.2.4 | 小型或单层拓扑 | 4 |
| 1.2.5 | 中型或双层拓扑 | 4 |
| 1.2.6 | 大型或三层拓扑 | 5 |
| 1.2.7 | 基于地理位置的分布式拓扑 | 6 |
| 1.3 | 服务应用程序架构 | 6 |
| 1.3.1 | 服务应用程序模型 | 6 |
| 1.3.2 | 可用的服务应用程序 | 8 |
| 1.3.3 | 服务应用程序的生命周期 | 10 |
| 1.4 | 多租户托管架构 | 18 |
| 1.4.1 | 网站订阅 | 19 |
| 1.4.2 | 服务应用程序分区 | 19 |
| 1.4.3 | 租户管理 | 22 |
| 1.4.4 | 功能包 | 23 |
| 1.5 | 搜索架构 | 23 |
| 1.5.1 | 爬网和内容处理组件 | 24 |
| 1.5.2 | 分析处理组件 | 24 |
| 1.5.3 | 索引和查询处理 | 24 |
| 1.5.4 | 搜索管理组件 | 25 |
| 1.5.5 | 多功能(包括搜索功能)中型 搜索场拓扑示例 | 25 |
| 1.6 | SQL Server 数据库架构 | 27 |
| 1.6.1 | 数据库拓扑 | 27 |
| 1.6.2 | SharePoint 2013 数据库 | 30 |
| 1.7 | 云托管架构 | 33 |
| 1.7.1 | 安全功能 | 33 |

| | | |
|-------|---------------------------------|----|
| 1.7.2 | 身份验证功能 | 34 |
| 1.7.3 | 管理模型 | 34 |
| 1.8 | 本章小结 | 35 |
| 第 2 章 | SharePoint 2013 的新功能 | 37 |
| 2.1 | SharePoint 2013 简介 | 37 |
| 2.2 | 部署方案 | 38 |
| 2.2.1 | 内部部署 | 38 |
| 2.2.2 | Office 365 部署 | 38 |
| 2.2.3 | 托管式部署 | 39 |
| 2.2.4 | 混合式部署 | 39 |
| 2.3 | 封装方案 | 39 |
| 2.3.1 | 完全信任的场解决方案 | 39 |
| 2.3.2 | 部分信任的沙盒解决方案 | 39 |
| 2.3.3 | SharePoint 应用程序 | 39 |
| 2.4 | 编程模型 | 40 |
| 2.4.1 | SharePoint 编程模型的改进 | 40 |
| 2.4.2 | SharePoint 2010 中 CSOM 面临的问题 | 42 |
| 2.4.3 | 服务器端代码的问题 | 42 |
| 2.4.4 | 新的编程模型 | 43 |
| 2.5 | 应用程序模型 | 47 |
| 2.5.1 | SharePoint 托管的应用程序 | 47 |
| 2.5.2 | 提供程序托管的应用程序 | 47 |
| 2.5.3 | Azure 自动托管的应用程序 | 47 |
| 2.5.4 | 应用程序或解决方案—— 如何取舍 | 48 |
| 2.5.5 | 应用程序安全模型 | 51 |
| 2.6 | 远程事件 | 54 |
| 2.6.1 | Reporting Services 2012 数据警告 | 55 |
| 2.6.2 | 外部列表事件 | 55 |

| | | | | | |
|---------|--|----|--------|---|-----|
| 2.6.3 | 远程事件接收器 | 56 | 2.11 | BCS | 85 |
| 2.6.4 | 远程事件中的安全模型 | 57 | 2.11.1 | OData 连接器 | 86 |
| 2.7 | 工作流 | 58 | 2.11.2 | BCS 驱动的应用程序 | 86 |
| 2.7.1 | SharePoint 2010 中工作流的 问题 | 58 | 2.11.3 | CSOM 与 REST 接口 | 87 |
| 2.7.2 | SharePoint 2013 中的工作流 架构 | 59 | 2.12 | 本章小结 | 88 |
| 2.7.3 | 构建自有工作流开发环境 | 61 | 第 3 章 | SharePoint 2013 的开发人员 工具 | 89 |
| 2.7.4 | 使用 SharePoint Designer 2013 设计工作流 | 63 | 3.1 | SharePoint 自定义选项 | 89 |
| 2.7.5 | 在应用程序中使用工作流 | 66 | 3.2 | OOB 开发体验 | 90 |
| 2.8 | 企业内容管理 | 66 | 3.3 | 了解 SharePoint Designer 2013 | 95 |
| 2.8.1 | 网站策略 | 66 | 3.3.1 | SharePoint Designer 新功能 | 96 |
| 2.8.2 | 托管元数据 | 67 | 3.3.2 | 用户界面导航 | 98 |
| 2.9 | Web 内容管理 | 69 | 3.4 | 了解 Visual Studio 2012 工具 | 100 |
| 2.9.1 | 结构化发布模型 | 70 | 3.4.1 | 开始创建新的 SharePoint 2013 项目 | 101 |
| 2.9.2 | 动态发布模型 | 70 | 3.4.2 | Visual Studio 集成的列表及 对内容类型的支持 | 102 |
| 2.9.3 | 分类驱动导航 | 70 | 3.4.3 | 服务器资源管理器中的 SharePoint 连接 | 102 |
| 2.9.4 | 术语驱动发布页面 | 71 | 3.4.4 | 解决方案资源管理器集成 | 103 |
| 2.9.5 | 跨网站发布 | 71 | 3.4.5 | Mapped 文件夹 | 103 |
| 2.9.6 | 主机名网站集 | 73 | 3.4.6 | SharePoint 应用程序 | 104 |
| 2.9.7 | 多语言功能 | 74 | 3.4.7 | SharePoint 解决方案 (经典解决方案) | 108 |
| 2.9.8 | 内容搜索 Web 部件 | 75 | 3.5 | 设置开发环境 | 112 |
| 2.9.9 | 设计管理器 | 76 | 3.5.1 | 用于 SharePoint 和 Office 365 开发环境的应用程序 | 112 |
| 2.9.10 | 移动端支持 | 77 | 3.5.2 | 本地开发环境 | 113 |
| 2.9.11 | 图像呈现形式 | 77 | 3.6 | 调试时的故障排除 | 115 |
| 2.9.12 | 应用程序驱动发布网站 | 78 | 3.6.1 | F5 调试 | 115 |
| 2.10 | 搜索 | 78 | 3.6.2 | 使用开发人员面板进行 调试 | 118 |
| 2.10.1 | 搜索架构 | 79 | 3.6.3 | 使用 SharePoint 日志进行 调试 | 119 |
| 2.10.2 | 搜索导航 | 79 | 3.6.4 | 调试 Silverlight 代码 | 120 |
| 2.10.3 | 结果源 | 80 | 3.6.5 | 其他有用的调试和 测试工具 | 120 |
| 2.10.4 | 显示模板 | 80 | 3.7 | 本章小结 | 121 |
| 2.10.5 | 结果类型 | 81 | | | |
| 2.10.6 | 查询规则 | 81 | | | |
| 2.10.7 | 连续爬网 | 82 | | | |
| 2.10.8 | 融会贯通 | 82 | | | |
| 2.10.9 | 查询语言 | 83 | | | |
| 2.10.10 | 导出和导入搜索设置 | 84 | | | |
| 2.10.11 | 搜索驱动的解决方案 | 84 | | | |

| | |
|--|------------|
| 第 4 章 SharePoint 2013 应用程序 | |
| 生命周期管理 | 123 |
| 4.1 ALM 入门 | 124 |
| 4.1.1 ALM 的三个角度 | 124 |
| 4.1.2 ALM 工具 | 125 |
| 4.1.3 了解关键的开发工具 | 126 |
| 4.2 了解 SharePoint 2013 解决方案模型 | 127 |
| 4.3 规划自定义模型及版本发布方式 | 129 |
| 4.3.1 自定义模型 | 129 |
| 4.3.2 发布封装包的方式 | 133 |
| 4.4 规划关键开发阶段与发布模型 | 136 |
| 4.4.1 关键开发阶段 | 136 |
| 4.4.2 发布模型 | 138 |
| 4.5 规划升级与补丁更新方式 | 143 |
| 4.5.1 升级解决方案和功能 | 143 |
| 4.5.2 为 SharePoint 2013 环境更新补丁 | 143 |
| 4.6 规划 SharePoint 团队环境 | 145 |
| 4.6.1 大型项目的环境 | 145 |
| 4.6.2 确定开发人员需要的环境 | 146 |
| 4.6.3 确定测试人员需要的环境 | 150 |
| 4.7 管理 SharePoint 2013 开发团队 | 150 |
| 4.7.1 选择软件开发方法 | 151 |
| 4.7.2 建立开发标准和最佳实践 | 151 |
| 4.7.3 管理、跟踪和记录关键设计决策 | 152 |
| 4.7.4 常规代码审查规划 | 152 |
| 4.7.5 大型项目的考虑事项 | 153 |
| 4.7.6 大型项目的生命周期模型 | 153 |
| 4.7.7 分散式开发团队 | 155 |
| 4.7.8 海外团队 | 155 |
| 4.8 管理 SharePoint 2013 测试团队 | 156 |
| 4.8.1 设立测试策略 | 156 |
| 4.8.2 单元测试 | 157 |
| 4.8.3 自动化生成与集成测试 | 157 |
| 4.8.4 测试用例规划 | 159 |
| 4.8.5 性能测试 | 160 |
| 4.8.6 功能测试 | 161 |
| 4.8.7 用户验收测试 | 162 |
| 4.8.8 缺陷跟踪 | 162 |
| 4.8.9 其他测试考虑事项 | 163 |
| 4.9 本章小结 | 163 |
| 第 5 章 Windows Azure 与 SharePoint 2013 的集成 | 165 |
| 5.1 步入云端 | 165 |
| 5.1.1 云 | 166 |
| 5.1.2 公司为什么要关注 | 167 |
| 5.1.3 开发人员为什么要关注 | 168 |
| 5.1.4 云技术使用警示 | 169 |
| 5.2 Windows Azure | 172 |
| 5.2.1 执行模型 | 172 |
| 5.2.2 移动服务 | 173 |
| 5.2.3 高性能计算 | 173 |
| 5.2.4 市场 | 173 |
| 5.2.5 数据管理 | 173 |
| 5.2.6 业务分析 | 174 |
| 5.2.7 媒体 | 174 |
| 5.2.8 网络 | 174 |
| 5.2.9 缓存 | 175 |
| 5.2.10 消息 | 175 |
| 5.2.11 身份 | 175 |
| 5.3 SharePoint 和微软的云产品 | 176 |
| 5.3.1 SharePoint 的 SaaS 授权 | 176 |
| 5.3.2 SharePoint 的 PaaS 授权 | 177 |
| 5.3.3 集成模式 | 177 |
| 5.4 本章小结 | 181 |
| 第 6 章 SharePoint 2013 应用程序开发入门 | 183 |
| 6.1 SharePoint 应用程序架构概览 | 183 |
| 6.1.1 编程模型概述 | 184 |
| 6.1.2 应用程序与解决方案对比 | 185 |

| | | | |
|--|-----|--|-----|
| 6.1.3 托管选项..... | 186 | 7.6.3 开发提供商托管的应用程序 与 Azure 托管的应用程序的 决策标准 | 231 |
| 6.2 应用程序探讨 | 187 | 7.7 本章小结 | 231 |
| 6.3 开发你的第一款应用程序 | 190 | 第 8 章 SharePoint 2013 社交应用 程序开发..... | 233 |
| 6.4 CSOM..... | 194 | 8.1 SharePoint 2013 中新的和 改进后的社交功能..... | 234 |
| 6.4.1 超越 SharePoint 2010 的 CSOM 改进 | 196 | 8.1.1 新的用户体验 | 234 |
| 6.4.2 RESTful SharePoint 2013 | 197 | 8.1.2 多浏览器支持 | 234 |
| 6.5 开发外部托管应用程序 | 200 | 8.1.3 富文本编辑器 | 235 |
| 6.5.1 单服务器开发环境的 高度信任应用程序 | 200 | 8.1.4 企业社交网络 | 235 |
| 6.5.2 开发提供商托管的 应用程序 | 203 | 8.1.5 内容共享 | 236 |
| 6.5.3 开发 Azure 自动托管的 应用程序 | 205 | 8.1.6 吸引用户为博客、Wiki 和 讨论贡献内容信息 | 237 |
| 6.6 本章小结 | 210 | 8.1.7 社交分类和反馈 | 238 |
| 第 7 章 SharePoint 2013 应用程序 深度开发..... | 211 | 8.1.8 企业分类 | 239 |
| 7.1 应用程序用户体验开发 | 211 | 8.2 传统社交用户生成内容的 体验 | 240 |
| 7.1.1 开发嵌入式应用程序或 应用程序部件 | 215 | 8.2.1 无处不在的 Wiki | 240 |
| 7.1.2 开发自定义操作应用程序..... | 219 | 8.2.2 博客 | 241 |
| 7.1.3 应用程序设计指南 | 222 | 8.3 社区 | 242 |
| 7.2 开发应用程序包 | 222 | 8.3.1 社区网站模板 | 243 |
| 7.3 开发应用程序部署 | 224 | 8.3.2 讨论 | 244 |
| 7.3.1 发布到 Office Store | 224 | 8.4 声望 | 246 |
| 7.3.2 发布到应用程序目录 | 225 | 8.4.1 授予徽章 | 247 |
| 7.4 应用程序升级功能开发 | 225 | 8.4.2 将最佳回复和问与答一起 使用 | 247 |
| 7.4.1 SharePoint 应用程序更新 | 226 | 8.4.3 成员页面 | 247 |
| 7.4.2 应用程序迁移 | 226 | 8.5 深入介绍 My Site | 248 |
| 7.5 应用程序开发人员需要 额外考虑的事项 | 227 | 8.5.1 用户个人资料 | 248 |
| 7.6 重要建议 | 227 | 8.5.2 My Site 文档库和 SkyDrive Pro | 249 |
| 7.6.1 云托管的应用程序与 SharePoint 托管的应用 程序的决策标准 | 228 | 8.5.3 关注内容和人员 | 250 |
| 7.6.2 开发应用程序与开发场 解决方案的决策标准 | 228 | 8.6 深入介绍新闻源 | 250 |
| | | 8.6.1 微博 | 251 |
| | | 8.6.2 活动源 | 252 |
| | | 8.7 使用社交 API 进行编程 | 253 |

| | | | |
|--------------------------|-----|--------------------------|-----|
| 8.8 本章小结 | 254 | 9.6 定制相关性 | 305 |
| 第 9 章 构建基于搜索的 SharePoint | | 9.6.1 通过查询和内容管理 | |
| 2013 应用程序 | 255 | 相关性 | 305 |
| 9.1 搜索架构与可扩展性 | 256 | 9.6.2 排名配置文件 | 307 |
| 9.1.1 新的搜索架构 | 256 | 9.7 整合使用 | 308 |
| 9.1.2 新搜索架构的本质 | 257 | 9.7.1 构建快速垂直搜索 | 308 |
| 9.1.3 搜索的扩展点 | 258 | 9.7.2 构建基于搜索的自定义 | |
| 9.1.4 顶层定制方案 | 258 | 应用程序 | 308 |
| 9.2 由搜索驱动的应用程序 | 259 | 9.7.3 搜索与其他工作负荷结合 .. | 310 |
| 9.2.1 开箱即用的搜索应用程序 .. | 260 | 9.8 本章小结 | 310 |
| 9.2.2 由搜索驱动的 Web 内容 | | 第 10 章 Web 内容管理 | 313 |
| 管理 | 260 | 10.1 分类法和信息架构师 | 314 |
| 9.2.3 由搜索驱动的社交功能 | 261 | 10.1.1 管理 | 315 |
| 9.2.4 由搜索驱动的 | | 10.1.2 网站导航 | 315 |
| e-Discovery | 262 | 10.2 页面创建 | 318 |
| 9.3 UX 层的工作 | 264 | 10.3 搜索驱动的发布 | 319 |
| 9.3.1 搜索中心组件 | 264 | 10.3.1 网站发布目录 | 319 |
| 9.3.2 搜索 Web 部件 | 265 | 10.3.2 跨网站集发布 | 321 |
| 9.3.3 内容搜索 Web 部件 | 266 | 10.4 创建和编辑内容 | 321 |
| 9.3.4 结果类型和显示模板 | 266 | 10.4.1 Ribbon 菜单的增强功能 .. | 321 |
| 9.3.5 使用托管属性 | 270 | 10.4.2 发布字段控件 | 322 |
| 9.3.6 悬停面板 | 273 | 10.4.3 图片呈现形式 | 324 |
| 9.3.7 本节小结——使用结果 | | 10.5 动态显示内容 | 325 |
| 类型和显示模板 | 274 | 10.5.1 内容搜索 Web 部件 | 325 |
| 9.4 使用查询 | 274 | 10.5.2 搜索驱动 Web 部件 | 326 |
| 9.4.1 查询处理 | 275 | 10.6 以编程方式访问内容 | 327 |
| 9.4.2 预定义查询链接 | 276 | 10.6.1 服务器端 API | 327 |
| 9.4.3 使用 REST 和 CSOM 来 | | 10.6.2 CSOM | 328 |
| 查询搜索 | 276 | 10.6.3 REST CSOM | 329 |
| 9.4.4 搜索查询语法 | 278 | 10.7 网站品牌建设 | 329 |
| 9.4.5 查询生成器 | 281 | 10.7.1 母版页 | 330 |
| 9.4.6 查询规则 | 283 | 10.7.2 页面布局 | 330 |
| 9.5 使用内容 | 287 | 10.7.3 组合外观 | 330 |
| 9.5.1 内容捕获——爬网和 | | 10.7.4 在 Express Web 中进行 | |
| 连接器 | 287 | 品牌自定义 | 332 |
| 9.5.2 安全性修整 | 293 | 10.8 设计管理器 | 335 |
| 9.5.3 定制内容填充 | 296 | 10.8.1 将设备通道用于移动体验和 | |
| 9.5.4 使用联合和结果源 | 300 | 设备定向 | 336 |

| | | | | | |
|---------|-----------------------------------|-----|--------|----------------------------------|-----|
| 10.8.2 | 使用设计文件和资源 | 338 | 11.5 | 本章小结 | 399 |
| 10.8.3 | 转换 HTML 母版页 | 339 | 第 12 章 | 企业文档管理 | 401 |
| 10.8.4 | 代码段库 | 340 | 12.1 | 文档管理思维模式 | 402 |
| 10.8.5 | 部署封装 | 343 | 12.1.1 | ECM 功能 | 402 |
| 10.9 | 使用情况分析 | 343 | 12.1.2 | 扩展的 ECM 对象模型 | 405 |
| 10.9.1 | 架构与改进 | 344 | 12.2 | 挖掘 Document Center 的 最大价值 | 406 |
| 10.9.2 | 在代码中运用使用事件 | 344 | 12.3 | 内容路由 | 408 |
| 10.9.3 | 扩展使用事件 | 345 | 12.4 | 在 Document Center 中使用 文档库 | 411 |
| 10.10 | 多语言支持 | 346 | 12.4.1 | 元数据导航与筛选 | 412 |
| 10.11 | 本章小结 | 347 | 12.4.2 | Visual Studio 和文档库 | 414 |
| 第 11 章 | 在 SharePoint 2013 中使用 InfoPath | 349 | 12.5 | 记录管理 | 416 |
| 11.1 | 培训管理应用程序介绍 | 350 | 12.5.1 | 记录标识 | 416 |
| 11.2 | 定制 SharePoint 列表表单 | 351 | 12.5.2 | 审计与报告 | 418 |
| 11.2.1 | 定制 SharePoint 列表 表单 | 351 | 12.5.3 | 记录管理 API | 419 |
| 11.2.2 | InfoPath 控件 | 353 | 12.5.4 | 记录的声明与取消声明 | 419 |
| 11.2.3 | 使用规则和视图来创建 业务逻辑 | 354 | 12.5.5 | 创建管理器规则 | 420 |
| 11.2.4 | 发布列表表单 | 357 | 12.6 | eDiscovery 与合规性 | 424 |
| 11.3 | 设计 InfoPath 表单模板 | 359 | 12.6.1 | eDiscovery 中心网站 模板 | 424 |
| 11.3.1 | 表单库 | 360 | 12.6.2 | SharePoint 2013 中的 eDiscovery | 425 |
| 11.3.2 | 设计表单模板 | 360 | 12.6.3 | 创建 SharePoint 2013 合规性 | 428 |
| 11.3.3 | 查询表单中的 SharePoint 列表 | 365 | 12.6.4 | 保留计划 | 431 |
| 11.3.4 | 查询 REST Web 服务 | 367 | 12.7 | CMIS | 436 |
| 11.3.5 | 提交行为 | 370 | 12.8 | 本章小结 | 436 |
| 11.3.6 | 表单编程 | 372 | 第 13 章 | 业务连接服务 | 439 |
| 11.3.7 | 发布 InfoPath 表单 | 379 | 13.1 | 业务连接服务简介 | 440 |
| 11.3.8 | 表单安全性 | 382 | 13.1.1 | 创建简单的 BCS 解决方案 | 441 |
| 11.3.9 | 沙盒化表单 | 383 | 13.1.2 | 创建简单的 O365 解决方案 | 445 |
| 11.3.10 | 表单解析 | 383 | 13.2 | 理解 BCS 架构 | 447 |
| 11.3.11 | 在代码中处理表单 XML | 388 | 13.2.1 | 理解连接器 | 447 |
| 11.4 | 表单开发人员工具 | 396 | 13.2.2 | 理解业务数据连接 | 447 |
| 11.4.1 | Rule Inspector | 396 | 13.2.3 | 理解安全存储服务 | 452 |
| 11.4.2 | Design Checker | 397 | | | |
| 11.4.3 | InfoPath JavaScript 工具 | 398 | | | |

| | | | | | |
|--------|------------------------------|-----|---------|---|-----|
| 13.2.4 | 理解部署包 | 454 | 15.1.1 | 工作流 | 520 |
| 13.2.5 | 理解应用程序架构 | 455 | 15.1.2 | 工作流活动 | 520 |
| 13.3 | 使用 BDC 元数据模型 | 455 | 15.1.3 | 工作流操作 | 523 |
| 13.4 | 使用外部数据源 | 457 | 15.1.4 | 工作流表单 | 524 |
| 13.4.1 | 使用 SQL Server 连接器 进行连接 | 458 | 15.1.5 | 工作流任务 | 526 |
| 13.4.2 | 使用 WCF 服务连接器 进行连接 | 462 | 15.1.6 | 工作流历史 | 527 |
| 13.5 | 创建方法 | 465 | 15.1.7 | 工作流流程控制模板 | 527 |
| 13.5.1 | 实现方法原型 | 465 | 15.1.8 | 工作流作用域 | 529 |
| 13.5.2 | 为数据库创建方法 | 469 | 15.1.9 | 工作流关联和订阅 | 530 |
| 13.5.3 | 为 Web 服务创建方法 | 475 | 15.1.10 | 工作流事件侦听器 | 531 |
| 13.5.4 | 定义关联 | 476 | 15.1.11 | 工作流可视化 | 532 |
| 13.6 | 使用外部列表 | 480 | 15.1.12 | 工作流授权 | 533 |
| 13.6.1 | 创建自定义列表操作 | 481 | 15.2 | 了解 SharePoint 2013 中的 工作流平台架构 | 533 |
| 13.6.2 | 创建自定义表单 | 481 | 15.3 | 了解关键的工作流开发 工具 | 535 |
| 13.7 | 本章小结 | 482 | 15.3.1 | Visio Professional 2013 | 535 |
| 第 14 章 | 高级业务连接服务 | 483 | 15.3.2 | SharePoint Designer 2013 | 537 |
| 14.1 | 创建 .NET 程序集连接器 | 483 | 15.3.3 | Visual Studio 2012 | 539 |
| 14.1.1 | 了解项目工具 | 484 | 15.4 | 设置开发的前提条件 | 540 |
| 14.1.2 | 开发过程演示 | 485 | 15.4.1 | 创建工作流和 SharePoint 开发环境 | 540 |
| 14.2 | 封装的注意事项 | 500 | 15.4.2 | 安装工作流管理器 | 540 |
| 14.3 | 启用搜索支持 | 501 | 15.4.3 | 安装开发工具 | 541 |
| 14.4 | 使用 BDC 服务器运行时 对象模型 | 504 | 15.5 | 使用 Visio 2013 进行工作流 建模 | 541 |
| 14.4.1 | 连接到元数据目录 | 505 | 15.6 | 使用 SharePoint Designer 2013 开发工作流 | 543 |
| 14.4.2 | 检索模型元素 | 506 | 15.6.1 | 设置网站和列表的 先决条件 | 543 |
| 14.4.3 | 执行操作 | 507 | 15.6.2 | 导入 Visio 工作流图表 | 544 |
| 14.5 | 在 SharePoint 应用程序中 使用 ECT | 514 | 15.6.3 | 创建 Vacation Request 列表工作流 | 545 |
| 14.5.1 | 了解应用程序级别的 ECT | 515 | 15.6.4 | 发布工作流 | 553 |
| 14.5.2 | 了解 BCS 客户端对象 模型 | 516 | 15.7 | 使用 Visual Studio 2012 开发 自定义工作流活动和操作 | 553 |
| 14.6 | 本章小结 | 517 | 15.7.1 | 创建自定义声明式活动和 操作 | 555 |
| 第 15 章 | 在 SharePoint 2013 中开发 工作流 | 519 | | | |
| 15.1 | 核心工作流概念 | 520 | | | |

| | | | | | |
|---------|--|-----|---------|---|-----|
| 15.7.2 | 开发自定义代码活动和 操作 | 559 | 16.3.12 | 使用 Report Builder 3.0 打开报表 | 583 |
| 15.8 | 使用 Visual Studio 2012 开发 工作流 | 560 | 16.4 | 缓存和快照 | 584 |
| 15.9 | 关键的开发注意事项 | 561 | 16.4.1 | 存储的凭据 | 585 |
| 15.9.1 | SharePoint 2010 与 SharePoint 2013 工作流 对比 | 561 | 16.4.2 | 参数管理 | 586 |
| 15.9.2 | 声明式与编程式工作流 对比 | 561 | 16.4.3 | 规划 | 587 |
| 15.9.3 | 考虑工作流的执行位置 | 561 | 16.4.4 | 报表缓存 | 588 |
| 15.9.4 | 决定是否将 SharePoint 2010 工作流转换成 SharePoint 2013 工作流 | 562 | 16.4.5 | 管理缓存刷新计划 | 588 |
| 15.9.5 | SharePoint Designer 与 Visual Studio 2012 对比 | 562 | 16.4.6 | 快照 | 589 |
| 15.9.6 | 选择顺序工作流还是 状态机工作流 | 563 | 16.4.7 | 缓存与快照之间的差异 | 591 |
| 15.10 | 本章小结 | 563 | 16.5 | SharePoint 数据报表 | 592 |
| 第 16 章 | 集成报表服务 | 565 | 16.5.1 | 创建示例列表 | 593 |
| 16.1 | 历史沿革 | 565 | 16.5.2 | 构建报表 | 593 |
| 16.2 | SSRS 2012 介绍 | 566 | 16.5.3 | 查询大型列表 | 594 |
| 16.2.1 | 制作工具 | 567 | 16.6 | 多个区域 | 595 |
| 16.2.2 | 配置 BI 中心 | 570 | 16.7 | 匿名访问报表 | 595 |
| 16.3 | 构建和部署报表 | 571 | 16.8 | Reporting Services 执行 账户 | 595 |
| 16.3.1 | 制作报表 | 571 | 16.9 | 本章小结 | 597 |
| 16.3.2 | 报表布局 | 573 | 第 17 章 | 在 SharePoint 2013 中开发 Excel 应用程序 | 599 |
| 16.3.3 | 数据可视化效果 | 574 | 17.1 | 新功能 | 601 |
| 16.3.4 | Tablix | 575 | 17.1.1 | 客户端更新 | 601 |
| 16.3.5 | 将报表发布到 SharePoint | 576 | 17.1.2 | 定制功能增强 | 601 |
| 16.3.6 | 发布报表部件 | 578 | 17.1.3 | Excel Service API 的 新改进 | 601 |
| 16.3.7 | 报表查看器 Web 部件 | 579 | 17.2 | 仪表板与 Excel 混搭 | 601 |
| 16.3.8 | 局限性 | 581 | 17.3 | Excel | 605 |
| 16.3.9 | 可连接的报表查看器 Web 部件 | 581 | 17.3.1 | 快速填充 | 605 |
| 16.3.10 | ATOM 数据源 | 581 | 17.3.2 | Web 服务函数 | 605 |
| 16.3.11 | 报表服务数据警告 | 582 | 17.4 | Excel Web 应用程序 | 606 |
| | | | 17.5 | Office 的 Excel 应用程序 | 607 |
| | | | 17.5.1 | Excel 任务面板 应用程序 | 608 |
| | | | 17.5.2 | Excel 内容应用程序 | 609 |
| | | | 17.6 | Excel Services | 610 |
| | | | 17.6.1 | Excel Services 架构 | 610 |
| | | | 17.6.2 | Excel Services 数据访问 | 610 |

| | | | | | |
|--------|-------------------------------------|-----|--------|---|-----|
| 17.6.3 | REST API | 616 | 18.4.1 | PerformancePoint 2013 的 可扩展性目标 | 668 |
| 17.6.4 | Excel Services Web 访问 | 623 | 18.4.2 | 自定义数据源 | 669 |
| 17.6.5 | JavaScript 对象模型 | 623 | 18.4.3 | 自定义报表、筛选器和 转换 | 676 |
| 17.6.6 | 用户定义函数 | 627 | 18.4.4 | 部署定制项 | 677 |
| 17.6.7 | SOAP Web 服务 | 632 | 18.5 | 本章小结 | 678 |
| 17.7 | 本章小结 | 632 | 第 19 章 | 使用 Access 开发应用程序 | 679 |
| 第 18 章 | PerformancePoint 仪表板 | 633 | 19.1 | Access 2013 | 681 |
| 18.1 | 商务智能 | 633 | 19.2 | Access Services | 687 |
| 18.1.1 | 历史沿革 | 634 | 19.2.1 | 内部部署架构 | 687 |
| 18.1.2 | SharePoint 2013 商务智能 组件 | 636 | 19.2.2 | 托管架构 | 688 |
| 18.1.3 | 重要的 BI 术语和概念 | 640 | 19.2.3 | 升级到 Access 2013 的 注意事项 | 688 |
| 18.1.4 | 商务智能中心 | 641 | 19.2.4 | 数据库组件 | 689 |
| 18.2 | PerformancePoint 2013 介绍 | 643 | 19.3 | 使用 Access Services 构建 应用程序 | 695 |
| 18.2.1 | PerformancePoint 服务 架构 | 644 | 19.3.1 | 先决条件 | 696 |
| 18.2.2 | PerformancePoint 2013 的 变化 | 645 | 19.3.2 | 创建 Access Web 应用 程序 | 702 |
| 18.3 | 使用 PerformancePoint 2013 | 647 | 19.3.3 | 部署 Access 应用程序 | 708 |
| 18.3.1 | 配置 PerformancePoint 服务 | 647 | 19.4 | 本章小结 | 709 |
| 18.3.2 | 使用 AdventureWorks 示例数据库 | 648 | 附录 | 额外帮助与资源 | 711 |
| 18.3.3 | 创建仪表板 | 652 | | | |
| 18.3.4 | 还有什么可以开箱即用 | 663 | | | |
| 18.4 | 扩展 PerformancePoint | 668 | | | |

第 1 章

SharePoint 2013 架构概览

本章内容

- ? 了解内部部署服务器场(on-premise server farm)架构
- ? 使用服务应用程序架构来部署、配置和发布应用
- ? 获悉搜索架构在可扩展性和冗余性方面的改进
- ? 探讨 SQL Server 数据库架构
- ? 了解云托管的架构

Microsoft SharePoint Server 2013 引入了许多需要了解的新功能，其目的是为了在 SharePoint 2013 平台上编写更好的应用程序。开发新的功能依赖于健全的逻辑和物理架构。因此，必须很好地领会和理解 SharePoint 场架构，以实现为你所用并开发长效的 SharePoint 解决方案。

本章对 SharePoint 2013 可用的常见内部部署服务器场架构进行了简要概述。详细介绍服务应用程序，并深入探讨改进后的 SharePoint 2013 搜索架构。之后介绍有关 SQL 数据库层的改进和更新。最后解释云托管的服务器场架构。

本章介绍的内容面向架构师、开发主管以及为 SharePoint 2013 场拓扑量身定制解决方案的开发人员，但对任何使用 SharePoint 2013 进行工作的人来说这些内容也很有帮助。尽管本章所有的主题都很重要，但其结构旨在使你能够快速定位到感兴趣的部分。

1.1 从架构角度看有哪些新内容

从架构角度看，SharePoint 2013 的拓扑进行了很多改进。这些新增和改进内容将持续推动 SharePoint 平台功能的优化，从而更好地处理 SharePoint 平台上不断增加的工作任务。

关键的更新包括以下几个方面。

- ？ SQL 改进和零碎存储——数据库层改进了不少，减少了可能调用全表扫描的场景所带来的影响，对 SQL Server 2008 R2 和 SQL Server 2012 高级索引功能的使用做了改进，集成了一项名为零碎存储的新功能，该功能改变了 SharePoint 存储和更新 SQL 中文档的方式。现在文档是碎片式的，只有改变的碎片会在数据库层进行更新。这就降低了文档更新带来的影响。
- ？ 分布式缓存服务——基于 Windows Server AppFabric 分布式缓存，SharePoint 2013 提供了一种新的缓存服务。所有 Web 前端和应用服务器均默认启用。它通过缓存诸如社交数据验证令牌的信息来提高性能。
- ？ 统一的搜索架构——SharePoint 2013 整合了 SharePoint 2010 的所有搜索功能。SharePoint 2013 搜索在内容爬网、内容处理、分析处理、索引、查询处理和搜索管理组件方面进行了许多改进。
- ？ 集成请求管理(RM)——请求管理为 SharePoint 提供了更多应对传入请求的认知与控制能力。这包括限制流量和将请求路由到合适的 Web 前端，请求的优先级处理和过滤，以及基于加权方案的负载均衡。
- ？ 新的服务应用程序——新的服务应用程序包括：支持和管理 SharePoint 2013 中应用程序的应用程序管理服务、支持文件自动化语言翻译的机器翻译服务及提供任务聚合功能的工作管理服务。
- ？ Office Web 应用程序现在成为单独的产品——Office Web 应用程序独立成为一个专门的产品，提供一个用于查看和编辑文档的统一应用程序，这也包括那些不在 SharePoint 中的文档。Office Web 应用程序服务器支持 Web 应用程序开放平台接口 (WOPI)，SharePoint 实现了该接口以便对 SharePoint 托管的办公文档进行支持。
- ？ Web 分析平台——Web 分析平台取代了 SharePoint 2010 中的 Web 分析服务应用程序。它已被完全重新设计并集成到 SharePoint 2013 的搜索服务应用程序中。
- ？ Windows Azure 工作流——现在 SharePoint 2013 的内部部署和托管式部署均支持 Windows Azure 工作流。

1.2 内部部署服务器场架构

服务器场体现了向最终用户提供 SharePoint 服务的拓扑。服务器场是一个协同工作以承载 SharePoint 服务和工作任务的服务器集合。

SharePoint 2013 为拓扑规划提供了高度的灵活性。要实现服务器场架构，其背后的核心原则在于根据需求扩展硬件环境的能力，以支持额外的工作负荷、方案和由组织机构安置在场上的负载。

服务器场可以小到单台 SharePoint 服务器，也能大到在物理服务器专用集合上进行服务器角色托管的高度向外扩展的架构。图 1-1 显示了一个典型的中型 SharePoint 服务器场，

就如同 TechNet 上“SharePoint Server 2013 拓扑：模型”(<http://go.microsoft.com/fwlink/p/?LinkId=259119>)一文中描述的那样。

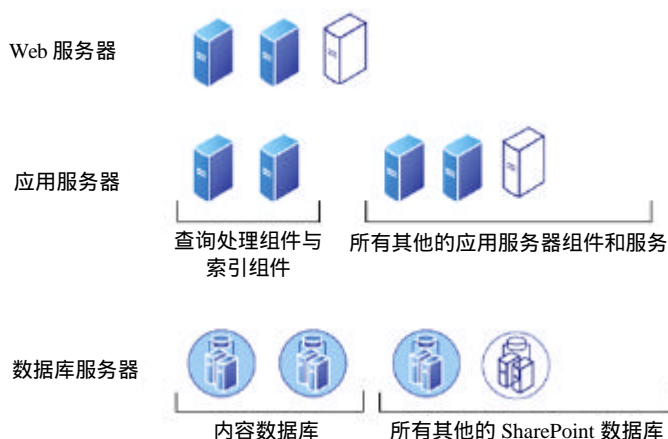


图 1-1

拓扑中的每一层都表明对应层内托管服务器的用途，或专用于那些服务器的服务。服务器场的核心组件可以归纳为以下三层(参见图 1-1)。

- ？ Web 服务器——Web 服务器是用来响应用户请求和呈现 SharePoint Web 页面的服务器。场内的所有 Web 服务器均互为镜像且实现负载均衡。
- ？ 应用服务器——应用服务器是运行后端应用程序的服务器(例如，承载搜索爬网和查询组件的服务器)。可以对多台冗余应用服务器进行负载均衡。
- ？ 数据库服务器——数据库层将几乎所有场数据承载在 SQL 数据库中。这包括配置数据库、与服务应用相关的数据库和内容数据库。所有数据库都可分配到一台数据库服务器或者分布到多台服务器中。

SharePoint 2013 可以用多个拓扑配置来部署。基本拓扑结构包括小型、中型和大型——也称为单层、双层和三层部署——用来定义服务器场拓扑中单台服务器的位置和用途。

1.2.1 Web 服务器层

Web 服务器层由接收和响应 HTTP 请求的 Web 服务器或其他服务器构成。Web 服务器在互联网信息服务(Internet Information Service, IIS)中对 SharePoint Web 应用程序进行托管。它们可以支持附加的服务，如搜索查询组件，用来将请求发送到数据库服务器层中的数据库服务器，或者与应用服务器层的应用服务器通信以使用由这些服务器托管的服务。Web 服务器层的服务器直接面向最终用户，所以应确保有防火墙保护或在内部网络中运行。

1.2.2 应用服务器层

应用服务器层是一个可选层，由专用于托管 SharePoint 2013 相关服务应用程序的服务器组成。举例来说，应用服务器层的服务器，除了对 PerformancePoint 或 Excel Services 这样的服务进行托管外，还包括专用于托管搜索服务、管理和查询组件的服务器。

应用服务器层通常与大型服务器场环境关联度最高，大型服务器场需要专用计算资源以支持大量搜索查询、大型索引或将服务应用程序与 Web 服务器层隔离以释放资源支持高并发率。

1.2.3 数据库服务器层

数据库服务器层由托管 SQL Server 的服务器组成。数据库层的数据库服务器响应由 Web 和应用服务器发起的请求，并对支持 SharePoint 2013 的基础数据库进行更新。数据库服务器层既能纵向扩展(以提高性能)，也能横向扩展(以提高性能并提供额外的服务器场弹性保障)。

1.2.4 小型或单层拓扑

小型或单层拓扑通常由单台服务器部署而成，其中包括 SharePoint 环境所需的所有组件，安装在包括数据库服务器在内的一台机器上。图 1-2 显示了一个单层拓扑示例，它常用于开发环境或收缩性与冗余度相对没那么重要的中小型企业。

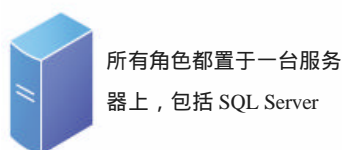


图 1-2

单层拓扑不提供任何级别的冗余性。因此，它需要实现积极的备份与还原策略，这是该拓扑部署模式所能提供的数据保护的最大限度。由于所有组件都安装在一台服务器上，所以导致单层拓扑最不灵活，也不支持无缝扩展。

1.2.5 中型或双层拓扑

中型或双层拓扑由两台或多台支持 SharePoint 和 SQL Server 组件隔离的服务器组成。这包括一台或多台安装了 SharePoint 2013 的 Web 服务器，以及一台或多台安装了 SQL Server 的数据库服务器。中型或双层拓扑受益于其灵活性，可无缝扩展以满足不断变化的业务需求或组织机构的要求。

图 1-3 显示了一个最小的双层拓扑结构，由一台在 Web 层运行 SharePoint Server 2013 的 Web 服务器和一台在数据库服务器层运行 SQL Server 2008 R2 SP1 或 SQL Server 2012 的数据库服务器组成。

图 1-4 显示了一个规模扩大的双层拓扑，它包括两台在 Web 服务器层运行 SharePoint Server 2013 且做了负载均衡的 Web 服务器和两台在数据库服务器层运行 SQL Server 2008 R2 SP1 或 SQL Server 2012 的数据库服务器，这两台数据库服务器可以是集群或者镜像，目的是提供高可用性和冗余。



图 1-3

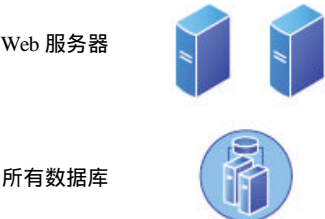


图 1-4

双层拓扑提供了最灵活的部署类型，并推荐所有规模的企业将其作为基本拓扑。这种拓扑可通过添加或移除服务器而得以扩展或收缩。因此，这是一种最常见的服务器场部署，其提供了一个灵活、可扩展的解决方案。双层服务器场使得组织能够无缝地实现硬件或软件的负载均衡，如 Windows NT 负载均衡服务(Windows NT Load Balancing Service ,WLBS)，将传入的 HTTP 请求均匀分布到各 Web 服务器上。这提供了一种方法以满足随着提交的请求量增多而增加的需求(例如，公司合并或收购所带来的业务量增加)。

双层服务器场还可以在数据库服务器层无缝扩展，这基于引入镜像或集群配置中的额外数据库服务器来实现。这样就提供了在服务器场环境内额外的弹性和负载分布机制。

1.2.6 大型或三层拓扑

大型或三层拓扑专为那些要求性能、规模和一致性的业务连续性管理目标的大型组织机构而设计。

图 1-5 显示了一个三层拓扑，包含两台或多台安装了 SharePoint 2013 的 Web 服务器，一台或多台安装了 SharePoint 2013 的应用服务器，以及两台或多台安装了 SQL Server 的数据库服务器。



图 1-5

为 SharePoint 2013 选定的物理拓扑将决定服务应用程序拓扑的布局。很多情况下，三层拓扑可能更容易将服务应用程序拓扑映射到物理拓扑，以帮助确保有足够的资源支持整体部署。

1.2.7 基于地理位置的分布式拓扑

按地理位置分散部署涉及分配 SharePoint 资源以支持区域或全球用户。例如，一个组织可能把其总部设在华盛顿州的西雅图。然而，其大量用户可能分布在全球各地以支持各种企业职能，或者响应特定地理位置的商业机会。

这种情况下，部署一个 SharePoint 2013 的专用实例来支持小众用户将得不偿失。因此，该组织可能选择引入 WAN 优化设备，无论对称与否，以适应延迟或支持使用如 Windows Server 2008 R2 和 Windows Server 2012 中的分支机构缓存这样的技术。

在这样的使用场景中，如果地理位置分散的用户群基数庞大到足以平衡区域化、专用 SharePoint 2013 部署的成本，组织机构就可以选择联盟或从中央服务器场发布服务应用程序到分布式区域的服务器场中。这为远程用户群提供了统一的体验。可以有选择地隔离这些服务器场以支持遵从特定地理位置相关的法规。

1.3 服务应用程序架构

本节旨在帮助你理解 SharePoint 2013 中的服务。目的是让你熟悉 SharePoint 2013 中的服务应用程序架构，以及如何运用此架构在平台上提供新的和改进的功能。

1.3.1 服务应用程序模型

SharePoint 2013 沿用了早在 SharePoint 2010 中就引入的服务应用程序模型。从 SharePoint 2010 开始且延续到 SharePoint 2013 中，SharePoint Foundation 2013 提供了托管服务应用程序的基础架构。图 1-6 显示了 SharePoint 2010 和 2013 中的服务应用程序模型。

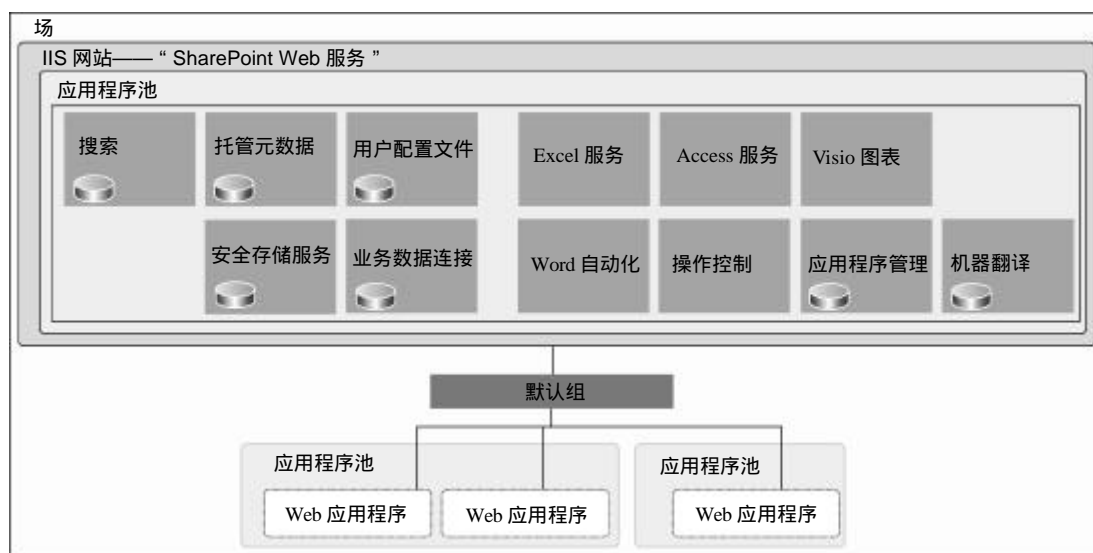


图 1-6

SharePoint 2013 中服务应用程序模型的概念很简单。如果你不需要某个服务应用程序，

不将其部署到场就行了——就是如此简单！另外，可以部署多个服务实例。实际上，可以随心所欲地创建任意数量的特定服务应用程序实例。

服务模型第二层粒度配置位于 Web 应用程序级别。在 SharePoint 2013 中，Web 应用程序可以与想要使用的服务应用程序一起进行配置，以进行组合使用。

在部署服务应用程序实例到服务器场后，它可以被同一个场中的多个 Web 应用程序共享，甚或在不同的场之间共享。无论共享模型怎样，部署好后始终可以修改 Web 应用程序和服务应用程序之间的关联。

服务应用程序可以部署到不同的应用程序池以支持进程隔离。可以选择哪些服务应用程序应该在同一进程中，或在单独的进程中运行。



提示：跨多个应用程序共享服务的数据，是可能需从性能或安全层面考虑进程隔离的一个原因。

图 1-7 显示了如何将不同的服务分布在两个应用程序池中。



图 1-7

尽管在大多数实现方案下，服务仅存在于一个应用程序池中对于服务器场的性能是最佳优化的方式；但是对于某些方案，还需要服务的高级物理隔离。SharePoint 2013 服务应用程序模型使你能够创建服务应用程序的不同实例，并将它们放在不同的应用程序池中。

服务应用程序提供了更好的扩展性模型。可以在管理中心 Services on Server 页面选择哪个服务器来托管和运行某个服务应用程序的服务。

SharePoint 2013 提供了一个基础的负载均衡器，它使用轮询算法将请求发送到服务应用程序。当 Web 应用程序为相关服务应用程序请求一个端点(通过代理)时，开箱即用的负载均衡器将返回首个可用端点。某些服务(如 Excel Calculation Services)提供自带的软件负载均衡功能，以确保在任何时间都没有指定服务的实例出现超负荷运行的情况。

SharePoint 2013 支持跨服务器场的服务应用程序。换句话说，任何场都可以发布服务应用程序，也可以使用来自其他场的服务应用程序。每一个场可以使用来自多个父场的服务。这使得 SharePoint 2013 场中的 Web 应用程序既能使用本地的服务应用程序，也能使用

远程的服务应用程序。

1.3.2 可用的服务应用程序

作为架构师或开发人员，你必须知道你的 SharePoint 授权版本提供了哪些服务应用程序。表 1-1 提供了 SharePoint 2013 不同版本自带哪些服务应用程序的概览，这并不包括其他微软产品提供的服务应用程序，如 Project Server、PowerPivot 服务等。

表 1-1 SharePoint 2013 不同版本可用的服务应用程序对比

| 服务应用程序 | 存 储 类 型 | SharePoint Foundation | SharePoint Standard | SharePoint Enterprise |
|---------------------------------------|---------|--------------------------|------------------------|--------------------------|
| Access Services | 应用程序 DB | | | ? |
| Access Services 2010 | 内容 DB | | | ? |
| App Management Service | DB | ? | ? | ? |
| Business Data Connectivity Service | DB | ? | ? | ? |
| Excel Services | 缓存 | | | ? |
| Machine Translation Services | DB | | ? | ? |
| Managed Metadata Service | DB | | ? | ? |
| PerformancePoint | 缓存 | | | ? |
| PowerPoint Automation | | | ? | ? |
| Search Service | DB | | ? | ? |
| Secure Store Service | DB | | ? | ? |
| State Service | DB | | ? | ? |
| Usage and Health Data Collection | DB | ? | ? | ? |
| User Profile | DB | | ? | ? |
| Visio Graphics Service | 博客缓存 | | | ? |
| Word Automation Services | DB | ? | ? | ? |
| Work Management Service | DB | | ? | ? |
| Subscription Settings Service | DB | ? | ? | ? |



提示：Office Web Application Services 现在是一个单独的产品，并且不再作为服务应用程序提供服务。Web 分析服务应用程序现在作为搜索服务应用的一个关键组件来托管。

以下是每个服务应用程序的说明。

- ? Access 服务——此服务应用程序可以创建、查看、编辑以及使用 Access 2013 Office 客户端或在浏览器中进行交互。
- ? Access 服务 2010——此服务应用程序可以使用 Access 2010 和 2013 Office 客户端来持续维护 SharePoint 2010 Access 服务应用程序。但用户不能使用该服务来创建新的应用程序。
- ? 应用程序管理服务——应用程序管理服务使你能从内部应用目录或公共 SharePoint-Store 安装应用程序。
- ? 业务数据连接服务——业务连接服务(Business Connectivity Service , BCS)允许你上传数据(BDC)模型 ,这些数据模型定义了其他企业的业务线(Line-Of-Business ,LOB)系统接口 , 并且该服务允许连接这些系统。
- ? Excel 服务——此服务应用程序允许查看来自浏览器内的 Excel 文件并与其交互。
- ? 机器翻译服务——该服务提供文件和网站的自动机器翻译功能。
- ? 托管元数据服务——该服务应用程序使你能够管理 SharePoint 2013 的分类层次结构、关键字和社会标签功能。此服务应用程序还负责处理跨网站集内容类型的发布。
- ? PerformancePoint——此服务应用程序作为与 SharePoint 2013 企业版集成的商务智能(BI)产品 , 支持 PerformancePoint 的配置和监控。
- ? PowerPoint 自动化服务——该服务应用程序使得服务器端的演示文稿能在各种文件格式之间转换。
- ? 搜索服务——顾名思义 , 该服务应用程序(来自其自身的拓扑管理配置)用来索引内容和提供由用户或自定义代码执行的服务搜索查询。
- ? 安全存储服务——这是一个用来访问其他企业级服务应用程序或后台企业系统的凭据映射服务。
- ? 状态服务——状态服务为处理用户会话的所有数据提供临时存储。
- ? 使用情况和健康数据收集——此服务应用程序提供了服务器场级别的存储使用情况和健康信息 , 并提供基于这些数据的各种报表功能。
- ? 用户配置文件——用户配置文件服务应用程序是 SharePoint 2013 的核心服务应用程序之一。它支持的功能包括 : 我的站点、我的链接、同事追踪器、配置页面、个人标记和注释 , 以及其他社交功能。
- ? Visio 图形服务——该服务应用程序允许查看、交互并刷新浏览器内的 Visio 图表。
- ? Word 自动化服务——该服务应用程序使你能够在 Web 浏览器中查看和编辑 Word 文档。它也可用于文档转换。
- ? 工作管理服务——工作管理服务允许与关键用户相关的信息聚合到一个中心位置。该服务支持提供程序模型 , 从而使其他系统能够使用此服务。
- ? 订阅设置服务——这是 SharePoint 2013 平台提供的多租户功能的关键组件。

现在你熟悉了 SharePoint 不同版本的服务应用程序 , 接下来就介绍服务应用程序的生命周期。

1.3.3 服务应用程序的生命周期

服务应用程序的典型生命周期一般分为数个阶段。当规划服务应用程序时，需要考虑该生命周期的每个阶段。比如，需要清楚何时使用配置向导或者 Windows PowerShell 来配置服务应用程序，以及何时需要为服务应用程序创建自定义代理组。

图 1-8 显示了服务应用程序生命周期的阶段。

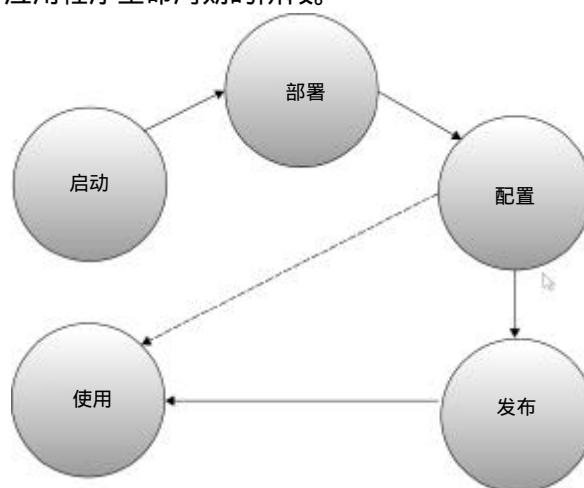


图 1-8

1. 启动服务

虽然服务应用程序与服务不同，但是仍然让很多使用 SharePoint 2013 的人感到困惑。

打开 SharePoint 管理中心的 Services on Server 页面，该页面列出了可在服务器场指定服务器上启动和停止的所有服务，如图 1-9 所示。

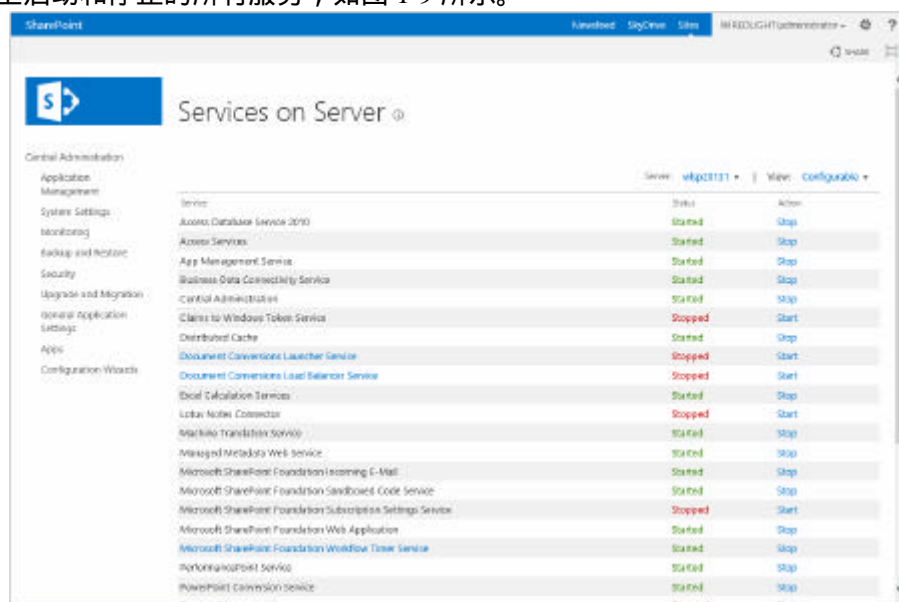


图 1-9

这些服务大多是 Windows 服务的 SharePoint 包装器，且并不一定需要相关的服务应用程序。例如，管理中心实际上只是在服务器场内某台服务器上启动的一项服务，启动后即可将其转变成可以托管管理中心网站的服务器——并没有任何服务应用程序与其关联。

本章之前介绍过，服务应用程序代表了以特定方式配置和共享的某项服务的具体实例。服务应用程序由 Windows 服务、计时器作业、缓存、SQL 数据库和其他部分组成。服务应用程序只是一个比 Windows 服务更广泛的概念而已。

2. 部署服务应用程序

使用以下方法在服务器场内部署服务应用程序：

- ？ 在场的初始配置向导中选择服务应用程序；
- ？ 在管理中心网站中添加新的服务应用程序或者现有服务应用程序的新实例；
- ？ 使用 Windows PowerShell。

表 1-2 描述了可用于管理服务应用程序的 Windows PowerShell 命令。

表 1-2 与服务应用程序相关的 Windows PowerShell 命令

| 命 令 | 说 明 |
|--|---|
| Install-SPService | 在服务器场中安装服务。每个场需要运行一次 |
| Get-SPServiceInstance | 对于指定服务器或整个服务器场与管理服务实例相关的操作 |
| Start-SPServiceInstance | |
| Stop-SPServiceInstance | |
| | |
| Get-SPServiceApplication | 与管理部署到服务器场的服务应用程序相关的操作(比如将指定的本地服务应用程序共享到服务器场之外) |
| Publish-SPServiceApplication | |
| Remove-SPServiceApplication | |
| Set-SPServiceApplication | |
| Unpublish-SPServiceApplication | |
| Get-SPServiceApplicationProxy | 与管理服务应用程序代理相关的操作 |
| Remove-SPServiceApplicationProxy | |
| Add-SPServiceApplicationProxyGroupMember | |
| Get-SPServiceApplicationPool | 与管理服务应用程序逻辑架构相关的操作 |
| New-SPServiceApplicationPool | |
| Remove-SPServiceApplicationPool | |
| Set-SPServiceApplicationPool | |

无论部署方式如何，服务应用程序都可以隔离。为此，在配置过程期间，可指定使用

现有的应用程序池，或创建一个新的应用程序池，使服务应用程序在各自的工作进程中运行。

3. 配置服务应用程序

在服务器场级别配置好服务应用程序后，可以在管理中心网站对其进行管理。单击 Manage Service Applications 菜单，屏幕上将出现 Service Applications 页面，如图 1-10 所示。

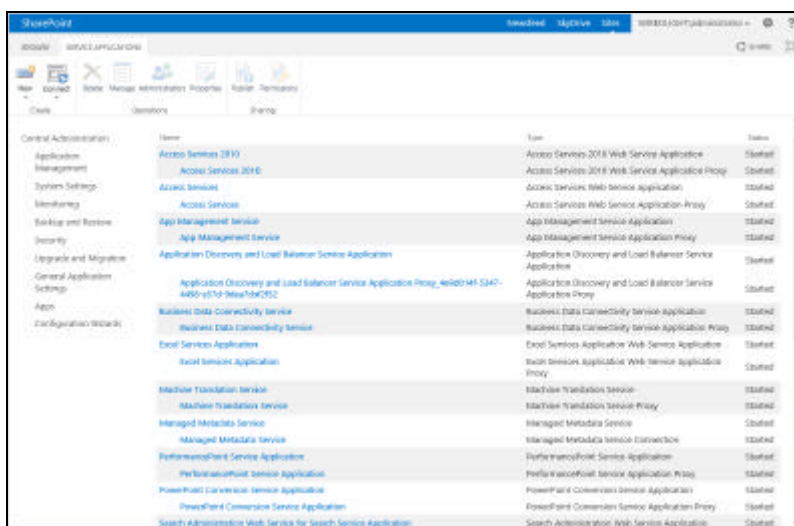


图 1-10

在 Service Applications 页面，应注意以下三件事。

- ？ 所有部署好的服务应用程序均全部列出。
- ？ 所有服务应用程序连接(代理)均全部列出。
- ？ 在 Ribbon 菜单上单击 New 按钮来添加新的服务应用程序。

设置好服务应用程序后，打开 Internet 信息服务(IIS)管理器，可以看到一个叫做 SharePoint Web 服务的 Web 应用程序，其下是一些虚拟目录。每个虚拟目录的名称都是一个全局唯一标识符(GUID)，或该服务应用程序的标识符，如图 1-11 所示。



图 1-11

从服务数据库层面看，大部分服务应用程序都使用其专用的数据库集。



提示 要记住的重要一点是，一个服务应用程序可能有一个或多个数据库。例如，用户配置文件服务应用程序包括了配置文件、同步和社会标签数据库。另一个例子是，搜索服务应用程序包括了爬网、链接、分析和管理数据库。如果数据库容量规划得不合理，数据库的数量将迅速增加，导致难以管理。

使用配置向导配置服务应用程序的一个问题是，相关的虚拟目录数据库名称可能会以一大串 GUID 结尾。例如，某用户配置文件数据库的名称可能是 User Profile Service Application_ProfileDB_899fd696a54a4cbe965dc8b30560dd07。

虽然在某些情况下这可以接受，但一般来说，更直观的命名约定会更有意义。解决此问题的一种方法是，在管理中心网站的 Manage Service Applications 页面中单独添加地服务应用程序，然后指定有意义的数据库名称。其他的替代方法是使用 Windows PowerShell 来配置服务应用程序。

下面的代码段显示了如何使用 Windows PowerShell 配置状态服务应用程序。注意如何在代码中指定 SQL Server 数据库和服务名。

```
New-SPStateServiceDatabase -Name "StateServiceDatabase" -DatabaseServer
"dhsqslsrv" | New-SPStateServiceApplication -Name "State Service
Application" | New-SPStateServiceApplicationProxy -Name "State Service
Application Proxy" -DefaultProxyGroup > $null
```

如前所述，可以创建和部署自己的服务应用程序。要这样做，可以重写前面的 Windows PowerShell 命令并添加自己的参数。

4. 配置服务应用程序代理

如果使用配置向导或通过管理中心来部署服务应用程序，那么会自动创建服务应用程序代理。如果使用 Windows PowerShell，则必须手动创建服务应用程序代理。

那么，到底什么是服务应用程序代理呢？

本质上，服务应用程序代理是将 Web 应用程序链接到某特定服务应用程序的一个虚拟链接。所以，创建 Web 应用程序时，可以指定关联到服务应用程序的代理，实际上是由该代理来管理往返通信的。

除了将 Web 应用程序链接到服务应用程序外，一些代理也包括来自服务应用程序的可以独立修改的设置。例如，托管元数据服务应用程序的代理，表明关联的服务应用程序是否为企业分类储存的默认存储位置(如关键词和指定列术语集)，如图 1-12 所示。



提示：如果托管元数据服务应用程序有多个实例(因此有多个代理)，那么必须指定一个托管企业分类存储的主实例。其余实例作为次要实例，为主数据提供额外的数据。有一个例外，与托管元数据服务应用程序一同工作的 Web 部件，可以使用所有实例的数据。

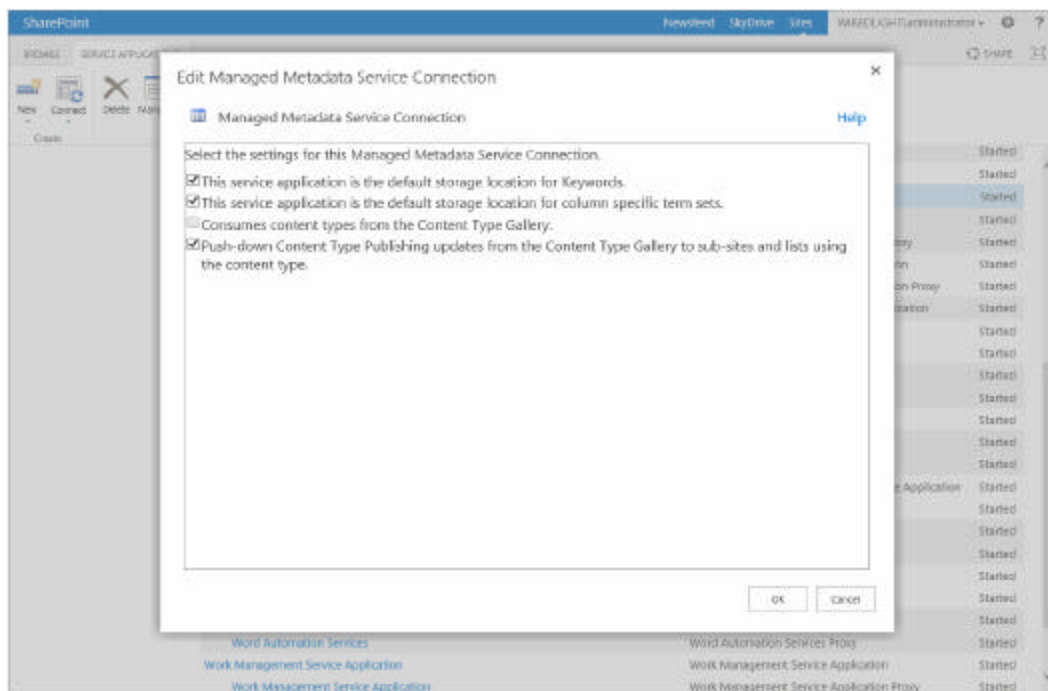


图 1-12

5. 配置代理组

顾名思义，服务应用程序代理组是为 Web 应用程序选择的一组服务应用程序代理。一个服务应用程序代理可以包含在多个代理组中，或基于目标 Web 应用程序的要求，代理组可以选择不包括某个服务应用程序代理。

当设置服务器场时，一个包含所有服务应用程序代理的默认代理组也将创建。在创建 Web 应用程序的过程中，可以选择默认代理组或创建自定义代理组。图 1-13 显示了在默认代理组中配置的服务应用程序列表。



提示：为某个 Web 应用程序创建的自定义服务应用程序代理组不能与其他 Web 应用程序进行关联。

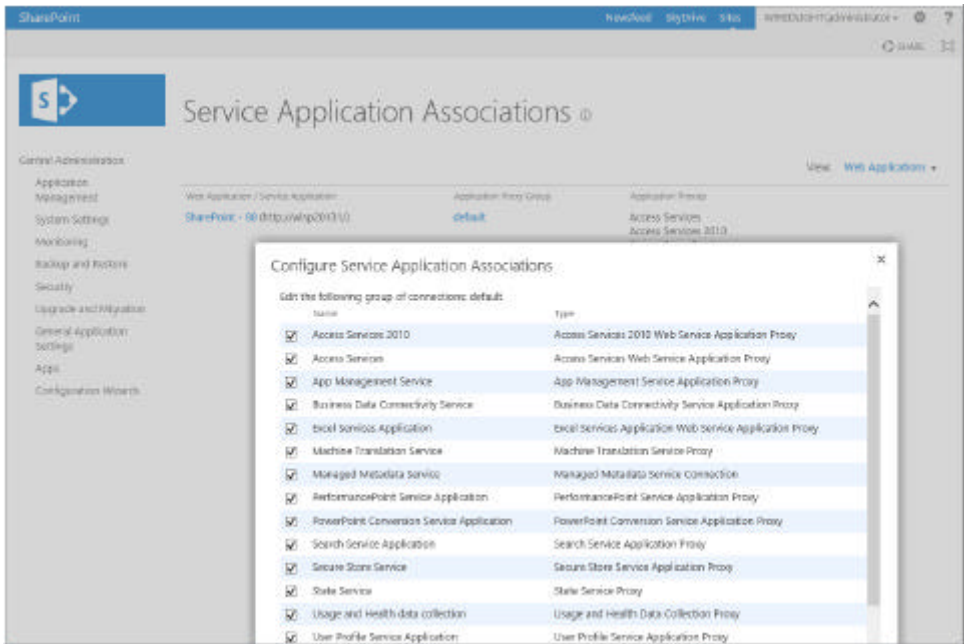


图 1-13

可在 Windows PowerShell 中运行 `Get-SPServiceApplicationProxy` 命令,如图 1-14 所示,从而列出服务应用程序代理的 ID。然后,可使用 `Remove-SPServiceApplicationProxy`(以 ID 作为参数)来删除服务应用程序代理,或使用 `Add-SPServiceApplicationProxy GroupMember` 将新的服务应用程序代理添加到代理组中。

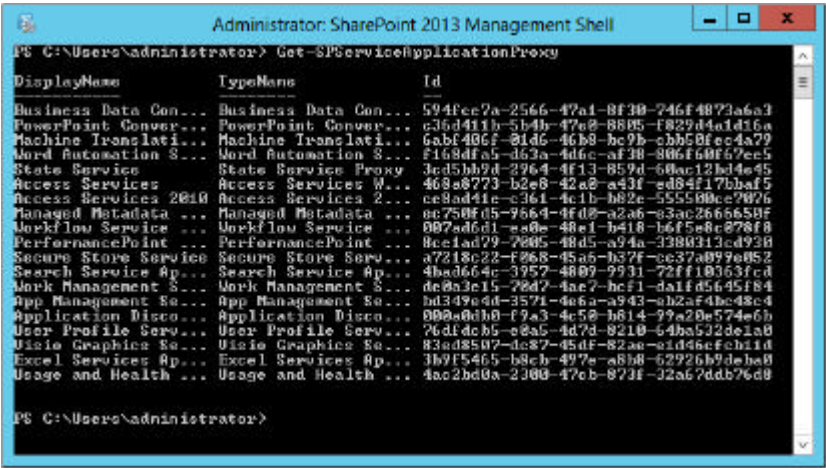


图 1-14

6. 使用服务应用程序

默认情况下,本地服务器场的所有 Web 应用程序都与默认代理组关联。这意味着不必担心在本地场中使用服务,因为这些服务已经自动设置好了。如果决定要创建自定义代理组,那么你必须判断你想要该指定的 Web 应用程序如何使用服务应用程序。

要更改 Web 应用程序的默认代理组,必须选择管理中心网站中的 Application Management,再单击 Configure Service Application Associations。在 Service Application Association 页面中,可以看到应用程序代理组标题下的默认文本。单击此文本,将出现能管理该默认代理组成员的页面。此外,如果各个 Web 应用程序都有自定义代理组,则这些代理组都将会列出在该页面上。

值得再次一提的是,一些连接可能包含可以修改的设置。例如,如果 Web 应用程序连接到托管元数据服务的多个实例,则必须指定哪一个服务应用程序托管企业分类。

7. 发布服务应用程序

一个服务应用程序可以被本地服务器场中的一个或多个 Web 应用程序使用,也可以被远程服务器场中的 Web 应用程序使用。

在介绍更多细节之前,请了解一些术语,以确保有一个清晰的认识:

? 发布服务应用程序——这意味着让某服务应用程序可以跨服务器场使用。

? 跨场服务应用程序——这是远程场可以使用服务应用程序。

概括来说,要部署场的服务应用程序,有三项工作必须做。

(1) 必须确保托管服务应用程序的场和需要使用服务应用程序的场进行证书交换以取得彼此信任。

(2) 必须发布该服务应用程序。要发布一个服务应用程序,必须在管理中心的 Manage Service Applications 页面的 Ribbon 菜单中,单击 Publish 按钮。然后将转向 Publish Service Application 页面,在那里可以明确一些设置,如图 1-15 所示。

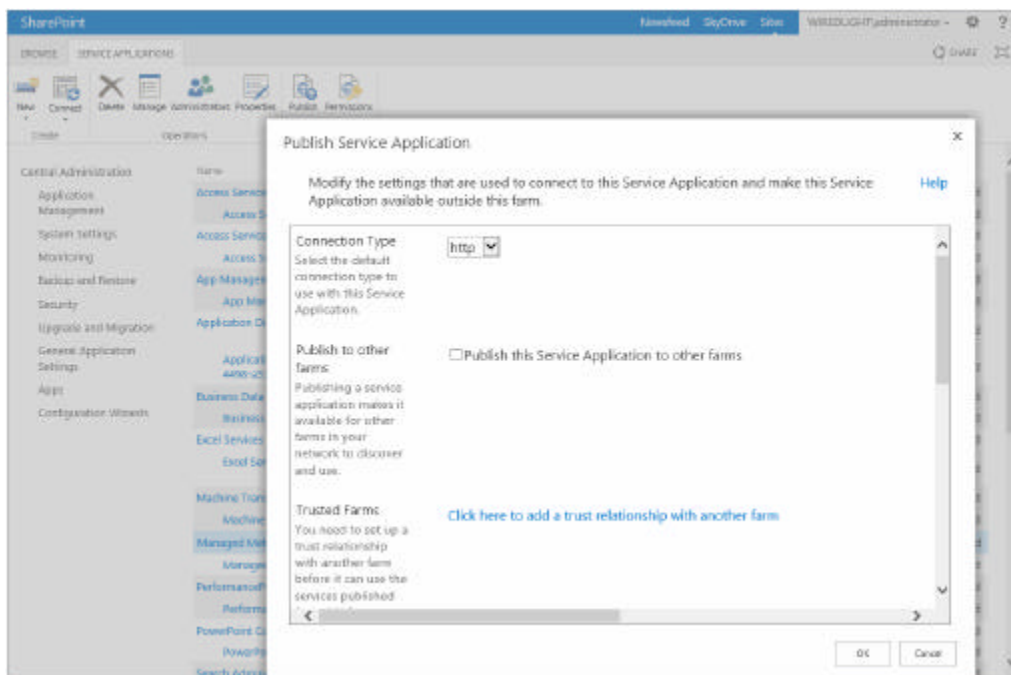


图 1-15



提示 :必须强调的是在 Publish Service Application 对话框中可用的 Publish URL。这是在远程场中使用的 URL，用来定位该服务应用程序。

(3) 要使用已经发布的服务，可以打开远程场中的 Manage Service Applications 页面，单击 Ribbon 菜单上的 Connect 按钮。然后选择需要连接的服务类型，接下来会提示你输入已发布服务的 URL，如图 1-16 所示。假设信任已经建立并正常工作，则只需创建本地场的服务应用程序代理来连接到远程场上的服务应用程序。有了代理，本地场中的任何 Web 应用程序都可以使用来自远程场的服务应用程序。

不是所有服务应用程序都可以在场间共享的。例如，BCS 是一个跨场服务应用程序，而其他服务应用程序并非是为了在场间共享而设计的。不推荐在广域网(WAN)环境中使用一些跨场服务应用程序。简而言之，那些使用 Windows Communication Foundation(WCF)端点的服务应用程序就是使用 ASMX Web 服务的服务应用程序。

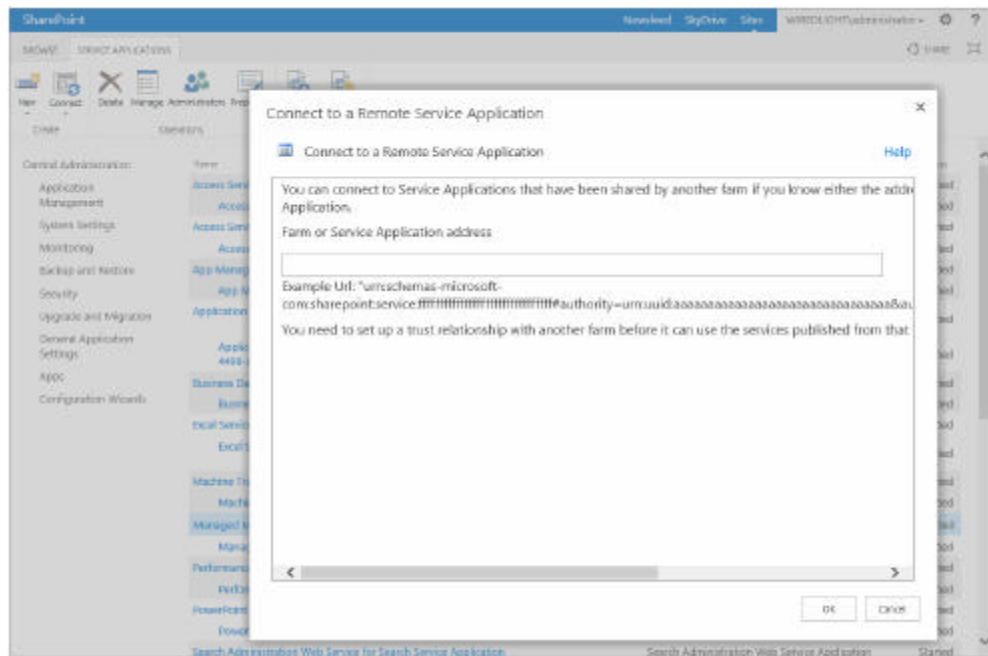


图 1-16

表 1-3 列出了跨服务器场或者在 WAN 中部署服务应用程序的一些建议。

表 1-3 部署服务应用程序的建议

| 服务应用程序 | 跨 场 支 持 | WAN 友好性 |
|------------------------|---------|---------|
| Access Services | 不支持 | N/A |
| Access Services 2010 | 不支持 | N/A |
| App Management Service | 不支持 | N/A |

(续表)

| 服务应用程序 | 跨 场 支 持 | WAN 友好性 |
|------------------------------------|---------|---------|
| Business Data Connectivity Service | 支持 | 有限制 |
| Excel Services | 不支持 | N/A |
| Machine Translation Services | 支持 | 支持 |
| Managed Metadata Service | 支持 | 支持 |
| PerformancePoint | 不支持 | N/A |
| PowerPoint Automation | 不支持 | N/A |
| Search | 支持 | 支持 |
| Secure Store Service | 支持 | 不支持 |
| State Service | 不支持 | N/A |
| Usage and Health Data Collection | 不支持 | N/A |
| User Profile | 支持 | 不支持 |
| Visio Graphics Service | 不支持 | N/A |
| Word Automation Services | 不支持 | N/A |
| Work Management Service | 不支持 | N/A |
| Subscription Settings Service | 不支持 | N/A |

1.4 多租户托管架构

多租户架构是一种在同一个 SharePoint 2013 服务器场上为多个租户分别托管其独有部署的能力，这通过隔离数据、运营服务以及在其余租户正在使用同一个场的情况下对一个租户进行管理来实现。

多租户架构的经典(也最准确)定义是为多个组织或客户提供服务的软件的单个实例，将其数据和配置进行虚拟分区，使客户能够在各自定制的应用程序实例内工作。SharePoint Server 2013 所提供的特性和功能有助于支持真正的多租户架构，不仅对托管提供商有用，而且对企业有用。

如果在企业内进行谨慎地计划和应用，多租户托管将是有助于降低成本、复杂性以及全局管理的众多解决方案之一。

在 SharePoint 2013 中，多租户架构要求配置服务器场及其服务应用程序，来支持多个租户。要实现基于多租户的架构，需要结合以下主要功能：

- ？ 网站订阅
- ？ 服务应用程序分区
- ？ 租户管理
- ？ 功能包

现在仔细看一下其中的每一项。

1.4.1 网站订阅

网站订阅是 SharePoint 2013 托管功能集的核心。网站集由其订阅 ID 组合在一起，形成租户的基础。订阅 ID 用于将功能、选项设置和服务分区映射到租户。换句话说，网站订阅可以大致描述为订阅了一组服务分区、设置、和个性功能的网站集合。网站订阅也称为租户。

可将网站订阅处理成内容的松散关联。在对象模型中，网站订阅由 Microsoft.SharePoint.SPSiteSubscription 类表示。

网站订阅的限制和约束如下。

- ？ 网站订阅不能进行跨服务器场的网站集组合。
- ？ 带有跨 Web 应用程序网站集的网站订阅，不能通过租户管理模板来管理(更多相关信息将在 1.4.3 节中介绍)。
- ？ 在单个 Web 应用程序和内容数据库内支持多网站订阅。
- ？ 服务可以分区以便为指定租户提供服务，这使得各个粒度的数据能够隔离。
- ？ 租户可以使用未分区的服务。

1.4.2 服务应用程序分区

通过新的服务应用程序功能提供数据、使用和运营的隔离。将 SharePoint 2013 的许多服务应用程序进行分区的功能，让各个租户在保持相互间逻辑分离的同时，能够使用相同的分区服务应用程序。

要在 SharePoint Server 2013 中新建分区服务应用程序，管理员必须请按照下列步骤操作。

- (1) 使用-partitionmode 标记创建分区服务应用程序。
- (2) 使用-partitionmode 标记创建分区服务应用程序代理。

图 1-17 显示了托管模式下这些概念间的关系。

SharePoint 2013 使用网站订阅的订阅 ID 映射到分区 ID，该分区 ID 代表租户可接触到的数据子集。

图 1-18 显示了 SharePoint 2013 中一个多租户架构的具体实现。该应用场景中有带有两个租户的一个 Web 应用程序，每个租户在同一 Web 应用程序中有数个网站集。该 Web 应用程序使用多租户式服务应用程序，每个租户的服务数据都在后端数据库进行了分区(即数据隔离)。尽管两个租户使用的是同一服务应用程序，但他们看不到其他租户的数据，因为服务数据进行了分区。

关于图 1-18，这里需要强调两个方面。

首先，不是所有服务应用程序都可以分区。因为有些服务不需要存储租户数据，所以它们可以在多个租户之间共享，而没有泄露租户特殊数据的风险。表 1-4 列出了不包含分区功能的服务应用程序。其次，不能分区的服务应用程序不是多租户式服务应用程序。

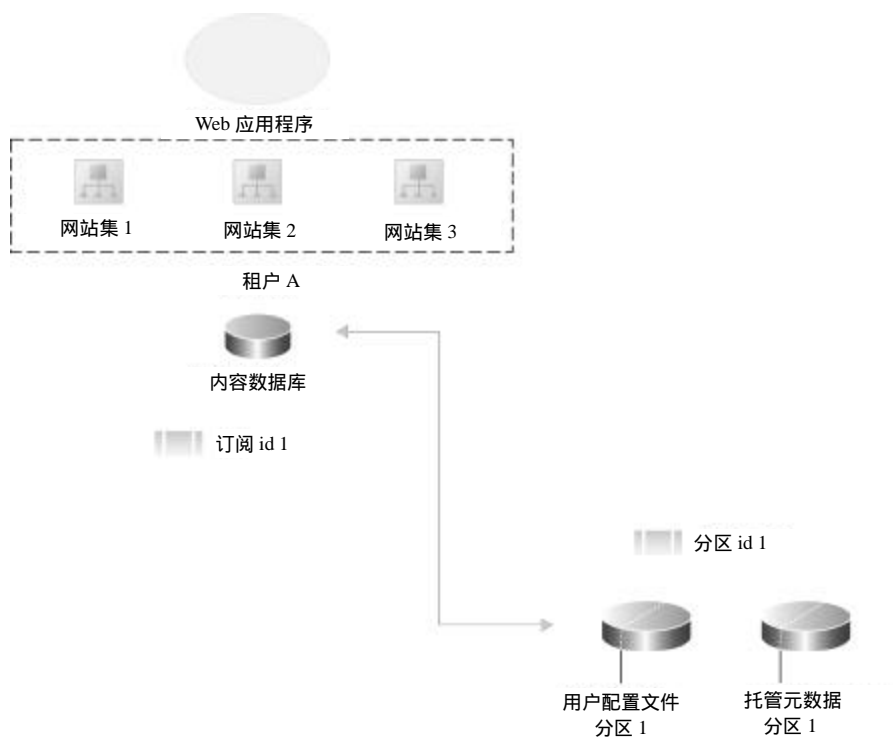


图 1-17

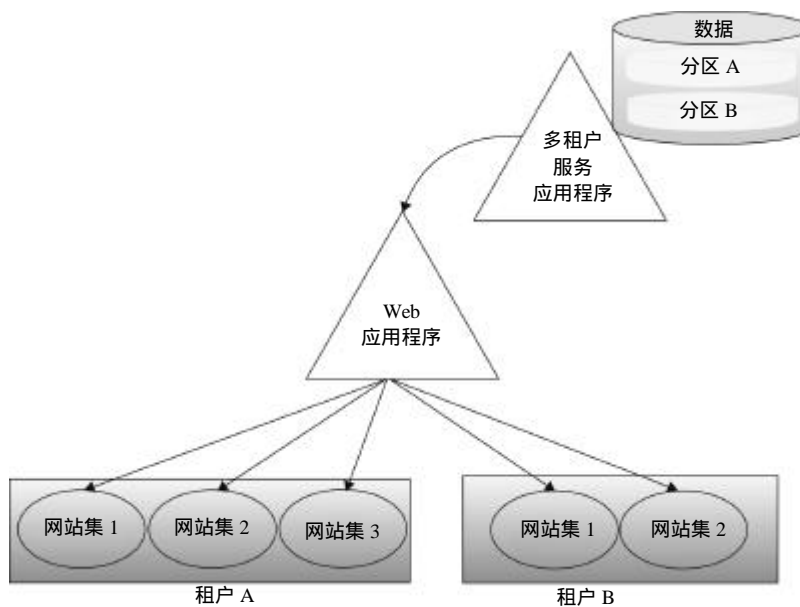


图 1-18

服务应用程序可以配置为在单个数据库中驻留分区数据。

表 1-4 不能分区的服务应用程序

| 服务应用程序 | 多租户支持 |
|------------------------------------|-------|
| Access Services | 不支持 |
| Access Services 2010 | 不支持 |
| App Management Service | 不支持 |
| Business Data Connectivity Service | 支持 |
| Excel Services | 不支持 |
| Machine Translation Services | 支持 |
| Managed Metadata Service | 支持 |
| PerformancePoint | 不支持 |
| PowerPoint Automation | 不支持 |
| Search | 支持 |
| Secure Store Service | 支持 |
| State Service | 不支持 |
| Usage and Health Data Collection | 不支持 |
| User Profile | 支持 |
| Visio Graphics Service | 不支持 |
| Word Automation Services | 支持 |
| Work Management Service | 不支持 |
| Subscription Settings Service | 支持 |

如果决定把所有租户放在一个使用不同网站集的 Web 应用程序中,有一些新的或改进的网站集功能可以随意使用。

- ？ 对于使用主机头网站集的虚名网域提供了额外支持(即在一个 Web 应用程序内的多个根级网站集合)。
- ？ 主机头网站集支持托管路径(比如,租户 A 的网站集 <http://foo.com> 和 <http://foo.com/sites/foo>, 与租户 B 的网站集 <http://bar.com> 和 <http://bar.com/sites/bar>, 可以共存于同一 Web 应用程序中)。
- ？ 额外的通配符现在可用于 SharePoint 2013 的托管主机头路径。
- ？ 包括对负载均衡器单套接字层(Single Socket Layer, SSL)终结点的支持。
- ？ Windows PowerShell 命令 New-SPSite 可以接受一个参数,使你能够将网站集驻留在一个特定的内容数据库中。
- ？ 可插拔式自定义代码(网站创建提供程序)允许跨所有租户执行数据库管理。这主要是为确保如果租户创建一个新的网站集,该网站集将仅限于使用所需的数据库,而不必遵循开箱即用、轮询的算法。

？ 沙盒解决方案允许各租户部署自定义代码到各自的网站集中。

尽管 SharePoint 2013 的分区服务应用程序和网站集的新功能在整个多租户架构中发挥着重要作用，但事实上，很多其他功能在 SharePoint2013 多租户托管架构中也发挥重要作用。以下是其中一些功能。

- ？ Microsoft SharePoint Foundation 订阅设置服务可以为服务应用程序添加多租户功能 (在 SharePoint 所有版本中都可用)。
- ？ 功能集是场管理员允许租户激活和使用的产品功能组。
- ？ 网站订阅是网站集的逻辑组，可以共享设置、功能和服务数据。每个网站订阅有一个订阅 ID，用于将功能、服务和站点映射到租户，并分区其服务数据。
- ？ 集中和委托管理允许将某些管理中心的任务委派给租户管理员，这些租户管理员可以使用自己的管理用户界面，而主体的管理中心网站用于管理整个 SharePoint 安装。

1.4.3 租户管理

网站订阅管理通过一个称为租户管理的新管理网站模板来实现，租户管理用于对使用来自指定订阅服务的网站集进行多方面管理。在单个服务器场环境内支持多个租户，这使 IT 管理员能够集中管理特性与功能的部署。此外，IT 管理员可以向各租户的所有者或业务管理员授权网站集的指定管理控制权。

例如，在托管方案中，托管服务器场环境的组织对服务器场级的设置和配置进行管理。使用者(或租户)可以管理网站集，特别是可以管理委派给他们的特性和功能(如服务)。图 1-19 显示了租户管理的用户界面。

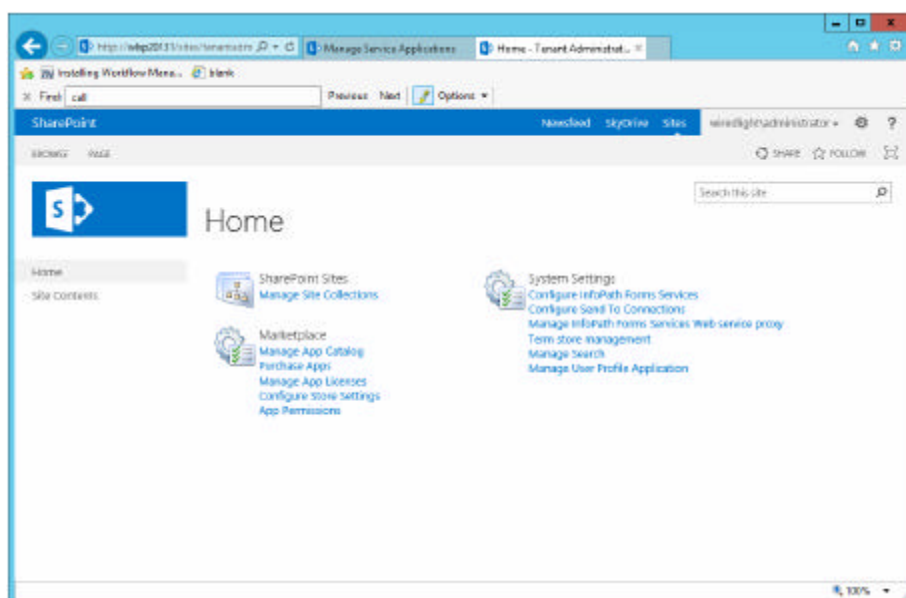


图 1-19

要在 SharePoint Server 2013 中创建新的网站订阅对象，管理员必须创建一个 SPSiteSubscription 对象，然后为 SPSiteSubscription 创建并添加一个 SPSite 对象。

要新建 SPSiteSubscription 对象，请按照下列步骤操作。

(1) 依次单击 Start | All Programs | Microsoft SharePoint 2013 | Microsoft SharePoint 2013 Management Shell 来打开 Microsoft SharePoint 2013 Management Shell。

(2) 在管理工具中，在命令提示符下，请输入以下 Windows PowerShell 命令。

```
$subscription=New-SPSiteSubscription

$site=New-SPSite -Url http://AdventureWorks.com/sites
/TenantAdministration -Template TenantAdmin#0
-OwnerEmail someone@example.com
-OwnerAlias Domain\Username
-SiteSubscription $subscription
Set-SPSiteAdministration -Identity http://AdventureWorks.com/sites
/TenantAdministration
-AdministrationSiteType TenantAdministration
```

1.4.4 功能包

功能包是一种允许开发人员将单项功能(网站或 Web 范围内)的集合组合成一个更大的整体包的方式。功能包向多租户模型的各网站订阅提供功能，可以针对每个租户来允许或阻止其对某些功能或解决方案的访问。

1.5 搜索架构

SharePoint Server 2013 的搜索架构已经大为改进，新的组件集改进了由 SharePoint 服务场提供的可扩展性和冗余性。

从图 1-20 中可以看出，SharePoint Server 2013 的搜索架构现已扩展到 4 个主要方面：爬网和内容处理、分析、索引和查询处理组件以及搜索管理。

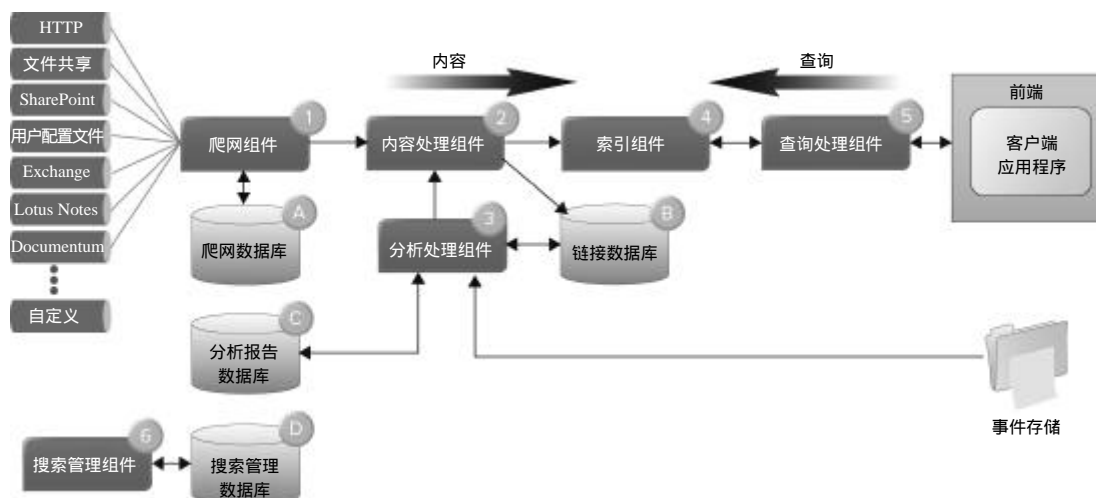


图 1-20

现在让我们来看看这些方面的更多细节。

1.5.1 爬网和内容处理组件

在 SharePoint Server 2013 中，爬网和内容处理架构负责从支持的内容源中抓取内容，将抓取到的条目及其元数据发送到内容处理组件，并进行内容处理。这些可以分成以下组件。

- ？ 爬网组件——爬网组件使用面向目标内容源的关联连接器和协议处理程序来抓取配置的内容源。然后将实际内容和相关元数据传递到内容处理组件。
- ？ 爬网数据库——爬网组件使用爬网数据库来存储有关抓取条目的信息，并且追踪发生过的爬网信息与历史记录。
- ？ 内容处理组件——内容处理组件用于接收条目，使用格式处理程序和 iFilter 处理和解析条目，并将条目转换为可以添加到搜索索引中的内容。这包括将提取的属性映射到使用搜索管理组件定义的属性。
- ？ 链接数据库——链接数据库存储在内容处理过程中获取到的相关链接和 URL 信息。

SharePoint Server 2013 爬网和内容处理搜索架构非常灵活，能够通过将额外的爬网组件实例无缝添加到搜索拓扑来提高爬网和内容处理的效率。

1.5.2 分析处理组件

在 SharePoint Server 2013 中，分析处理组件已经直接集成到搜索架构中了，不再是一个独立的服务应用程序。它可以分为以下组件。

- ？ 分析处理组件——分析处理组件负责处理搜索和基于用户的分析。其通过分析抓取到的条目来实现搜索分析以及通过分析用户如何与这些条目交互来实现使用情况分析。例如，在事件库中检索的用户交互信息，由服务器场中各 Web 前端的用户文件聚合而成并由分析处理组件进行分析。这为更广泛地进一步分析奠定了基础。

- ? 内容处理组件——内容处理组件接收搜索和用户分析结果，并反过来用于更新索引。
- ? 链接数据库——链接数据库存储由内容处理组件提取出来的信息。分析处理组件更新链接数据库来存储更多的分析信息，例如，一个条目被单击的次数。
- ? 分析报告数据库——分析报告数据库存储使用情况分析的结果。

SharePoint Server 2013 分析处理搜索架构具有灵活性，可以通过向搜索拓扑无缝添加额外的分析组件实例来扩展分析处理效率。这无疑缩短了分析处理的完成时间。

1.5.3 索引和查询处理

在 SharePoint Server 2013 中，索引和查询处理架构负责编撰从内容处理组件接收到的经过处理的条目，为查询处理组件处理查询并返回结果集，以及将有变化的索引内容移动到搜索拓扑中。

如图 1-21 所示，SharePoint Server 2013 搜索维护着所有经过处理的内容的索引(包括分析信息)。使用下列功能可以对索引组件进行规模伸缩。



图 1-21

- ? 索引分区——索引分区是整个搜索索引的一个逻辑部分。索引分区启用了水平扩展，从而允许将整个索引分布到服务器场中的多个服务器上。
- ? 索引组件——单个索引分区可由一个或多个索引组件支持。这些索引组件托管索引分区的副本。主索引组件负责更新索引分区，而被动索引组件用于容错功能和增加的查询吞吐量。这实际上支持搜索拓扑的垂直扩展。
- ? 查询处理组件——查询处理组件负责接收来自 Web 前端的查询，分析与处理该查询，并将处理过的查询提交给索引组件。

1.5.4 搜索管理组件

在 SharePoint Server 2013 中,搜索管理组件负责运行基于配置和搜索拓扑的系统进程。搜索管理组件可分为以下组件。

- ？ 搜索管理组件——搜索管理组件执行搜索所需的系统进程,执行对搜索拓扑进行的更改,以及协调搜索拓扑中各种搜索组件的活动。
- ？ 搜索管理数据库——搜索管理数据库存储搜索配置信息。这些信息包括拓扑、爬网规则、查询规则、托管属性映射、内容源以及爬网计划。

1.5.5 多功能(包括搜索功能)中型搜索场拓扑示例

图 1-22 显示了一个中型搜索服务器场拓扑示例。有几个方面需要考虑。首先,“中型”一词指的是可由 SharePoint 拓扑搜索子系统处理的条目的数量。其次,虽然示例中的服务器场显示了 Web 层和应用程序层,但本节将忽略这些内容而纯粹专注于介绍该拓扑搜索方面的内容。

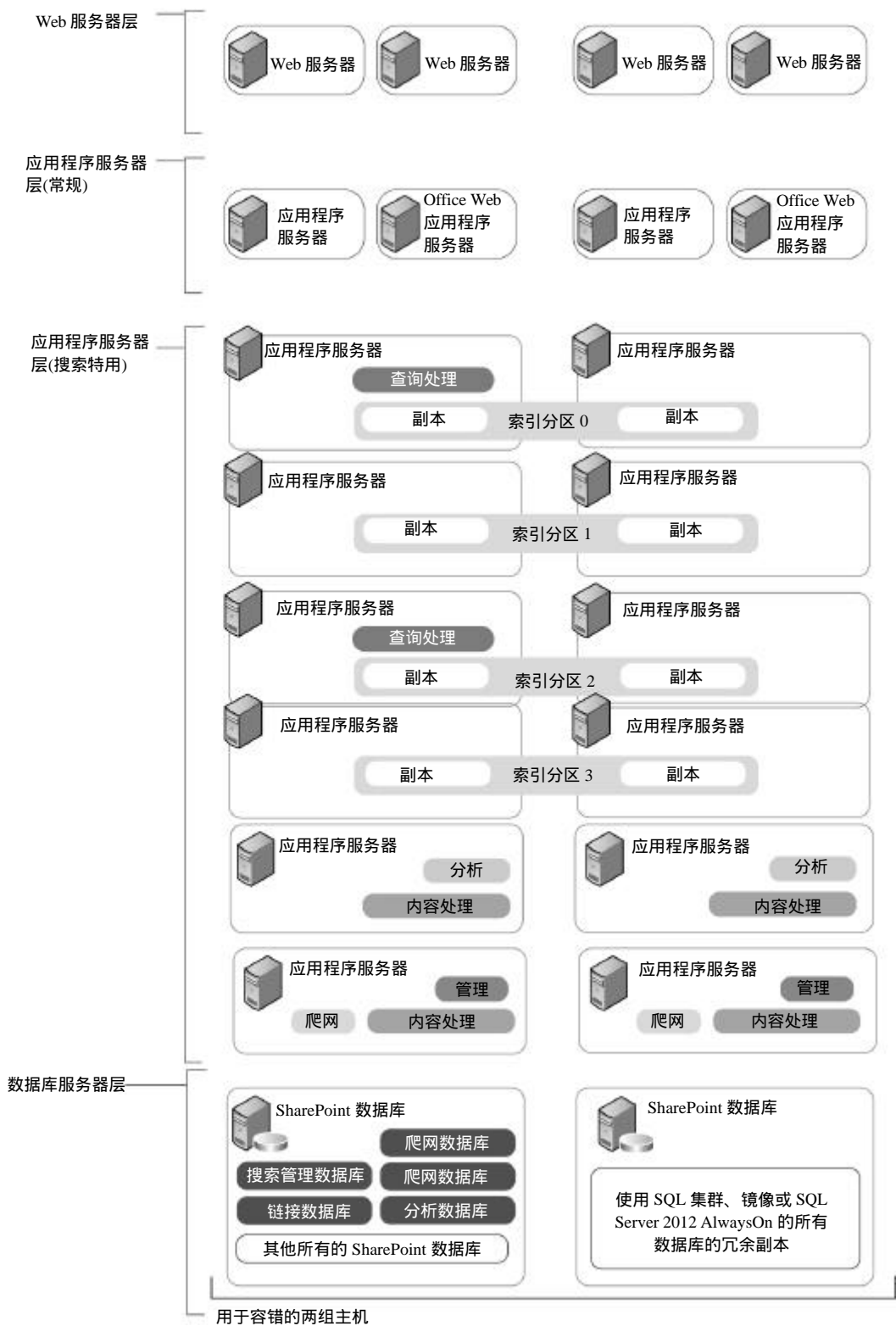


图 1-22

再者，该中型搜索场拓扑基于 TechNet 的建议，参见 <http://www.microsoft.com/en-us/download/details.aspx?id=30383>，可以支持大约 4000 万个条目的索引。看看有关该场拓扑的关键要素。

- ？ 搜索数据库——微软建议在数据库层进行容错，以在一个数据库层失效时维持系统环境的运行。尽管容错选项只能选择 SQL 集群，但请记住，索引中的条目数量越接近 4000 万，越应该考虑将搜索相关的数据库移动到一个专门的 SQL 集群中。
- ？ 管理和爬网组件——管理和爬网组件安置在两台应用程序服务器上。这给管理组件提供了容错功能，并提高了面向多个服务器的爬网操作能力。
- ？ 内容处理组件——来自爬网操作结果的条目由 4 台应用服务器处理。这有助于降低整体爬网时间从而缩短搜索结果的更新周期。
- ？ 分析组件——分析组件安置在两台应用程序服务器上。这提供了在托管分析组件的应用服务器发生故障情况下的容错功能。
- ？ 索引组件——全局搜索索引分成 4 个分区，每个分区驻留在两台应用程序服务器上。这提供了容错能力，并且缩短了查询的延迟时间和提高了查询吞吐量。
- ？ 查询处理——查询处理组件安置在两台应用程序服务器上。这为查询处理组件提供了容错能力，增强了处理来自 Web 服务器层的查询的能力。

1.6 SQL Server 数据库架构

SharePoint Server 2013 很大程度上依赖于精心规划和高性能的数据库层。因此，微软产品组投入了巨大精力来改善 SharePoint 服务器场所使用的数据库的性能和可管理性。

1.6.1 数据库拓扑

为数据库服务器确定适当的拓扑是一个重要的步骤，以确保 SharePoint 2013 平台具备足够的整体性能。当开始规划硬件时，需要清楚 SharePoint 2013 仅支持 64 位的应用程序，这就要求 64 位版本的 Windows 和 SQL Server 产品。Windows Server 2008 R2 Service Pack 1 或 Windows Server 2012 均支持 SQL Server 2008 R2 Service Pack 1 或 SQL Server 2012。

1. 单服务器部署

下列情况下，推荐单服务器部署配置。

- ？ 小到中等规模的用户量，且并发会话数在数据库服务器能力范围之内易于处理。
- ？ 必须开发与 SharePoint 2013 集成的自定义解决方案的开发人员。
- ？ 需要评估 SharePoint 2013 时。

单服务器部署配置最易于安装和维护。默认的 SQL Server 安装选项就是单服务器部署拓扑。在评估期间，如果发现该部署配置符合组织的需要，就应该继续使用，当需求增加时，再升级硬件或添加额外的服务器实例。图 1-23 显示了单服务器部署配置的示例。

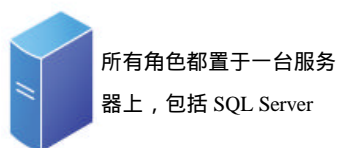


图 1-23

如果使用基于单个 SQL Server 实例的设计，应该考虑使用 SQL Server 连接别名，以允许无缝迁移到新的数据库服务器拓扑(例如，故障转移群集或数据库镜像)。

通过使用连接别名，可保持应用程序配置一致。不过，必须指示基础操作系统如何查找该数据库。

可使用两个实用工具之一创建别名：

- ? SQL Server 配置管理器
- ? SQL Server 客户端网络实用工具



提示：要了解更多信息配置别名的信息，请参阅 <http://msdn.microsoft.com/en-us/library/ms188635.aspx>。

2. 标准服务器部署

在标准服务器部署中，由两台数据库服务器来服务 SharePoint 数据库，以实现故障转移群集服务或数据库镜像设计。图 1-24 显示了一个标准服务器部署配置的示例。



图 1-24

若用户数量适度，也即其需求的处理量全天均匀分布，且服务器处理能力能轻松应对并发会话数，这种情况下推荐使用标准部署配置。

除了具有更大的弹性，标准部署方案还提供了比单服务器部署更好的性能。例如，在数据库镜像设计中，负载可分布于主体和镜像服务器上，因而缓解了包括处理资源在内的常见压力，比如当托管在同一台计算机上时，可以缓解 CPU 时间、内存和磁盘读写面临的

压力。有些 SharePoint 操作是资源密集型的，因此这些操作在单独的服务器上运行可以减少处理资源的竞争。此外，SharePoint 数据库占用的空间起初可能很小，但在运行时，磁盘空间的需求和 I/O 子系统的使用率会显著增加。

基于硬件配置，当决定选择单服务器部署或标准服务器部署时，需考虑以下几点。

- ? 处理资源
- ? 内存资源
- ? 磁盘空间可用量
- ? I/O 性能
- ? 冗余

如果发现该部署配置符合组织的需要，就应该继续使用它，当需求增加时，可再升级硬件或添加额外的服务器实例。

3. 横向扩展的服务器部署

在横向扩展的服务器部署中，故障转移群集或数据库服务器镜像配置的多台 SQL 服务器支持 SharePoint 数据库。这些拓扑包括主动、主动、被动(AAp)型故障转移群集拓扑，或两对独立的数据库镜像。图 1-25 显示了一个横向扩展的服务器部署配置示例。



图 1-25

横向扩展部署允许将工作负载分布到大容量环境中。在横向扩展部署中，该部署中的每台后端数据库服务器称为一个节点。

下列情况下，推荐使用横向扩展的服务器部署配置。

- ? 对于大容量用户负载，其活动以并发用户数来衡量，或者由需要长时间处理或提交的复杂操作(如大数据量的搜索场景)来衡量。
- ? 对于高可用性方案，重要的是 SharePoint 环境不会遇到意外宕机或变得不可用。
- ? 希望提高计划操作或服务应用程序的性能。

通过在作为故障转移群集的一部分的实例上托管 SharePoint 数据库，能够提高环境的容错能力。故障转移群集也可适用于标准部署，但并不常用，因其环境并非高可用性方案配置(如横向扩展部署的环境)。

必须仔细确定并记录可用性需要和测试解决方案，以确保其提供预期的可用性。表 1-5 列出了支持和不支持的高可用性配置。

表 1-5 支持和不支持的高可用性配置

| 配 置 | 支持性说明 |
|---------------------------|-------|
| Failover Clustering | 支持 |
| Log Shipping | 支持 |
| Database Mirroring | 支持 |
| Transactional Replication | 不支持 |
| Merge Replication | 不支持 |
| Snapshot Replication | 不支持 |

1.6.2 SharePoint 2013 数据库

如果你刚接触 SharePoint 2013，将可能惊讶于其使用的数据库的数量。不要着急；这是有意设计并且对 SharePoint 服务器场性能至关重要的。表 1-6 提供了 SharePoint 2013 的数据库概览。

表 1-6 SharePoint 2013 核心的及相关的数据库

| 目 录 | 数据库名称 | 数 量 | 说 明 |
|---------------------------|--------------------------------|-----|--|
| App Management | App License | 单个 | 存储从全局 Store 中下载的应用许可及授权 |
| Business DataConnectivity | | 单个 | 存储外部内容类型及相关的对象 |
| Core Databases | Configuration Database | 单个 | 包含与以下内容相关的数据：所有 SharePoint 数据库、所有互联网信息服务(IIS)网站或 Web 应用程序、受信任的解决方案、Web 部件包、网站模板、Web 应用程序以及为 SharePoint 2013 Preview 指定的场设置，如默认配额及阻止的文件类型 |
| | Central Administration Content | 单个 | 管理中心网站使用的内容数据库 |

(续表)

| 目 录 | 数据库名称 | 数 量 | 说 明 |
|------------------------------|-----------------------|-----|---|
| Core Databases | Content Databases | 多个 | 存储所有的网站内容，包括网站文档或文档库文档、列表数据、Web 部件属性、审计日志、沙盒解决方案以及用户名和权限。一个网站的所有数据都存储在一个内容数据库中。内容数据库可以包括多个网站集。也为 Office Web 应用程序(如果使用)存储数据 |
| Machine Translation Services | | 单个 | 存储与挂起的和已完成的带有已启用的扩展名的批量文档翻译相关的信息 |
| Managed Metadata Service | | 单个 | 存储托管元数据和联合内容类型 |
| PerformancePoint | | 单个 | 存储临时对象以及持久化的用户评论及设置 |
| PowerPivot | | 单个 | 存储按计划更新的数据以及从主要使用情况数据收集数据库中复制过来的 PowerPivot 使用情况数据。SharePoint 2013 的 PowerPivot 需要 SQL Server 2012 分析服务、商务智能或企业版 |
| Project Server | | 单个 | 存储单个 Project Web App(PWA) 的下列所有数据： ? 所有 Project and Portfolio Management(PPM)数据； ? 时间追踪和时间表数据； ? 聚合的 SharePoint 项目网站数据 |
| Search | Search Administration | 单个 | 托管搜索应用程序的配置及爬网组件的访问控制列表(ACL) |
| | Link | 单个 | 存储由内容处理组件抽取出的信息以及单击次数信息 |
| | Analytics | 多个 | 存储使用情况分析报告的结果以及需要时从链接数据库中抽取的信息 |

(续表)

| 目 录 | 数据库名称 | 数 量 | 说 明 |
|-------------------------------|---------------------------|-----|---|
| Search | Crawl | 多个 | 存储爬网数据的状态以及爬网历史 |
| Secure Store | Secure Store | | 存储和映射诸如账号名及密码之类的凭据 |
| Subscription Settings Service | | 单个 | 为托管用户存储功能和设置信息。本数据库并非默认创建且必须使用 Windows PowerShell 或者 SQL Server 来创建 |
| State Service | | 多个 | 为 InfoPath 表单服务、Exchange、图形 Web 部件以及 Visio 服务存储临时状态信息 |
| SQL Server | Master | 单个 | 为 SQL Server 实例存储所有系统级别的信息，这些信息包括登录、配置及其他数据库 |
| | Model | 单个 | 在实例中用作为所有数据库创建的模板 |
| | MSDB | 单个 | 记录操作者，且由 SQL Server 代理按计划告警和执行任务时使用 |
| | TempDB | 单个 | 保存所有临时表和临时存储过程以及满足任何其他存储需要。TempDB 数据库在 SQL Server 实例每次启动时都将重建 |
| Usage and Health | Usage and Health Database | | 临时存储健康监控和使用情况数据，也用于报告与诊断。使用情况数据库是唯一一个可以直接查询及由微软或第三方应用程序修改计划的 SharePoint 数据库 |
| User Profiles | Profile | | 存储和管理用户及其社会化信息 |
| | Synchronization | | 当配置文件数据由如活动目录一样的目录服务同步后用来存储配置与阶段数据 |
| | Social Tagging | | 存储由用户创建的带有其各自 URL 路径的社会化标签与注释 |

(续表)

| 目 录 | 数据库名称 | 数 量 | 说 明 |
|-----------------|-------|-----|-------------------|
| Word Conversion | | | 存储挂起的与已完成的文档转换的信息 |
| Word Automation | | | 存储挂起的与已完成的文档转换的信息 |

SharePoint2013 使用了各种各样的数据库(请参阅表 1-6)。系统环境中要用到的数据库需要综合你所购买的产品授权版本(SKU)以及部署的服务应用程序的情况来考量。恰当的数据库规划能够满足当前和未来规模及性能的相关需求。确保了解每个数据库的使用目的是什么及其特点, 以提供数据库服务器层所需的无缝可扩展性和性能。

1.7 云托管架构

SharePoint Online 可用于两项独特的服务(标准和专用), 根据每个组织的规模、需求和目标来量身定制——这两项服务均以每月每用户为单位来收费。SharePoint Online 服务可在功能、灵活性和定价的基础上体现高度的差异化区别。

SharePoint Online 提供了丰富的特性集和功能集, 可用作组织的主要协作平台, 也可以用来增强组织的现存内部部署以支持轻量级外部网络或外部共享方案, 或者使得在组织防火墙之外的协作成为可能。

SharePoint Online 将 SharePoint 2013 作为云服务, 通过微软数据中心面向全球提供服务, 允许人们分享各自的见解和专业知识, 构建自定义网站和解决方案, 并迅速定位信息以响应不断变化的业务需求——却无须在自己的数据中心部署 SharePoint。除了 SharePoint 提供的服务和解决方案之外, SharePoint Online 提供了高可用性、全面的安全性以及简化的管理方式, 因此组织可以放心选择其作为协作平台。

SharePoint Online 支持一些最复杂的用户分布模式, 无论用户是集成的还是在地理上处于分散状态。无须购买和部署服务器, 企业可以迅速部署 SharePoint 到远程办公室, 或者支持收购导致的系统需求的增长。这种灵活性使用户能以最小的开销和最短的延时而快速受益于 SharePoint。

1.7.1 安全功能

SharePoint Online 通过一组功能集提供了企业级的可靠性和灵活性, 以确保一个安全的协作环境。SharePoint Online 提供了以下一组常见功能。

- ？ 安全访问——SharePoint Online 基于 128 位安全套接字层(SSL)或传输层安全(TLS)加密技术提供服务。
- ？ 入侵监控——SharePoint Online 连续监测不寻常或可疑的活动。

- ？ 审计——微软会定期评估 SharePoint Online 的基础结构，以确保策略设置和防病毒签名可用。配置设置与安全更新包括以下几个方面。
- 达到 ISO 27001 认证标准。
 - 完成关于审计现状(SAS) 70 I 型和 II 型的审计声明。
 - 增加控制以帮助客户遵守美国医治保险携带和责任法案(HIPAA)，以及家庭教育权和隐私权法案(FERPA)。
 - 取得欧盟(EU)信息安全港认证。

1.7.2 身份验证功能

SharePoint Online 提供多种方法用于管理和使用身份验证——不论是缺乏现成身份验证基础结构的中小型企业还是使用活动目录域服务(Active Directory Domain Service, ADDS)的大型组织。

有身份验证基础结构(如 ADDS)的组织，可以通过配置活动目录联合认证服务(Active Directory Federation Service, ADFS)与微软在线服务联盟网关联合，来实现单点登录(Single Sign-On, SSO)验证。来自联盟域的用户身份可以使用其已有凭据为服务进行自动认证。

微软在线服务提供了目录同步工具，以方便进行目录同步。目录同步工具提供到微软在线服务的单向目录同步，同步内容包括所有用户账户和已启用邮件的联系人以及来自本地 ADDS 目录服务的群组。目录同步工具应该安装在连接到 ADDS 林的服务器上，以执行同步并能够访问林中所有域的域控制器。

1.7.3 管理模型

通过基于角色的访问，微软在线服务提供了一种委托和分粒度管理模型。可以为用户账户分配全局或密码管理员权限，以提供所有服务设置的完全访问权限，或者阅读公司和用户信息以及重置用户密码和管理支持请求的功能使用权限。

SharePoint 在线通过 Web 门户进行管理，在该门户上 SharePoint 在线管理员可以创建和管理网站集，如图 1-26 所示。

SharePoint 在线管理 Web 门户相对于整个微软在线服务管理门户是独立的。如图 1-27 所示，SharePoint 在线管理 Web 门户允许管理员管理网站集、配置 Send To 连接、配置 InfoPath 表单服务和 Web 服务代理、管理用户档案，以及管理该服务的网站集所使用的术语库。

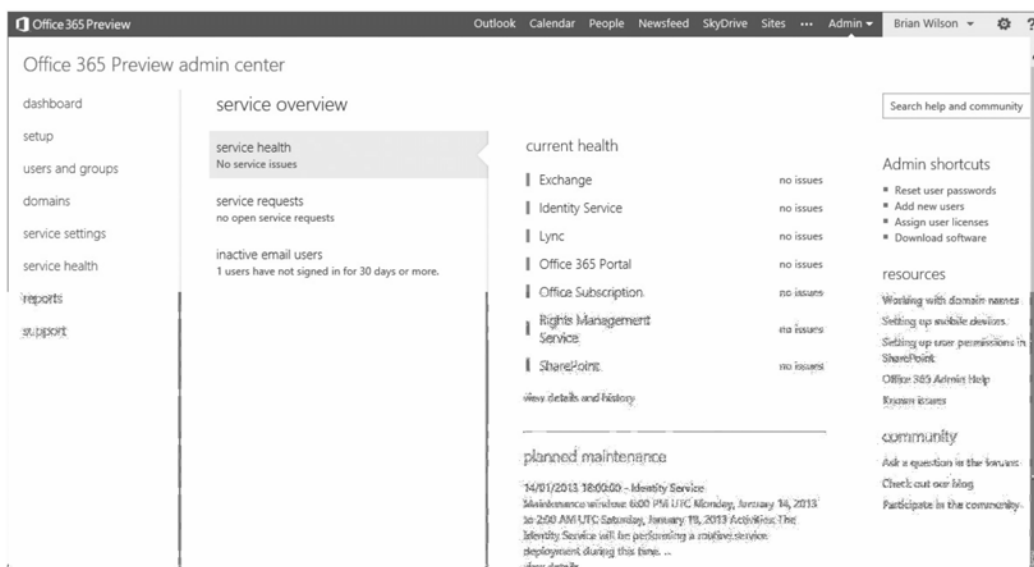


图 1-26

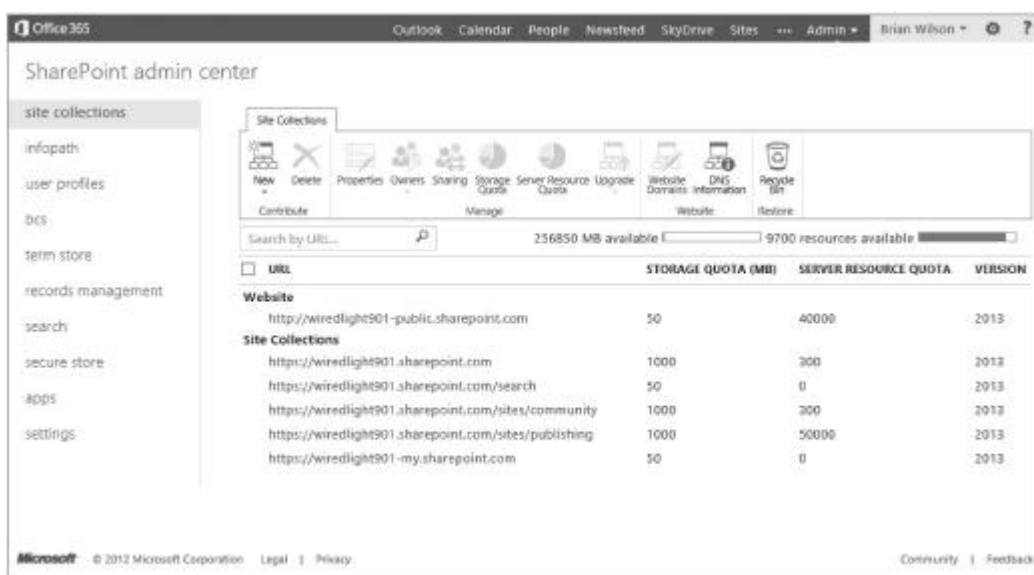


图 1-27

1.8 本章小结

本章提供了 SharePoint 2013 架构的一个直观视角。也提供了一次标准概览，以帮助开发人员和架构师理解他们的自定义设置和插件运行在哪里。透过本书的其余部分，你将了解关于这些新功能的更多细节并掌握如何面向这些功能进行编程，以构建稳健、实用的 SharePoint 2013 应用程序。

第 2 章

SharePoint 2013 的新功能

本章内容

- ? 探讨封装与部署模式的改变
- ? 充分利用新的编程模型
- ? 了解客户端对象模型的数据协议(OData)¹ 投资
- ? 理解 OAuth 开放授权认证协议与应用程序安全模型
- ? 了解通知推送功能
- ? 与 Windows Azure Workflow 集成
- ? 最大限度地使用企业内容管理(ECM)与网站内容管理(WCM)工作负荷
- ? 见证通用可扩展的搜索平台的强大功能
- ? 高效利用改进后的业务连接服务(BCS)

2.1 SharePoint 2013 简介

在微软的历史卷轴上,2012 年无疑是一个革命性的年份。如果你看到微软在这一年发布的内容清单,你将感到无比震惊。除了针对最终用户在功能上进行了增强与增加了大量

1 OData——程序数据库格式标准化的开源数据协议。为了增强各种网页应用程序之间的数据兼容性,微软公司启动了一项旨在推广网页程序数据库格式标准化的开源数据协议(OData)计划,与此同时,他们还发布了一款适用于 OData 协议的开发工具,以方便网页程序开发人员使用。OData 运用且构建于很多 Web 技术之上,如 HTTP、Atom Publishing Protocol(AtomPub)和 JSON,提供了从各种应用程序、服务和存储库中访问信息的能力。OData 用来从各种数据源中暴露和访问信息,这些数据源包括但不限于:关系数据库、文件系统、内容管理系统和传统 Web 网站。

的新功能外，SharePoint 2013 的开发与以往相比也变得超群出众。SharePoint 2013 的变化重心落在了 Web 标准技术上，如 JavaScript、Html、客户端编程以及远程访问。为此，SharePoint 2013 为开发人员提供了通过各类选项配置来扩展产品的开箱即用(out-of-the-box)功能，以实现在构建业务解决方案时无须编写或只须编写少量服务器端代码。

SharePoint 2013 的每个新功能和开发选项都取决于诸多因素，这是开发人员应该预先知道的。希望读者阅读完这些内容后都能够收为己用。

2.2 部署方案

在 SharePoint 2013 中，SharePoint 服务器场的物理位置是一个重要因素，它直接决定了你应该选择哪种开发模式以及有哪些可用选项。作为一名开发人员，首先应该明确的，就是你要构建的自定义解决方案将应用在哪种部署方案中。SharePoint 2013 提供了 4 种不同类型的部署方案：内部部署、Office 365²、托管式部署以及混合式部署。

2.2.1 内部部署

在这种部署方案中，SharePoint 位于公司防火墙之后，并且由 IT 部门来维护。与以往的 SharePoint 版本一样，这种部署方案为开发方式与开发工具的选择提供了极大的灵活性。

2.2.2 Office 365 部署

在这种部署方案中，SharePoint 场部署在 Office 365 上并由微软来托管。除了不能在应用程序中运行服务器端代码外，你可以使用所有与内部部署(方案一)一样的开发方式与开发工具。在 SharePoint 2013 中，应用程序是一种封装和部署代码的新机制，这也是本章会通篇讲述的内容。



提示：本章中 Office 365 和 SharePoint Online(SPO)表达的是一个意思，可以理解为同一个术语。实际上，Office 365 是一个总的概括性术语，它是指一组云计算产品的总称，包括 SharePoint Online、Office Web Apps、Exchange Online、Lync Online 以及 Office Suites。可以申请一个免费的 Office 365 开发授权。要进一步了解，可在此地址 [http://msdn.microsoft.com/en-us/library/fp179924\(v=office.15\).aspx](http://msdn.microsoft.com/en-us/library/fp179924(v=office.15).aspx) 查阅产品文档。

2 Office 365——是微软下一代云计算产品，包括 Office SharePoint Online、Exchange Online 和 Lync Online 组件，并为这些组件提供实时升级服务。Office 365 是唯一一个基于 Web 并且与 Microsoft Office 完全兼容的服务。Office 365 已于 2012 年 6 月 28 日正式发布。简而言之，Office 365 是含有云存储空间的整体 Office 软件。

2.2.3 托管式部署

与 Office 365 部署方案类似，在托管服务部署方案中，SharePoint 服务器场由第三方供应商安装以及管理，该第三方供应商可能是微软也可能是其他厂商。例如，可以将 SharePoint 场完全虚拟化部署到 CloudShare、Amazon EC2 或者微软旗下的 Windows Azure 虚拟机中。根据选择的 SharePoint 场托管服务供应商不同，你面临的情况也将不同，并且你的开发模式以及开发方式也会有所不同。不过，一些第三方的托管服务供应商提供非共享专有云，这将为你提供更多的可选项。

2.2.4 混合式部署

混合式部署是唯一一种从企业防火墙内部覆盖到云端的部署方案。在这种部署方案下，部分安装部署由企业 IT 部门在内部管理、运行和维护，而一部分应用程序则部署到托管在 Office 365 或第三方托管供应商的 SharePoint 场中。

2.3 封装方案

安装和部署 SharePoint 的位置与方式决定着你选择何种封装和部署自定义代码的方案。有三种可供选择的部署方案：完全信任的场解决方案、部分信任的场解决方案以及应用程序部署方案。

2.3.1 完全信任的场解决方案

场解决方案在 SharePoint 2007 中引入，对于 SharePoint 2013，其只能在内部部署或者一些基于云的专有部署方案中使用。这些类型的解决方案可以承载供整个 SharePoint 场使用的定制功能。在完全信任的场解决方案下，自定义代码被部署到 SharePointWeb 应用程序的\Bin 目录或者全局应用程序集缓存(Global Assembly Cache, GAC)中。这些代码可以使用 .NET 的代码访问安全(Code Access Security, CAS)机制进行担保，但通常情况下它们是在完全信任的环境下运行的。该类型解决方案由具有场管理员权限的 IT 专业人士使用控制台访问并且部署到 SharePoint 场中。

2.3.2 部分信任的沙盒解决方案

这种方式在 SharePoint 2010 中引入，可用于所有类型的部署方案。如果在网站集中部署了沙盒解决方案，那么该网站集中的所有网站都可以使用。此类解决方案由网站集管理员进行上传与部署。沙盒解决方案给开发人员带来了不少限制，本章会进行详细介绍。

2.3.3 SharePoint 应用程序

SharePoint 2013 提供了一种新的代码封装和部署方式。这种方式称作应用程序(App)模式，它极具独立性且以小部件的形式提供。如果你使用过智能手机，我想你应该熟悉这个概念。应用程序针对特定的业务构建与部署，其目的是用来处理某些事务。不同于那些

为智能手机定制的应用程序，SharePoint 的应用程序实际上并非存在于 SharePoint 内部，其实际运行过程也并非由 SharePoint 执行。SharePoint 应用程序运行在客户端浏览器或者远程服务器上；这些应用程序通过 SharePoint OAuth 身份认证进行授权从而获得访问 SharePoint 网站的权限，同时，它们也能通过最新改进的 SharePoint 2013 客户端对象模型应用编程接口(CSOM API)进行数据交互。

SharePoint 应用程序、OAuth 身份认证以及最新改进的客户端对象模型(CSOM)将在后续章节进行更详细的介绍。

2.4 编程模型

SharePoint 2013 带来了一种与以往有着根本性区别的编程模型。新的编程模型更多地侧重于云计算、标准的 Web 技术(如 JavaScript 和 HTML)、远程访问。所有新变化都符合微软公司的总体战略——“一切尽在云中”。

在潜心去研究这些新变化之前，首先要重温这几年中 SharePoint 在定制化功能和编程方面的改进。

2.4.1 SharePoint 编程模型的改进

图 2-1 从高级层面上分别展示了 SharePoint 2007 以及 SharePoint 2010 的编程模型。

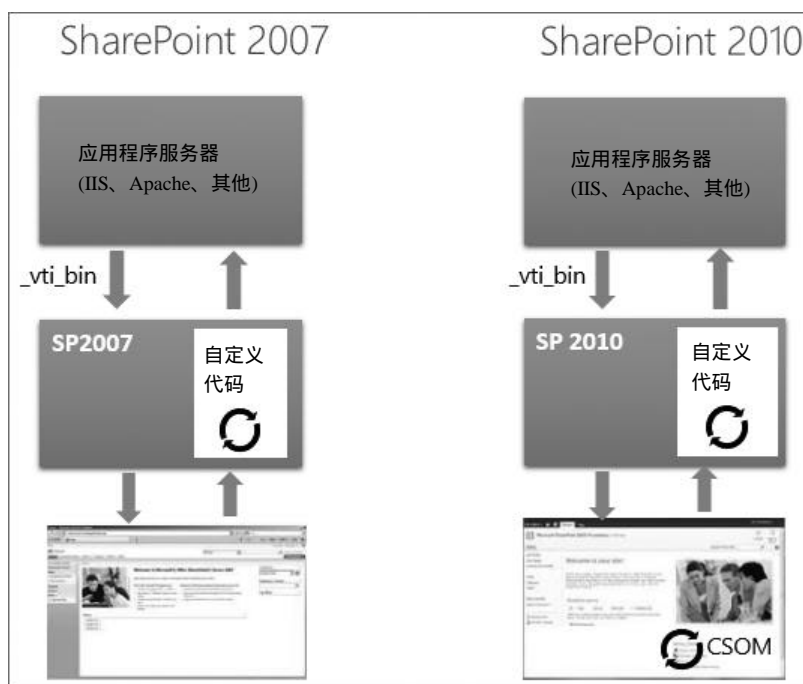


图 2-1

在 SharePoint 2007 中，自定义代码要么运行在 SharePoint 服务器端，要么托管在

自己的 Web 服务层(IIS、Apache，或者其他 Web 服务软件)上，并通过_vti_bin 文件夹(%COMMONPROGRAMFILES%\Microsoft Shared\web server extensions\12\ISAP)下 SharePoint 内置的服务远程调用 SharePoint 对象模型来进行工作。SharePoint 2007 有一个受限的客户端对象模型，而 Core.js 是其实现的核心(就好比 OWS.js 是 Windows SharePoint Services 2003 的客户端模型的核心一样)。该 JavaScript 文件被大部分 SharePoint 页面所引用，其包含必要的受限逻辑来处理各种客户端事件，开发人员可以开箱即用而不必修改 Core.js 文件。例如，Core.js 提供了一种挂钩程序来添加额外的菜单项以显示在上下文菜单(ECB)³中，下面是示例代码段：

```
function AddDocLibMenuItems(m, ctx)
{
    if (typeof(Custom_AddDocLibMenuItems) != "undefined")
    {
        if (Custom_AddDocLibMenuItems(m, ctx)) return;
    }
    ... // build the rest of OOTB ECB menu items
}
```

使用 SharePoint Designer(或者 Windows SharePoint Services 中的微软 Office FrontPage)，开发人员可以将附加的 Custom_AddDocLibMenuItems 函数的实现方法注入 SharePoint 页面，或者也可以使用内容编辑器 Web 部件(Content Editor Web Part，CEWP)将这段自定义的 JavaScript 方法注入 Web 部件页。无论哪种方式，Core.js 都会在运行时调用它的方法。这种扩展模式有一个问题，就是其本身是受限的而且不够灵活。比如，如果 Core.js 由于某种原因无法加载，那么自定义函数也将无法运行。

在 SharePoint 2010 中，客户端对象模型(Client Side Object Model，CSOM)进行了大幅改进与优化，以解决 SharePoint 早期版本中客户端编程所面临的困难。SharePoint 2010 CSOM 基本上可以看成一种 Windows Communication Foundation(WCF)服务，它称为 Client.svc 并附带了三种不同的代理服务以启用 Silverlight、JavaScript 以及 .NET 托管代码(C#或者 VB.NET)来对 SharePoint 进行远程调用。下面的代码段展示了开发人员如何使用支持 JavaScript(sp.js)的代理服务以便在 SharePoint 2010 中基于 CSOM 来执行异步查询：

```
ExecuteOrDelayUntilScriptLoaded(GetTitle, "sp.js");
// Code omitted for brevity
function GetTitle() {
    //Retrieve current client context
    context = SP.ClientContext.get_current();
    //Load the site
    site = context.get_web();
    context.load(site);
}
```

3 ECB——Edit Control Block，指的是列表项上下文菜单，也称为编辑控制块(ECB)菜单，以包括用于启动 Word Automation Services 文档转换作业的项目。


```
//Execute asynchronously with callbacks for successful or failed calls
context.executeQueryAsync(onQuerySucceeded, onQueryFailed);
}
```

2.4.2 SharePoint 2010 中 CSOM 面临的问题

在使用 SharePoint 2010 客户端对象模型时，开发人员主要面临着以下两个问题。

第一，SharePoint 2010 中的 CSOM 不支持直接调用 Client.svc，所有调用都必须通过其所支持的入口点来实现(如 sp.js)。因为代理服务仅对 .NET 托管代码、Silverlight 以及 JavaScript 客户端开放，所以那些不能解释并运行任何一种此类技术的平台或设备就不能远程调用 SharePoint。

第二，CSOM 仅仅涵盖 Microsoft.SharePoint.dll 中的接口并且其工作负荷在 SharePoint 中受到限制。为了克服这种局限性，开发人员别无选择，只能依靠 ListData.svc、基于内置 ASMX Web Service 的 jQuery，或者服务器端代码来访问更完整的 SharePoint 对象模型。

相比 JavaScript，开发人员更倾向于从托管代码(如 C #和 VB.NET)来使用 CSOM。这是因为他们可以面向强类型对象进行编程，也能够使用编译时类型的智能提示，而且更加易于调试。而使用 JavaScript 面向 CSOM 进行开发并不轻松。因为调用必须是异步的方式，而且没有编译时类型检查以及智能提示，这使得编写与调试 JavaScript 代码困难重重。

不过这些经验并不适用于 SharePoint 2013，下一节将会介绍在编写 SharePoint 2013 服务器端代码前需要三思而后行的缘由。

2.4.3 服务器端代码的问题

微软有一些原因来大力推进 SharePoint 2013 中的客户端编程以及远程访问的应用。但最本质的原因是，运行在场或沙盒解决方案中的服务器端代码是引发性能与安全性问题的根源。现在我们来暂时扮演一下魔鬼代言人的角色。假如你运营的软件公司构建了一个平台，为其他人发布他们的产品，并有可能赚钱，如果有人仅仅编写一个 Web 部件就能肆意掠夺你的成果，你会有何感想？假如别人说你的产品运行得很缓慢，但是实际上是由于其他人的代码或者配置错误，而导致你的产品看起来很糟，这时你又会有怎样的感受？

另一个问题是，SharePoint 服务器端对象模型学习起来很困难，而且它与其他非微软技术的平台跨技术互操作性极低。另外，在大多数情况下，SharePoint 也无法与跟上微软其他技术更新换代步伐。有多少次，作为开发人员的你遭遇到了 SharePoint 对象模型的紧密耦合问题——要求使用特定版本的 .NET Framework 框架或者 Windows Workflow Foundation？又有多少次，你期望那些从事 PHP 开发的兄弟可以在你的 SharePoint 项目上伸出援手？

对于不了解 SharePoint API 以及使用其他开发技术的开发人员，SharePoint 2013 提供的新版编程模型将为他们带来更高效的生产能力。现在，可以使用 SharePoint 2013 来构建兼容多种开发技术的松散耦合解决方案了。这意味着你可以选择自己喜欢的脚本语言

(PHP、JavaScript、LAMP⁴等)或者任意版本的 ASP.NET (ASP.NET Web Form、MVC 以及 Web Page)来为 SharePoint 构建应用程序,并允许远程调用 SharePoint 来完成各种任务,而这些工作在以前只能通过服务器端对象模型来完成。

除了上述体验和技术方式上的使用问题外,服务器端解决方案往往有很多部署工作要做,而且需要对服务器进行直接访问。这在本地部署方案中是适用的,但是并不适合云部署的方案,除非你在云中拥有自己的私有服务器。当然,可以通过沙盒解决方案来绕过这个问题,但是沙盒解决方案也有其自身的问题。首先,沙盒解决方案有很多操作上的限制,因为只有一部分服务器端 SharePoint API 可供沙盒解决方案调用。其次,沙盒解决方案的执行时间不能超过 30 秒,且其每天的资源消耗不能超过一定额度。另外,服务器端代码还不能调用或请求外部托管的 Web 服务。然而,开发人员可以使用基于客户端的解决方案(使用 JavaScript 或者 Silverlight)来调用外部服务,甚至使用例如 HTTP Post 的技术来进行跨域访问以传递消息。

另一个常见的开发难点在于升级 SharePoint 2013 之前版本的解决方案。WSP 版本管理、功能升级、程序集版本管理以及重定向更新等,但凡你想得到的,都面临着相同的更新问题。不得不说,在企业级项目中复杂解决方案及其功能的生命周期管理确实很难,以至于开发人员必须不断升级解决方案以满足持续变化的业务需求。所以,开发人员必须能够未雨绸缪并在代码架构上能够深谋远虑,只有这样,升级和卸载 SharePoint 早期版本的解决方案才不会变成一个痛苦而漫长的梦魇。SharePoint 2013 改变了这种窘况,它提供了一个强健的基础结构来支持升级和卸载 SharePoint 应用程序,并能确保应用程序升级失败时实现自动回滚,因此托管的 SharePoint 网站不会陷入不可用状态,这就不会给开发人员带来额外负担了。

既然你了解了老版本 SharePoint 所面临的问题,就该讨论 SharePoint 2013 做出了哪些改进了。

2.4.4 新的编程模型

如果你是从本章开头一直阅读到此,那么你一定特别想知道 SharePoint 2013 中新的编程模型与之前的版本相比有什么新颖之处。在老版本的 SharePoint 中编写客户端代码以及使用标准的 Web 开发技术不就行了吗?

图 2-2 简单描述了新的编程模型。

4 LAMP——指 Linux(操作系统)、ApacheHTTP 服务器、MySQL(数据库软件)和 PHP(有时也是指 Perl 或 Python)的第一个字母,一般用来建立 Web 服务器。虽然这些开放源代码程序本身并不是专门设计成同另几个程序一起工作的,但由于它们的免费和开源,这个组合开始流行(大多数 Linux 发行版本捆绑了这些软件)。当一起使用时,它们像一个具有活力的解决方案包。

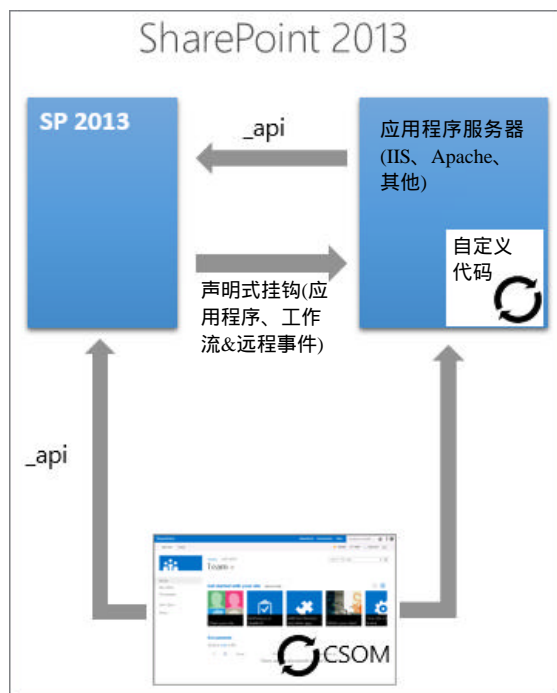


图 2-2

在 SharePoint 2013 中，SharePoint 场的服务器端代码基于诸如 SharePoint 应用程序、声明式工作流和远程事件等的声明式挂钩程序来运行，然后使用 CSOM 或者 REST 向 SharePoint 反馈数据。换句话说，新的编程模型强调的是客户端代码以及远程访问。根据部署方案(在本章开篇介绍过)，仍然可以使用沙盒解决方案以及场解决方案来部署服务器端代码；但是微软建议开发人员针对 SharePoint 2013 开发的首选方式是按照新的编程模型来构建应用程序。这是在向我们传递如此信息：“别再构建任何新的沙盒解决方案”以及“只有在不得已而为之的时候才构建场解决方案”(当然，前提是部署方案允许使用场解决方案)。

在 SharePoint 2013 中，_vti_bin/client.svc 现在可以用 _api 这一路径别名进行访问，以便与服务器端交互。当然，微软也赋予了 client.svc 完全兼容 OData 的服务。



提示：OData 是行业标准的开放式数据协议，可以对数据执行 CRUD(创建、读取、更新和删除)操作。CRUD 操作对应如 Get(检索)、Put(设置)、Update(更新)、Delete(删除)以及 Merge(合并)等标准的 HTTP 操作。OData 的执行基于一些流行标准，如 Atom(用于读取 Web 资源)协议以及 AtomPub(用于创建和更新资源)协议。OData 可用于许多非微软平台，如 iOS、Android，并用于知名的数据源，如 Netflix 公司的平台。要了解 OData 的更多信息，请查阅 <http://www.odata.org>。

如果你长久以来一直在使用 SharePoint，你可能还记得，在 SharePoint 2003 中，可以使用 Windows SharePoint Services Remote Procedure Call(RPC)协议来发起到 OWSSVR.dll 这一 ISAPI 扩展的 HTTP GET 请求(仅支持 HTTP GET)。例如，如果输入以下网址，将会在浏览器中输出指定 GUID 的列表结构并将该结构以 CAML⁵ 格式呈现出来。

```
http://Tailspintoys/sites/marketing/_vti_bin/owssvr.dll?
Cmd=ExportList&List=e6a9bb54-da25-102b-9a03-2db401e887ec
```

统一资源定位器(URL)在 SharePoint 中扮演着很重要的角色。长久以来，微软一直试图让用户能够更加便捷地获取 SharePoint 数据；其中的一些尝试在新版本的 SharePoint 中得以保留(比如，将列表导出到 Excel 中)。然而，由于设计模式中的种种限制以及这些设计模式中用到的一些声名狼藉的协议，这些尝试并不成功。OData(用作协议与启用程序)和 REST(用作设计模式)的使用基本保障了其他平台以及其他类型的设备通过 URL 和标准 HTTP 操作来访问 SharePoint 数据。



提示：SharePoint 2013 中仍然保留了 ListData.svc，以确保 SharePoint 2010 代码依靠 CSOM 平稳地迁移到 SharePoint 2013 中。这保证了能够向下兼容。然而，这意味着你需要使用 client.svc 来继续前进。

图 2-3 是一个架构图，它显示了本节讨论的 SharePoint 2013 中远程 API 的变化。

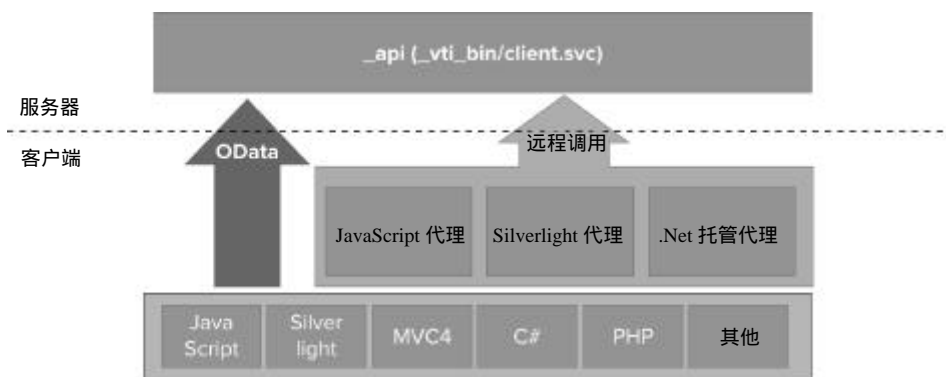


图 2-3

要使用 SharePoint 2013 的新版客户端 OData 服务，需要构建面向 client.svc 的 RESTful HTTP 请求，该请求将映射到 HTTP 操作并且与你想要使用的 CSOM API 相匹配。作为反馈，你会得到一个能轻易转换成嵌套对象的 Atom XML(默认类型)文件或者 JavaScript Object Notation(JSON)格式文件。下面这个例子告诉你如何使用 REST 来查询 Contacts 列表

5 协作应用程序标记语言(CAML)是一种基于 XML 的语言，用于在 Microsoft SharePoint Foundation 中定义在网站和列表中使用的字段和视图。

并且获取 ID 等于 1 的条目：

```
http://Tailspintoys.com/_api/web/lists/getByTitle('Contacts')/getItemByStringId('1')
```

下面的代码段展示了如何使用 JavaScript 和 REST 来删除 Contacts 列表中的条目。首先，在 JavaScript 里使用 CSOM，可以调用现成的 JavaScript 函数 `_spPageContextInfo.webServerRelativeUrl` 并传入 `id` 参数来拼接组成一个 REST URL。然后，调用 jQuery 的 `$.ajax` 函数来发送请求到 SharePoint Web 服务器以执行 HTTP DELETE 操作。最后，为成功和异常操作注册回调处理程序，该回调处理程序异步执行。

```
removeItem = function (id) {
    $.ajax(
        {
            url: _spPageContextInfo.webServerRelativeUrl +
                "/_api/web/lists/getByTitle('Contacts')/getItemByStringId('" +
                id + "')",
            type: "DELETE",
            headers: {
                "accept": "application/json",
                "X-RequestDigest": $("#__REQUESTDIGEST").val(),
                "IF-MATCH": "*"
            },
            success: function (data) {
                readAll();
            },
            error: function (err) {
                alert(JSON.stringify(err));
            }
        }
    );
}
```

请注意该代码段中 HTTP 的请求标头是如何创建的。这段代码使用了 `__REQUESTDIGEST` 这一标准名称来检索页面上表单摘要认证控件的值，并将该值赋予了 `X-RequestDigest` 标头。这一步非常重要，它能够确保修改内容数据库的 HTTP 请求顺利通过表单摘要认证，该认证是 SharePoint 自带的安全检测机制。

微软还为用户配置文件、工作流和搜索等功能服务以及在 SharePoint 2010 中仅能通过服务器端 API 来完成的许多其他工作负荷扩展了 CSOM。这还没结束，微软更进一步地改进了 CSOM，因此对于在拥有庞大用户群的生产环境中使用 SharePoint 这种情况，CSOM 就可以支持批量或同步操作，且不会对服务器性能造成重大负担。

到此为止，关于应用程序的抽象理论化介绍已经足够了，我们需要进一步探讨这些应用程序到底如何开发、部署以及使用。下一节我们将进入在 SharePoint 2013 中扩展使用应用程序的新世界。

2.5 应用程序模型

新的应用程序模型可以让你创建看起来像是 SharePoint 一部分的应用程序，但实际上它不是。如前所述，应用程序确实确实是独立于 SharePoint 服务器之外在客户端浏览器的上下文中运行的(通过客户端脚本、REST 以及 CSOM)，或者在云端或你自有的基础架构中托管的远程服务器中运行。

如果要开发和使用应用程序，需要面向两种平台：Office 2013 和 SharePoint 2013。你可以面向这两种平台开发应用程序；不过，本章以及本书接下来的内容都关注为 SharePoint 构建应用程序。可以为 SharePoint 2013 创建三种类型的应用程序：SharePoint 托管的应用程序、提供程序托管的应用程序以及 Azure 自动托管的应用程序。

2.5.1 SharePoint 托管的应用程序

SharePoint 托管的应用程序在所有的部署方案中均可用。此类应用程序部署在 SharePoint 中，但是它们的业务逻辑在客户端浏览器中执行。由于此类应用程序独立运行，因此它们无法与其他应用程序交互。

2.5.2 提供程序托管的应用程序

提供程序托管的应用程序在所有的部署方案中均可用。此类应用程序部署在 SharePoint 中，但是它们的大部分业务逻辑作为 SharePoint 外部的服务器端代码运行，比如，企业网络或云端中的其他 Web 服务器。



提示：除了 SharePoint 场中的服务器外，提供程序托管的应用程序需要运行在独立的服务器上。基于此，在项目的早期规划中就需要做硬件和软件层面的考量。

提供程序托管的应用程序(有时也称为开发人员托管的应用程序或者自托管应用程序)无法与其他应用程序交互。

2.5.3 Azure 自动托管的应用程序

Azure 自动托管的应用程序仅在 Office 365 部署方案中可用。此类应用程序与提供程序托管的应用程序相似，不同之处在于，其外部组件由 Windows Azure 网站以及可选的 Azure SQL 数据库组成，且由 Office 365 自动设置并提供。Azure 自动托管的应用程序也无法与其他应用程序交互。要启用 Azure 自动托管的应用程序模型，需要如图 2-4 所示的跨越 Office 365 和 Windows Azure 的端到端平台结构和工具模型(Visual Studio、SharePoint Designer、

Access 或者 Napa⁶)。



图 2-4



提示：本章提及的云端托管的应用程序，涵盖提供程序托管的应用程序以及 Azure 自动托管的应用程序。

对于开发人员或者 Office 365 用户，不需要为 Office 365 集成 Windows Azure 做任何设置工作。甚至都不需要 Windows Azure 账户。一旦注册并租用 Office 365 服务，就能使用已经集成并配置好的 Windows Azure。当最终用户决定安装 Azure 自动托管的应用程序时，Office 365 将自动部署并将该应用程序的副本以 Windows Azure 网站的形式提供出来，然后 Office 365 可以代表安装该应用程序的最终用户对其进行管理。这一新型应用程序让你能够使用极为诱人的自动安装方式来创建和分发自己的应用程序。

2.5.4 应用程序或解决方案——如何取舍

这个问题看起来简单，但要回答它出人意料地困难。开发人员开始为一个解决方案编写代码时，无法意识到随着时间的流逝该解决方案会变得多么臃肿。有一些因素会影响解决方案的迭代路径，如需求变更、项目周期的压力以及预算削减。为了克服这一前期的不确定性，始终必须进行臆测，并基于这些臆测条件来构建和交付解决方案。记住，交付才是主题。产品交付后，可以持续监控产品的使用方式以及用户的接受程度，再回头迭代改进解决方案。你需要自我提问来进行假设，然后比较这些问题的答案来挑选适合你的选项。

应用程序提供了三个层面的最高级别隔离：进程、用户以及内容层次结构。应用程序还为标准 Web 技术与基础架构选择提供了最具扩展性的方案，因为应用程序不会在安装

6 Napa 为 Visual Studio 富客户端提供了一种轻量级的、基于浏览器的工具，它是开始 Office 和 SharePoint 开发的一种不错的方式，而且不需要在计算机中安装任何内容。它通过 <http://dev.office.com/> 提供，支持构建针对 Office 和 SharePoint 的应用程序、能够深入到 Office 2013 应用程序(如 Excel)的应用、Office 的 Web 应用(例如，Excel 的 Web 应用)以及 SharePoint 的应用。这些应用都基于新的云应用模型(Cloud App Model)，其中 UI 和其他客户端逻辑都使用 Web 标准(例如，HTML、JavaScript、CSS)实现的，而后端逻辑会运行在服务器上，让开发人员可以自由地选择开发工具、语言和部署环境。

SharePoint 2013 的服务器上运行。

当面临在 SharePoint 应用程序和 SharePoint 解决方案之间做抉择时，有一些因素可以帮助你决定到底是做应用程序开发还是解决方案开发。

- ？ 服务器端对象模型——应用程序中不允许使用服务器端对象模型。要使用服务器端对象模型，需要使用沙盒解决方案的受限服务器端代码或者场解决方案的无限制服务器端代码。
- ？ 访问外部资源——在应用程序安装期间，可以对其赋予权限以访问托管该应用程序的 SharePoint 网站(SPWeb)范围之外的内容；例如，访问整个网站集的内容。很明显，只要有权限，所有类型的应用程序都可以访问 Web 服务以及那些不由 SharePoint 托管的资源。
- ？ 目标用户——这个很简单；比如，如果你要开发一个扩展功能来让场管理员或者 Web 应用程序管理员能够处理一些维护或清理工作，在应用程序中实现该功能就不是那么明智了。场解决方案也并非应用程序的一个合适的候选手段。
- ？ 复杂性——有一些任务要么用应用程序要么用解决方案来处理，并且用应用程序来处理这些任务比用解决方案来处理其复杂程度是一样的或者仅复杂一点而已。例如，应用程序或者沙盒解决方案都可用来部署自定义字段控件、网站栏、内容类型或者品牌化部件。在应用程序中完成这些任务可能比较困难。如果你对上述情况该选择哪种方式感到困惑，建议你还是选择应用程序来处理这些任务，要知道，应用程序才是未来，SharePoint 及许多其他产品都会进一步采用应用程序的模式。
- ？ 依赖性——如果要构建一个与其他自定义扩展功能高耦合的自定义扩展功能，并且该自定义扩展功能的安装前提是与其高耦合的扩展功能必须可用，这种情况下使用应用程序模式来开发该自定义扩展功能就不是一个好主意了。再次强调，应用程序是具有隔离性的并且是一个自我封闭型的容器，这种特性是为了保证其能够快捷安装、更新和卸载。同理，不要使用应用程序来承载其他应用程序或解决方案依赖的那些资源，因为这会阻止该应用程序的完全卸载或者该应用程序在卸载时会造成其他扩展功能的运行中断。
- ？ 适用功能范围——SharePoint 托管的应用程序不能是完整功能的应用程序，例如，案例管理应用或工资管理应用。SharePoint 托管的应用程序应该用来处理那些仅需一些主要资源的紧密关联的少量任务项。在之前的 SharePoint 版本中，可以创建以应付各种使用场景的大型业务解决方案，与此不同，SharePoint 2013 中的托管应用程序本质上是隔离的、事务型的，且创建它的目的是为了为了满足特定的业务需求。如果你需要完整功能的应用程序，要么使用 SharePoint 解决方案，要么使用云端托管的应用程序。

表 2-1 总结了在 SharePoint 解决方案(场或沙盒)和应用程序间抉择时需要考虑的决定因素。

表 2-1 SharePoint 解决方案与应用程序全对比

| 决 定 因 素 | 场解决方案 | 沙盒解决方案 | 应 用 程 序 |
|-----------------------|-------|--------|---------|
| 使用客户端 SharePoint API | | | |
| 使用服务器端 SharePoint API | | 受限 | × |
| 使用远程服务 | × | 受限 | |
| 应用程序身份验证(OAuth2) | × | × | |
| 友好的内部部署 | | | |
| 友好的托管部署 | × | | |
| 通过 Store 分发 | × | × | |
| 友好的安装/更新/卸载 | × | × | |

表 2-2 总结了在应用程序的类型抉择上应该考量的决定因素。

表 2-2 SharePoint 应用程序

| SharePoint 托管的应用程序 | 云端托管的应用程序 |
|-------------------------------------|------------------------|
| 具有具体业务需求目标的小型使用场景适用 | 小型或大型使用场景均适用 |
| 某些 Web 技术(HTML、JavaScript、jQuery 等) | 所有的 Web 技术；你可以从中选择 |
| 自动托管 | 可能需要你自己的基础架构 |
| 原生的多租户支持与隔离 | 可能需要你自己编写逻辑处理租约与隔离 |
| 原生的安装/更新/卸载功能 | 可能需要你自己编写逻辑来进行安装/更新/卸载 |

如果将目前为止了解的内容集中放置到一起，它看起来应该如图 2-5 一样。在托管或混合部署方案中选择封装方式取决于托管的提供程序或者 SharePoint 的部署方式，所以这部分内容没有体现在图 2-5 中。

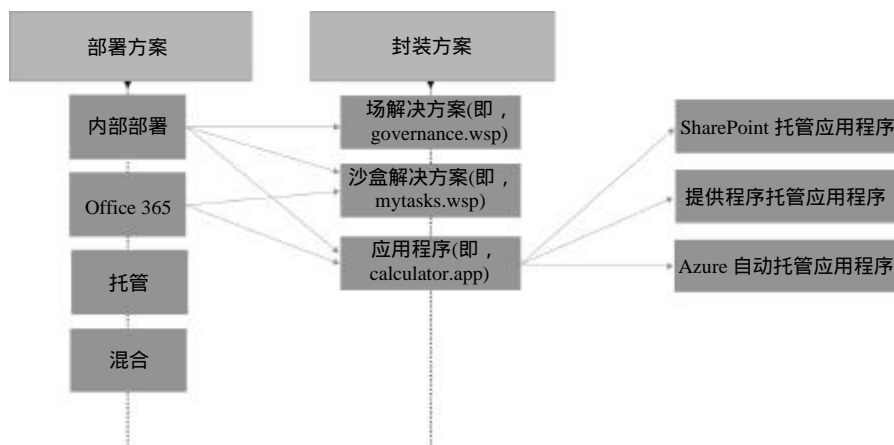


图 2-5

2.5.5 应用程序安全模型

除非为 SharePoint 网站开启了匿名访问，否则每一个访问 SharePoint 的请求都必须拥有其标识，也即我们通常所说的身份验证。身份验证必不可少，这是因为如果 SharePoint 不能识别访问者的身份，它就不能赋予访问者适当的权限来访问其资源。



提示：SharePoint 从未囊括身份验证和身份管理业务，在 SharePoint 2013 中这种情况依然如故。作为依赖方(Relying Party, RP)，SharePoint 依赖于如活动目录一样的目录服务或者如 Google 和 Facebook 一样的身份标识提供程序(Identity Provider, IdP)，并且 SharePoint 还依赖于如 IIS、Azure 访问控制服务(Access Control Service, ACS)和活动目录联合服务(Active Directory Federation Service, ADFS)这些服务来向外部提供其身份验证业务。SharePoint 同样支持如 SAML、WS-Trust 和 WS-Federation 这些身份验证标准和规范。

由于编程模型中的变化，SharePoint 2013 的授权通道必须面向两种不同类型的身份标识开放：用户标识和应用程序标识。后者是 SharePoint 2013 中一种全新的标识。

我们来举个例子，以便理解 SharePoint 2013 应用程序标识的必要性：想象一下，如果有个远程应用程序需要访问 SharePoint 网站中的一些产品图片并将它们推送到由市场部运维的一个推特账号中。显然，该应用程序首先需要登录 SharePoint，然后再访问和向推特推送那些图片。要完成这项任务，一个办法就是给这个应用程序赋予一组凭据(用户名和密码)，这样该应用程序就能以你或者其他人的账号登录 SharePoint 以完成任务。众所周知，这并不是一个好办法。首先，SharePoint 里的所有事物都与用户标识联系在一起，并且 SharePoint 授权通道必须区分清楚哪些行为是推特应用程序做的，哪些行为是你做的——SharePoint 就是这样工作的。其次，你能做的推特应用程序都能做，比如，删除内容或者以你的名义在 CEO 的博客中发表评论。你不会想冒这个风险的。再者，如果你决定不再使用推特应用程序了呢？你的账户凭据怎么办？如果你有很多像推特应用程序一样的应用程序呢？你又如何管理这么多应用程序的账户凭据？

SharePoint 使用存储在内容数据库中的 SharePoint 安全组和访问控制列表(Access Control List, ACL)配置条目来描述用户标识的行迹。不过，SharePoint 选取了不同的途径来进行身份验证以及对应用程序标识授权。

首先来看身份验证。在 SharePoint 2013 中应用程序是头等重要的；因此，它需要自己的标识。应用程序标识是由 Azure 访问控制服务(ACS)通过称为 OAuth 的广泛使用的互联网安全协议提供给 SharePoint 的。OAuth 2.0 是 OAuth 的第二个版本，它需要 HTTPS 协议支持。

新的架构使 SharePoint 2013 用户能够在不共享用户凭据的情况下对应用程序赋权以使其能够以该用户的名义运行。要启用该功能，需要颁发一些不同类型的安全令牌，并且需

要在用户、SharePoint、ACS 和应用程序本身之间来回通信，所有这些都是为了确保应用程序能够被 SharePoint 进行透明的身份验证。



提示：OAuth 安全令牌服务(如 ACS)并不会颁发登录令牌。登录令牌由身份验证提供程序(IdP)颁发，这就是对于 SharePoint ACS 仅仅提供并将一直作为应用程序身份验证提供程序的原因。

当 SharePoint 2013 接收到访问请求时，进程会立即开始检查该请求以识别其是否包含代表用户标识的登录令牌。如果该请求包含用户标识，SharePoint 会假定该请求由经过身份验证的用户而非应用程序发起。随后 SharePoint 会检查该访问请求的目标 URL 以判断该 URL 是否包括标准的 SharePoint 网站或者与应用程序关联的子网站(这称为 AppWeb)。如果访问请求的目标是标准网站，SharePoint 2013 会沿用与 SharePoint 2010 相同的处理机制来进行典型的授权处理。但如果访问请求的目标是 AppWeb，SharePoint 2013 会同时用用户标识与应用程序标识来初始化上下文。

当访问请求不包含登录令牌时，SharePoint 2013 便可以区分出该请求并非用户发起的。此种情况下，SharePoint 会查找 OAuth 令牌来确认远程应用程序(提供程序托管)的身份。一旦 SharePoint 2013 获取到该安全令牌，它将用应用程序标识以及可选的用户标识来初始化上下文。

现在，我们来看看授权机制。

当 SharePoint 对提供程序托管的应用程序进行了身份验证并且正确的上下文创建好后，SharePoint 就可以判定该应用程序在调用上下文中有权限执行哪些操作。值得一提的是，SharePoint 2013 并没有使用 ACS 或者 OAuth 协议以任何方式对应用程序进行跟踪或授予权限。相反，就像处理用户权限一样，SharePoint 仍依赖其自身的内部内容数据库来跟踪授权。

每一个应用程序都有个 manifest.xml 文件，开发人员可以用该文件中的 AppPermissionRequests 元素来定义应用程序需要访问的资源列表。下列代码段展现了在提供程序托管的应用程序中使用该元素的例子。

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
  <AppPermissionRequest Scope="http://sharepoint/content/sitecollection"
    Right="Read" />
  <AppPermissionRequest
    Scope="http://sharepoint/content/sitecollection/web/list"
    Right="Write">
    <Property Name="BaseTemplateId" Value="101" />
  </AppPermissionRequest>
  <AppPermissionRequest Scope="http://sharepoint/userprofilestore/feed"
    Right="Post" />
  <AppPermissionRequest Scope="http://exchange/calendars"
```

```
Right="Schedule"/>

</AppPermissionRequests>
```

注意上面代码段中突出显示的部分。应用程序权限需要启用 app-only 策略,这意味着只有应用程序而不是当前用户在请求所需的权限。如果 app-only 策略没有启用,那么应用程序和当前用户都需要必要的权限才能完成诸如访问整个网站集或写入列表的任务。返回的结果将是一段同时包含应用程序和用户标识的上下文。

关于 app-only 策略的一个重要方面是它可以提升应用程序的权限,这样应用程序就能够比当前用户处理更多的任务。如此一来,在没有当前用户的情况下,应用程序也能够回调 SharePoint 以访问应用程序 Web(与一个用程序关联的 SharePoint 网站)及其父 Web(例如,托管该应用程序的 SharePoint 网站)。当用户安装了带有 AppPermissionRequest 条目的应用程序时,该用户必须在安装应用程序时对 manifest.xml 文件中定义和应用程序请求的内容赋予权限。

如前所述,当创建需要与 SharePoint 通信并由提供程序托管的应用程序时,有几种类型的安全令牌需要来回通信,如上下文令牌和 OAuth 令牌。值得庆幸的是,Visual Studio 会自动添加一个名为 TokenHelper.cs 的类,以帮助你访问和使用这些安全令牌。

在没有租用 Office 365 的情况下,不允许在内部部署环境中使用 ACS。相应地,这意味着不能使用 OAuth 令牌。应用程序需要使用另外一种由服务器间(S2S)配置生成的安全令牌。更多详细信息可以参见 MSDN 上的产品文档 [http://msdn.microsoft.com/en-us/library/fp179901\(v=office.15\).aspx](http://msdn.microsoft.com/en-us/library/fp179901(v=office.15).aspx)。

不能仅仅因为应用程序在其自有域环境中运行(为了防止跨网站脚本攻击)以及这些应用程序是以 JavaScript 编写的,就断定它们是安全的。作为开发人员,你仍然需要面对如何设计应用程序以避免安全漏洞和敏感信息泄露的问题。下面是 SharePoint 安全方面需要考虑的问题。

- ❑ 不能针对特定列表或网站级别对 SharePoint 托管的应用程序授权。换句话说,如果 SharePoint 托管的应用程序被赋予了一个列表的写入权限,该应用程序同样拥有了对其他列表的写入权限。
- ❑ 用户为提供程序托管的应用程序授予权限后,由于赋权操作是一次性的,因此如果应用程序的逻辑和代码在赋权后进行了修改,SharePoint 无法检测到这些改动。
- ❑ 当应用程序使用其 OAuth 令牌处理任务时,相同页面上的另一个应用程序可以使用这个令牌以应用程序标识、用户标识的名义或同时以这两者的名义来执行操作。

黑客还可能通过不安全的通信通道(HTTP)来操纵 OAuth 令牌。

综上所述,你就会意识到在应用程序开发方面需要考虑许多安全层面的问题。为了确保应用程序足够安全以及保护敏感信息,在设计时就需要考虑安全问题并且使用 HTTPS 来保障通信通道的安全。

2.6 远程事件

随着 SharePoint 逐渐成为许多组织主要的协作核心平台，与外部系统(意指在 SharePoint 外部的系统)进行双向集成在过去几年中已成为主流需求。

为在 SharePoint 中展现外部数据，可使用与 SharePoint 紧密集成的诸如 Reporting Services、PerformancePoint 和 Excel 这样的商务智能技术，以及诸如搜索和业务连接服务(BCS)这样的功能。反过来看，SharePoint Web 服务和 CSOM 也提供了一组选项用以在外部系统中展现 SharePoint 数据，尽管这些方式是受限的。然而，SharePoint 并没有提供一套强健的通知基础设施，以用来在 SharePoint 事件触发或者外部系统展现的数据实际已经发生变化的情况下通知外部系统。

看看这样的情景：Tailspin 玩具公司管理团队宣布了一项新规定，现场销售团队必须对在 CRM 系统中创建的销售机会以最快速度进行销售可能性的反馈。过去，销售人员必须登录 CRM 系统来查看销售机会。但是，如果有一套合适的通知系统，一旦有新的销售机会产生，销售人员就可以通过 SharePoint 发送的邮件获得通知。销售机会的联系信息会附加在通知里，这样，现场的销售代表就可以即刻呼叫该顾客并提供高级经理期望他们提供的个性化服务。

在 SharePoint 的早期版本中，要构建一个满足类似于 Tailspin 玩具公司需求的有效解决方案并不容易，只能通过一些复杂的、完全信任的场解决方案来实现，这些解决方案要用到一些自定义事件接收器以及供 CRM 系统调用的 Web 服务。完成该任务的一种方法是使用 SharePoint 2010 的可插入工作流服务来创建自定义工作流活动，以实现远程事件接收器的功能。图 2-6 显示了如何在 SharePoint 2010 中使用 callExternalMethodActivity1 和 handleExternalEventActivity1 活动来启用插入式工作流服务。要了解可插入工作流服务的更多信息，可查阅《SharePoint 2010 开发高级教程》一书的第 13 章。

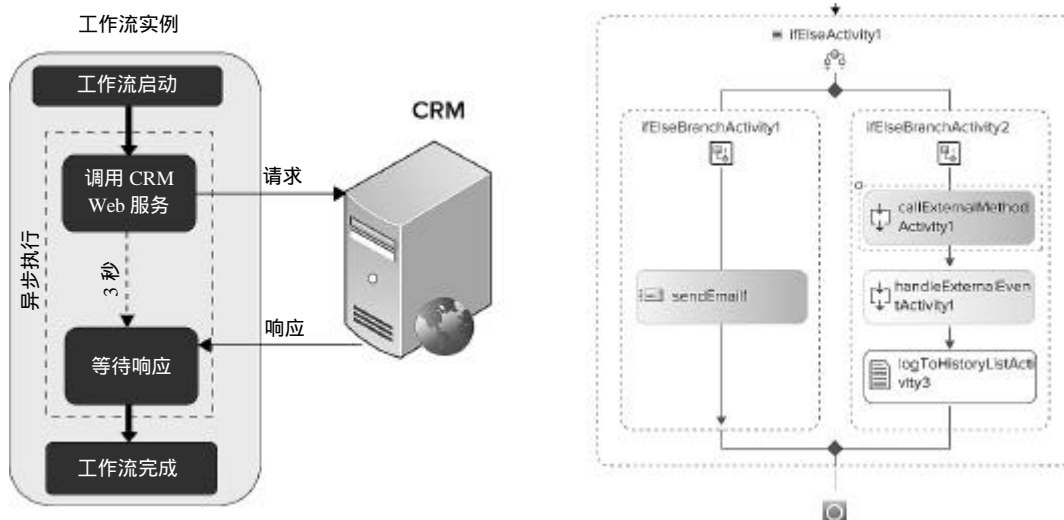


图 2-6

鉴于在 Office 365 和托管式部署模式中不允许使用完全信任的场解决方案，可插入工作流服务可查阅的资料又少得可怜，且开发人员并没有完全认可插入工作流服务，微软不得不在 SharePoint 2013 中提供内部语义和基础架构层面的支持以启用远程事件。

为应对跨平台通知带来的挑战，微软引入了三种全新的选项：Reporting Services 数据警报、外部列表事件和远程事件接收器。现在我们快速浏览一下每一种选项。

2.6.1 Reporting Services 2012 数据警告

此功能在 SQL Server 2012 中引入，并且仅在 SharePoint 集成模式下可用，Reporting Services 的这项新功能使得在那些数据源内存储了凭据的报表中配置警报成为可能。当报表中的基础数据发生更改时，将会通知收件人。因为在数据警报中可以指定收件人的电子邮件地址(参见图 2-7)，所以收件人可以是启用了电子邮件的文档库，在该文档库中需要注册一个 SPEmailReceiver 类型的事件接收器来侦听传入的邮件，然后解析该邮件包含的发件人、收件人、主题、正文和发送属性等信息并基于这些信息执行相关操作。

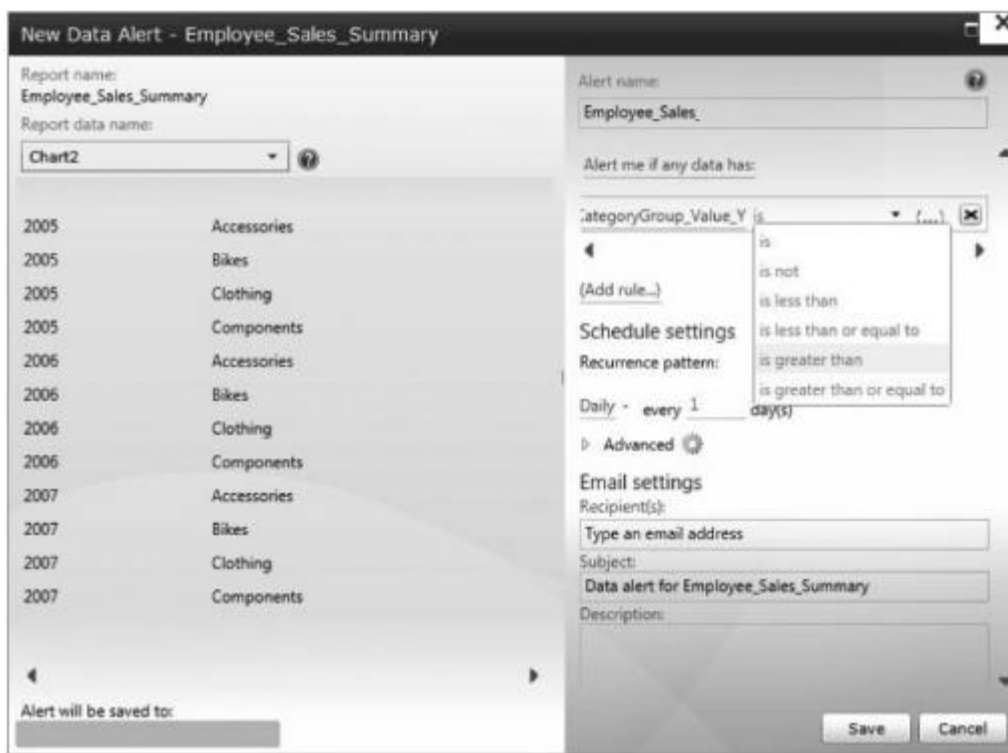


图 2-7

2.6.2 外部列表事件

SharePoint 2013 业务连接服务中的外部列表支持事件。BCS 对象模型新增了两种模式(订阅和退订)以支持外部列表通知。可以对 ItemAdded、ItemUpdated 和 ItemDeleted 事件进行配置以支持订阅功能。外部列表会在第 13 章和第 14 章中进行介绍。

2.6.3 远程事件接收器

这是 SharePoint 2013 的全新功能,开发人员可以使用与 SharePoint 2010 中注册本地事件接收器相类似的方法在 SharePoint 2013 中注册远程事件接收器。唯一的主要区别是,开发人员需要提供一个 Web 服务的 URL(而不是程序集)和该 Web 服务的类名,该 Web 服务在注册事件接收器时会调用。当注册的事件发生时,SharePoint 会将事件属性发送给 Web 服务,然后等待事件接收器反馈结果信息。



提示：可将 SharePoint 2013 的远程事件接收器看成一个提供程序托管的应用程序。不同于远程应用程序,远程事件接收器会为你提供一个 Web 服务,也不同于 default.aspx 页面,你可以得到一个能够回调 SharePoint 的服务(*.svc)。远程事件接收器使用的是与 CSOM、REST 和 OAuth 申请这些服务相同的核心远程通信技术。

下列代码段显示了作为 SharePoint 托管应用程序的一部分来部署的远程事件接收器中为指定列表而定义的 XML 元素。注意 URL 和类型节点处 Web 服务与事件类型如何指定。

```
<?xml version="1.0" encoding="utf-8"?>
<Elements xmlns="http://schemas.microsoft.com/sharepoint/">
  <Receivers ListTemplateId="10000">
    <Receiver>
      <Name>AnnouncementsReceiverItemAdded</Name>
      <Type>ItemAdded</Type>
      <SequenceNumber>10000</SequenceNumber>
      <Url> http://tailspintoys.com/AnnouncementsReceiver.svc</Url>
    </Receiver>
  </Receivers>
</Elements>
```

该 Web 服务只是一个公共类,它实现了 IRemoteEventService 接口和处理事件发生前的-ing 事件(此处指的是 ItemAdding)的 ProcessEvent()方法,以及处理事件发生后的-ed 事件(此处指的是 ItemAdded)的 ProcessOneWayEvent()方法。

```
public class AnnouncementsReceiver : IRemoteEventService
{
    public SPRemoteEventResult ProcessEvent(RemoteEventProperties
properties)
    {
        SPRemoteEventResult result = new SPRemoteEventResult();
        switch (properties.EventType)
        {
            case RemoteEventType.ItemAdding:
                //Code to handle ItemAdding
                break;
```

```
        case RemoteEventType.ItemDeleting:
            //Code Omitted for brevity
        break;
    }
    return result;
}
public void ProcessOneWayEvent(RemoteEventProperties properties)
{
    if (properties.EventType == RemoteEventType.ItemAdded)
    {
        //Code Omitted for brevity
    }
}
```

2.6.4 远程事件中的安全模型

当涉及远程事件和系统间的相互协作和通知能力时，首要关注点永远是安全性。图 2-8 显示了在 SharePoint 2013 中使用远程事件的情况下各功能角色如何协同工作。

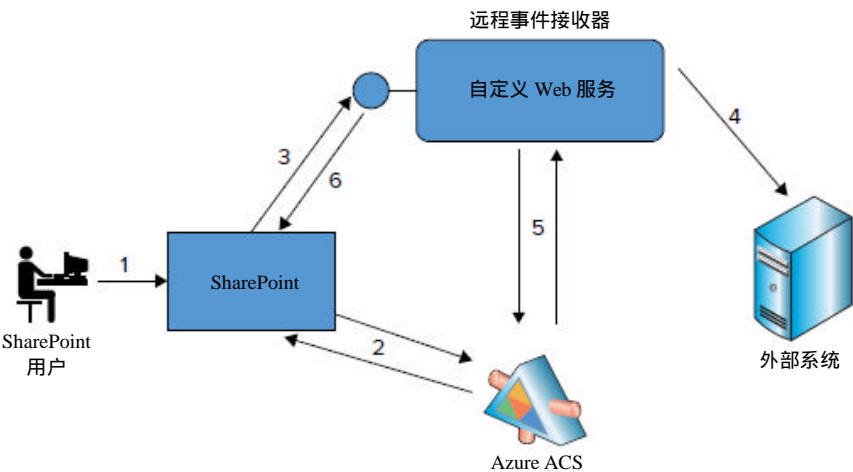


图 2-8

以下是图 2-8 中所示的基本步骤。

- (1) 用户在 SharePoint 中触发了一个事件(如 ItemDeleting)。
- (2) SharePoint 调用 ACS 来获取代表当前用户标识的安全令牌。
- (3) SharePoint 调用已注册的事件接收器(Web 服务)并将该安全令牌发送给 Web 服务。Web 服务校验该令牌并授权调用。
- (4) 该 Web 服务将在外部系统中执行某项操作，如更新业务线(LOB)数据。
- (5) 该 Web 服务从 ACS 请求安全令牌以回调 SharePoint。
- (6) 该 Web 服务使用安全令牌进行身份验证并回调 SharePoint 来执行任务。

下列代码段显示了远程事件接收器中的 Web 服务如何从 ACS 里获取上下文令牌以及如何构建 SharePoint 上下文和远程客户端上下文来处理 SharePoint 中的任务：


```

HttpRequestMessageProperty requestPro =
    (HttpRequestMessageProperty)OperationContext.
    Current.IncomingMessageProperties[HttpRequestMessageProperty.Name];
string ctxTokenString = requestPro.Headers["X-SP-AccessToken"];
SharePointctxToken ctxToken = TokenHelper.ReadAndValidatectxToken
(ctxTokenString,requestPro.Headers[HttpRequestHeader.Host]);
Uri spUrl = new Uri(properties.ItemEventProperties.WebUrl);
string accessToken = TokenHelper.GetAccessToken(ctxToken,
    spUrl.Authority).AccessToken;
ClientContext clientContext =
    TokenHelper.GetClientContextWithAccessToken(spUrl.ToString(),
    accessToken)

```

可以使用 TokenHelper 类来创建安全令牌和进行 SharePoint 身份验证,就像本章前述内容介绍的提供程序托管的应用程序一样。

在 Office 365 部署环境中,可以使用 OAuth 为 Web 服务创建安全令牌以便能够回调 SharePoint,创建过程中也需要 ACS 参与。在内部部署环境中,应该使用服务器间配置来创建安全令牌。要获取更多信息,可以参见 [http://msdn.microsoft.com/en-us/library/fp179901\(v=office.15\).aspx](http://msdn.microsoft.com/en-us/library/fp179901(v=office.15).aspx) 处的产品文档。

下一节将介绍工作流有哪些新的变化。

2.7 工作流

从本章开始以来,已经对 Office 365 和基于 Windows Azure 之间的紧密集成的服务(如 ACS)作了介绍。到目前为止,你也了解了在应用程序上下文和远程事件中这些服务的集成方式。

基于相同的后台通道,现在可以便捷地编写和上传声明式工作流到 Office 365 中,并且该工作流在 Windows Azure 工作流执行托管主机中运行,完全独立在 SharePoint 服务器场之外。然后,Windows Azure 中运行的工作流可以使用与基于云端的应用程序中应用到的相同的核心远程访问技术,如 CSOM、REST 和 OAuth。

按照本章惯例,我们先来看看早期版本 SharePoint 中的工作流有哪些问题。

2.7.1 SharePoint 2010 中工作流的问题

你应该还记得,工作流最早在 SharePoint 2007 中以 Workflow Foundation 3.0 的形式引入到 SharePoint 平台。从架构层面看,SharePoint 2010 中的工作流没有太多变化,因为整个平台还是使用了基于 .NET Framework 3.5 SP1 的 Workflow Foundation。

我们来看看在以下 4 种主要应用场景中使用 SharePoint 2010 工作流的情况。

- ？ 场景 1——高级用户在 SharePoint Designer 2010 中构建了一个声明式工作流并将其部署到 SharePoint 2010 中。声明式工作流不包含任何自定义代码并且本质上是严格顺序流(不支持状态机工作流)。

- ？ 场景 2——高级用户在 SharePoint Designer 2010 中构建了一个声明式工作流并且聘用了一名开发人员在 Visual Studio 2010 中进行编程以扩展该工作流。之所以需要这样做，可能是下列 SharePoint Designer Workflows 中的某项限制造成的。
 - 实现自定义业务逻辑
 - 调用 SharePoint 接口
 - 调用外部系统(Web 服务、订阅推送服务和数据库)
- ？ 场景 3——开发人员在 Visual Studio 2010 中构建了自定义活动并且将这些活动封装成操作以便在 SharePoint Designer 2010 中使用。之后管理工作流业务逻辑的高级用户就可以在其声明式工作流中使用这些操作了。
- ？ 场景 4——开发人员在 Visual Studio 2010 中构建了程式工作流并且将其封装以便部署。在这种场景下，所有的开发工作都在 Visual Studio 2010 中完成。

虽然上述场景涵盖并满足了大多数客户的需求，但工作流技术还是有一些限制的。这些限制包括以下几方面。

- ？ 扩展性——与大多数沿用 SharePoint 2010 服务应用程序模型以获得更好扩展性和支持分租模式的工作任务不同，在 SharePoint 2010 中工作流并非一种实际意义上的服务。所以，客户在每个 SharePoint 2010 场中只能使用一个由所有 Web 应用程序和分租请求共享的工作流执行托管主机。
- ？ 稳定性与性能——SharePoint 2010 工作流会频繁地在服务器场不同层级间切换运行。有缺陷的工作流或运行中的实例过多将大大影响服务器场的性能及其可用性。
- ？ 需要高级权限的上下文——工作流在网站应用程序池的安全上下文中以超级用户的权限运行。这在许多案例中是不可取的，因为要启动工作流就需要能够访问安全上下文的用户账号。例如，在实际发生的工作流中，其创建者和修改者字段将始终显示成系统账户。要了解关于工作流安全上下文的更多信息，请在 <http://technet.microsoft.com/library/dd365119.aspx> 处参阅 TechNet 的文章。
- ？ 内部部署环境——鉴于 SharePoint 2010 中工作流需要超级用户的权限才能运行，所以沙盒解决方案不能用来托管程式工作流。要使用沙盒解决方案，唯一的选项就是部署可以在 SharePoint Designer 2010 中使用的工作流操作。
- ？ 灵活性——因为 SharePoint 2010 与指定版本的 .NET Framework 和 Windows Workflow Foundation 紧密耦合，所以工作流开发人员就不能使用微软发布的最新工作流技术。这又是一个产品与指定技术的特定版本紧密耦合所带来的苦果。

2.7.2 SharePoint 2013 中的工作流架构

与其前身不同，SharePoint 2013 采用了不同的方式来实现工作流。在 SharePoint 2013 中，工作流(技术)现在可以被视为真正的服务了。这意味着 SharePoint 工作流不再运行在 SharePoint 服务器中；而由 Windows Azure Workflow 在独立的服务器中处理其执行过程。



提示：颇为有趣的是，微软在介绍执行 SharePoint 2013 工作流的基础结构时，使用的是 Windows Azure Workflow 场这一词语。

Windows Workflow Foundation 4.0 与 .NET Framework 4.5 是使用新架构的基础。这两者都在其早期版本基础上进行了大幅度的重新设计。图 2-9 显示了 SharePoint 2013 中工作流平台的技术架构。

SharePoint 2013 工作流的重点是以声明的形式构建工作流。这与 SharePoint 2010 不同，在 SharePoint 2010 中，高级用户使用 SharePoint Designer 构建声明式工作流，而开发人员使用 Visual Studio 构建编程式工作流。

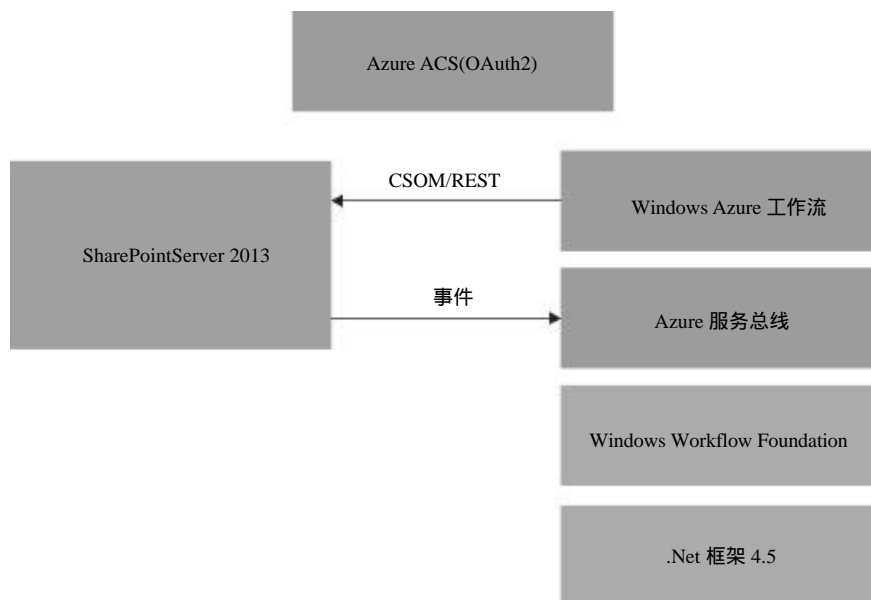


图 2-9

现在所有开箱即用的 SharePoint 2013 工作流都是声明式的，无论使用 SharePoint Designer 2013 还是使用 Visual Studio 2013 开发的新工作流都将是声明式的。

由于 SharePoint 2013 工作流全部以声明方式实现，因此可使用沙盒解决方案来将工作流作为一个整体进行部署。这就与在 SharePoint 2010 中使用沙盒解决方案仅能部署工作流操作不一样了。

微软建议使用 SharePoint 2013 工作流模型来构建新的工作流并且将自定义代码放到 Web 服务中。完全从头开始构建编程式工作流仍然是可行的，但这样做并不可取，原因有二。

？ 鉴于平台的新功能，没必要构建编程式工作流。稍后将介绍这些新特性。

- ? SharePoint 不能托管程式工作流, 程式工作流必须作为非 SharePoint 2103 工作流部署到 Windows Azure Workflow 中。然后开发人员需要面对实现工作流与 SharePoint 2013 之间通信通道的任务。



提示：可将 Windows Azure Workflow 看成 Azure 自动托管的应用程序，它提供了托管和执行 SharePoint 2013 工作流的“服务”。SharePoint 通知该应用程序执行工作流，该应用程序执行工作流并且将执行结果反馈给 SharePoint。在这两个产品之间有一套消息传递的基础架构，这样它们就能与彼此通信并交换信息。该消息传递的基础架构就是 Windows Azure 服务总线。

依照向后兼容的惯例，现有的 SharePoint 2010 工作流可以完全平稳地迁移到 SharePoint 2013 中。不仅如此，SharePoint 2013 工作流还可调用 SharePoint 2010 工作流。这是受欢迎的功能，因为它使得客户可在 SharePoint 2013 中继续使用他们在 SharePoint 2010 中投入精力开发的工作流。

新架构的另一个吸引力是，在内部部署和 Office 365 部署环境中，Windows Azure Workflow 和服务总线都可以使用。基于此，就可以构建自有的工作流开发环境。下一节将介绍完成该项任务的步骤。

2.7.3 构建自有工作流开发环境

Windows Azure Workflow 可以作为一个独立的产品进行下载，它使得客户可以在其内部部署环境安装和配置工作流服务器场。得益于此，就可以用相同的方式构建自有的工作流开发环境。

需要牢记的一点是，如果没有经过一些特殊处理，不能将 SharePoint 2013 与 Windows Azure Workflow 安装到域控制器上。因此，至少需要一台用作域控制器的独立服务器和一台承载 SharePoint 2013、Windows Azure Workflow 及服务总线的服务器。

总体来说，构建自有的工作流开发环境需要以下步骤。

- (1) 创建一个工作流服务账户(例如，Tailspintoys\wrkflowSvc)。
- (2) 在 SQL Server 中将该服务账户添加到 securityadmin 和 dbcreator 服务器角色里。该账户同时也是本地管理员组的成员。
- (3) 在 SQL Server 中启用 TCP/IP，这是 Windows Azure 服务总线所需的。
- (4) 使用第(1)步中创建的工作流服务账户登录到 SharePoint 服务器。此步骤很重要，它能确保安装向导是在工作流服务账户的安全上下文中运行的。
- (5) 在 <http://www.microsoft.com/web/downloads/platform.aspx> 处下载并安装 Microsoft Web Platform Installer。使用这个应用程序来下载并安装 Windows Azure Workflow 及服务总线。需要搜索 Workflow 1.0 Beta 版本。
- (6) 在这台机器上继续保持该工作流服务账户的登录状态，运行安装程序并选择 Create

a New Farm | Using Default Settings(Recommended), 如图 2-10 所示。

选择 Allow Workflow Management over HTTP on this Computer ; 否则, 就必须设置 HTTPS, 这对于以开发为目的的安装未免过于小题大做了。

当指定工作流服务账户时, 请使用完全限定的 UPN 格式(wrkflowSvc@Tailspintoys.com), 而非向导中显示的默认值(wrkflowSvc@Tailspintoys)或者 NetBios 名称(tp\wrkflowSvc)。

在 Certificate Generation Key 和 Confirm Certificate Generation Key 处, 输入 pass@word1 或者选择输入其他口令。



图 2-10

如果进展顺利, 你会看见如图 2-11 所示的一个总结页面。

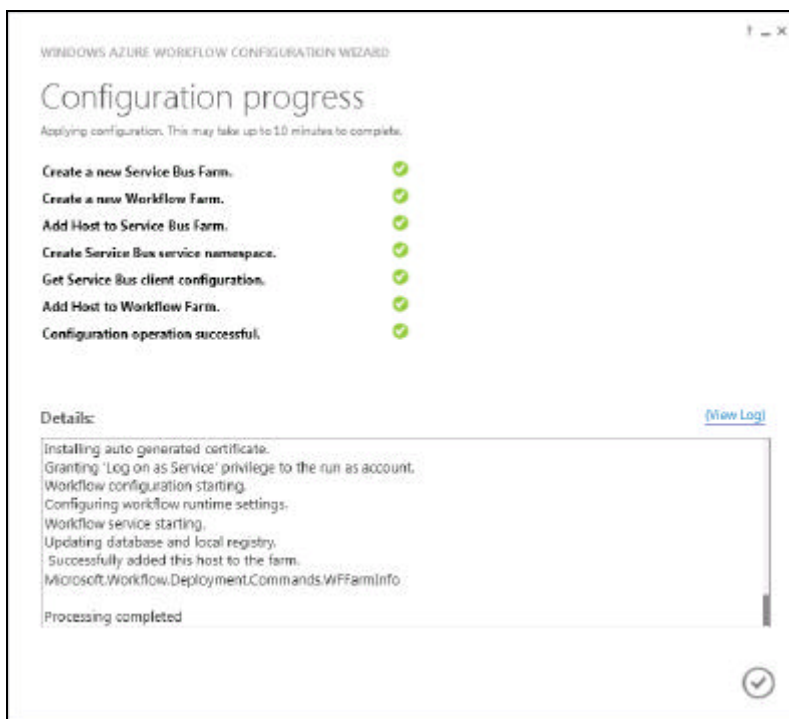


图 2-11

(7) 注销并使用服务器场账户登录。

(8) 执行下面的 PowerShell 脚本将 SharePoint 服务器场与工作流服务器场匹配起来。该脚本中以下参数非常重要。

-SPSite 为工作流服务器场指定网站路径，以便与 SharePoint 通信。每个 Web 应用程序只能指定一个网站路径(例子中使用 `http://Tailspintoys.com/sites/workflow`，但可以使用任何适合特定需求的值)。

-WorkflowHostUri 为 SharePoint 指定网站路径，以便与工作流服务器场通信(将 `mydevserv` 替换成你自己的值)。

-AllowOAuthHttp 将 HTTP 协议设置为允许 OAuth 调用，这对于开发目的合适，但不要将 HTTP 用于身份验证。

```
Register-SPWorkflowService -SPSite "http://Tailspintoys.com/sites/workflow" -WorkflowHostUri "http://mydevserv:12291" -AllowOAuthHttp
```

(9) 在 User Profile Service Application 中的 Configure Active Directory Synchronization 处进行配置。Windows Azure Workflow 会从 User Profile Service Application 里查找用户的 UserPrincipalName(UPN)属性，以验证并确保该用户拥有启动工作流的权限。

到此，Windows Azure Workflow 和服务总线就成功安装了，并且 SharePoint 2013 服务器场与工作流服务器场也成功匹配好了。现在，可以打开 SharePoint Designer 来创建新的 SharePoint 2013 工作流了，如图 2-12 所示。

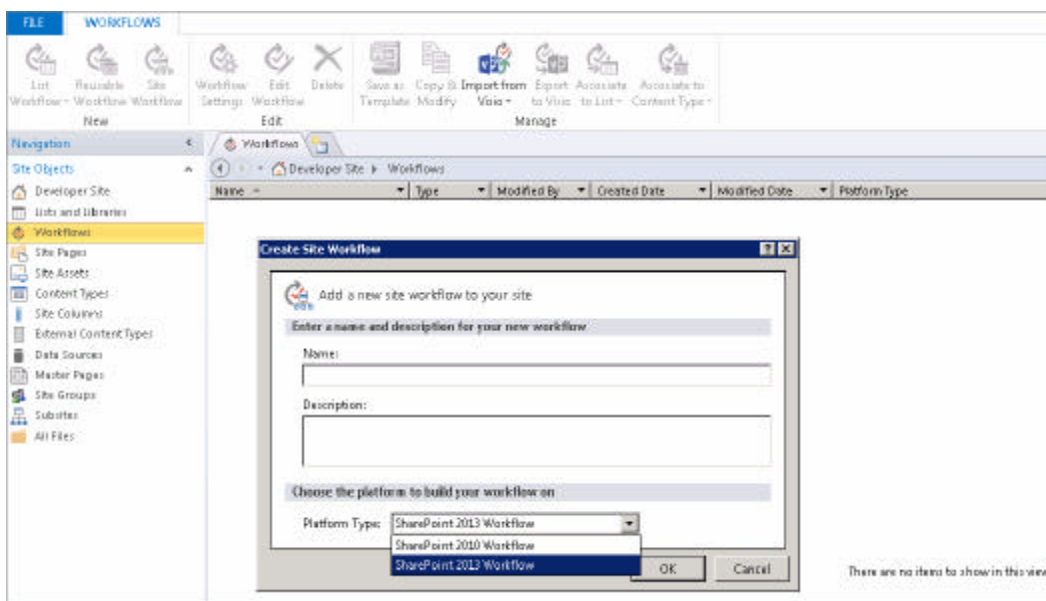


图 2-12

2.7.4 使用 SharePoint Designer 2013 设计工作流

SharePoint Designer 2013 为开发人员 and 高级用户提供了两种形式的编辑与定制体验。

- ？ 基于文本的设计器——这是设计器画布的改进版本。比如，在工作流设计画布中选择多个对象并将其复制和粘贴到同一工作流或其他工作流中。颇为有趣的是，该功能虽然简单，却是 SharePoint Designer 用户最为需要的功能。
- ？ 可视化设计器——这是全新的设计画布，仅在 Visio 2013 与 SharePoint Designer 2013 一起安装的情况下才能使用。

可以单击 Ribbon 菜单上的 Views 按钮来切换设计器。图 2-13 显示了新的可视化设计器。

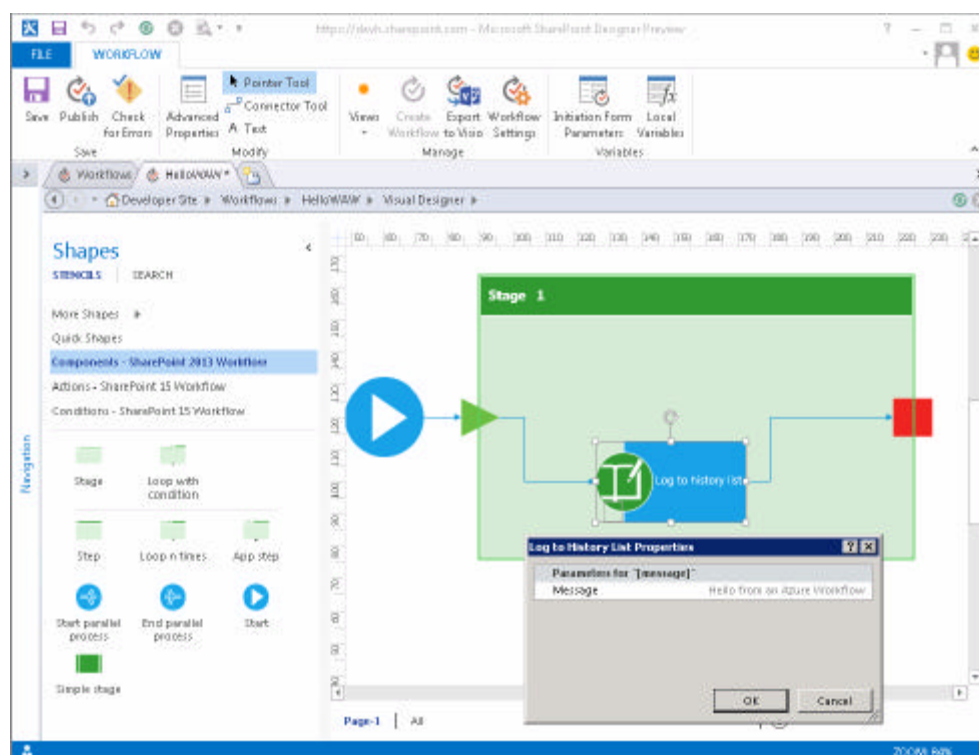


图 2-13

注意图 2-13 中左侧面板里的 Visio 图形。还有，注意可以使用可视化设计器直接修改工作流操作(如 Log to History List 操作)的属性。可以通过选择图形然后单击图形左下角显示出来的操作标签来设置所有的属性。单击操作标签的 Properties 选项，条件和操作的属性窗格会显示在可视化编辑器中，同样，在基于文本的设计器中也这样操作。

除了设计画布有所改进之外，SharePoint Designer 2013 现在也能够支持新的顶层容器和操作。下列是比较重要的几个。

- ？ 循环——一个新的顶层容器；顾名思义，循环是将操作、条件和步骤组合在一起的容器，这些在流程中都将重复执行。循环中的重复执行次数可以是固定的(n 次)或动态的(基于一个条件或表达式)。
- ？ 阶段——如前所述，在早期版本的 SharePoint 中声明式工作流原生仅支持众所周知的顺序模式。这些工作流具有一个起始点和终结点，一切都按顺序运行并且一切都向前进行。在 SharePoint Designer 2013 中，阶段是一个新的顶层容器，它能将操作、

条件或步骤组合在一起并让其按照既定顺序执行。当流程运行到阶段结尾时(也称为阀门), 将通过 Go To 操作来决定转向到该工作流的其他阶段。



提示：可在 Go To 转向阀门中插入判断条件来将流程从一个阶段转向到另一个阶段，直到逻辑判定该工作流已经结束。这意味着 SharePoint 2013 中的声明式工作流不再必须是顺序流，并且也可以作为模型状态机工作流来使用了。

- ？ 调用 SharePoint 2010 列表和网站工作流——SharePoint Designer 2013 允许 SharePoint 2013 工作流调用 SharePoint 2010 列表和网站工作流。可以使用称为 Start a List Workflow 和 Start a Site Workflow 的两个新候选操作来完成该任务。对于那些已经开发了 SharePoint 2010 工作流同时想要将其保留并移植到 SharePoint 2013 中来的客户，这非常重要，这些客户可以在稍后再将这些工作流完全移植到 SharePoint 2013 版本。另外，还可以用该技术调用未集成到 SharePoint 2013 中的工作流活动。
- ？ 调用 HTTP Web 服务——现在在声明式工作流中已经支持调用 SOAP、WCF 和兼容 OData 的服务了。可以使用称为 Call HTTP Web Service 的新操作来完成该任务。面向远程服务的调用在工作流运行的环境(Windows Azure Workflow)中发起，其返回的数据会存储在工作流变量中，这些数据可以被该工作流中的其他操作访问。在 SharePoint 2013 工作流中嵌入自定义逻辑和代码的建议方式是创建自定义 Web 服务，并且使用新的 Call HTTP Web Service 操作来调用该服务。

图 2-14 展示了 SharePoint Designer 2013 工作流中新的阶段概念，以及如何按条件从 Wait for Approval 阶段转向(跳转)到 Approved 或者 Rejected 阶段。

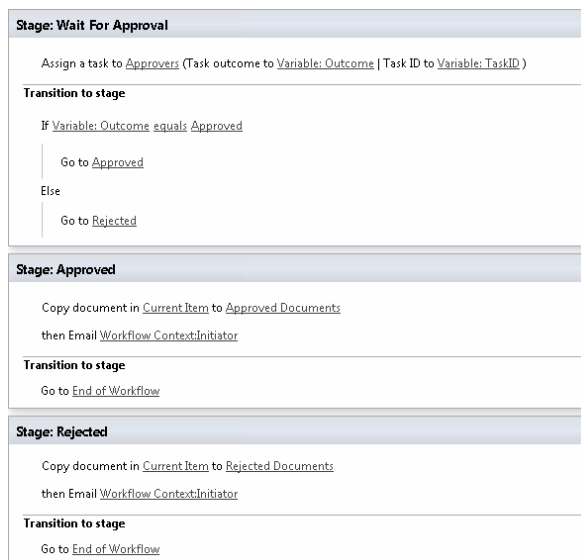


图 2-14

另一种使用 SharePoint 2013 工作流的有趣方式是应用程序，这将在下一节中介绍。

2.7.5 在应用程序中使用工作流

SharePoint 2013 中新的工作流模型使得一种有意思的开发方案类型成为可能。现在可以在应用程序中将工作流用作业务逻辑的中间层托管主机。

类似于之前介绍过的插入式工作流服务，对于保护应用程序中业务逻辑的知识产权(IP)或者对于开发人员想要将需要长时间运行的业务逻辑从应用程序剥离出来从而为应用程序减轻负荷，该技术都非常有用。比如，如果应用程序要与 CRM 系统的销售机会功能集成，可以将该业务逻辑封装在工作流中并且在 Windows Azure Workflow 执行主机中运行。



提示：需要使用 Visual Studio 2012 来构建使用工作流的应用程序。SharePoint Designer 2013 不支持使用工作流的应用程序设计。

应用程序可以使用新的工作流 JavaScript 对象模型(JavaScript Object Model, JSOM)来启动声明式工作流并与其交互。JSOM 使得应用程序可以将工作流明确地部署到应用程序 Web 和父 Web 中。

2.8 企业内容管理

企业内容管理(Enterprise Content Management, ECM)最早在 SharePoint 2007 时引入到平台中，当时通过添加两种网站模板(记录中心和文档中心)以及诸如签入、签出、版本、信息管理策略、保留和许多其他与 SharePoint 内容相关的功能来实现。

SharePoint 2010 通过将基础的 ECM 功能封装成特性从而拓展了这些功能。SharePoint 2010 中还为 ECM 引入了许多其他关键和合规的功能，比如，就地记录管理、文档 ID 服务、文档集和内容管理器。

在 SharePoint 2013 中，尽管微软引入了诸如 eDiscovery 和网站邮箱这样的一些新 ECM 特性和增强功能，但对于开发人员有两项功能尤为重要：网站策略与托管元数据。

先来看网站策略。

2.8.1 网站策略

信息管理策略是一组定义了特定行为或对 SharePoint 中保存的内容进行限制的规则集合。比如，对许多如 HR 一样的部门来说，对敏感信息进行审计是常见的需求。在 SharePoint 网站的运营周期内，许多诸如“谁在过去 10 天内修改了 HR 网站的权限？”或者“有人把工资单文档库中的内容移动到其他网站了吗？”的审计问题都可能会被问及。

在 SharePoint 2010 中，仅能为内容类型或者网站集创建策略。SharePoint 2013 使用了新的策略集。浏览 Site Settings | Site Collection Administration，有一个称为网站策略的新链

接，这使我们能够控制网站的整个生命周期。

图 2-15 显示了网站策略中的一些可用选项。

图 2-15

如图 2-15 所示，可以选择网站过期的方式以及过期时如何处理。例如，在网站创建 7 个月后自动删除该网站，并且在删除前运行工作流来处理一些自定义业务逻辑。为方便起见，如果定义了策略的网站集是内容类型集线器，就可以将该网站集的策略推送到所有订阅内容类型发布的网站集中。



提示：在 SharePoint 2013 中与自助式网站创建结合使用时，网站策略为网站生命周期管理提供了强有力的管理机制。用户申请新建网站时可以选择网站策略，并且该网站策略将基于其自身逻辑自动强制执行。

在网站集级别定义了网站策略后，网站所有者可以在子网站的设置页面导航到 Site Closure and Deletion 页面，然后选择网站策略。也可以使用 SharePoint 2013 中改进的 CSOM 接口以编程方式完成该操作。

下一节会介绍 SharePoint 2013 中托管元数据的改进。

2.8.2 托管元数据

对比 SharePoint 2010，托管元数据在 SharePoint 2013 中扮演了更为重要的角色。不过，从架构层面看，其核心功能还一样。

要在网站中使用托管元数据，还是需要托管元数据服务(Managed Meta-data Service, MMS)应用程序和称为术语库的工具。该工具允许你每次使用一个托管元数据服务应用程序的实例。术语按照类别来分组(高级容器)，每个分组中还将按照类别分成术语集。术语层次结构与联合内容类型一起存储在 MMS 应用程序数据库中。如果启用社会性标签，MMS 应用程序也可以使用由用户配置文件服务应用程序创建的社会性标签数据库。

如同 SharePoint 2010 一样, SharePoint 2013 的托管元数据组可以是本地的或者全局的。全局组可以被任意连接到相同 MMS 应用程序实例的网站集使用。本地组虽然存储在 MMS 应用程序数据库中,但它仅对指定网站集可用。SharePoint 2013 中的不同之处在于,通过指定要使用本地组的网站集的 URL,能够使本地组对其他网站集可用(在只读模式中)。

图 2-16 显示出在 SharePoint 2013 改进了的术语库中进行跨网站集的术语访问。

图 2-16

在 SharePoint 2010 中,用户可以在术语层次结构中重用术语。重用这些术语后,可以在其源术语集和引用术语集中对其进行更新和删除。在任意节点进行的更新都会同时更新整个术语层次结构,但是删除操作稍有不同。删除源术语集中的重用术语,并不会删除引用术语集中的该重用术语。因此,重用术语应看成一种指针,但实际上它们并非指针——这让人有些混淆。

SharePoint 2013 中保留了重用术语且其功能与 SharePoint 2010 中一致。SharePoint 2013 为术语引入了新的操作:术语固定。固定术语类似于重用术语,只是固定术语是只读的且不能在引用术语集中对其进行修改(更新或删除)。此外,如果在源术语集中删除了固定术语,也会在引用术语集中删除该固定术语。现在,你拥有了真正的指针。



提示:跨网站集的术语基于固定术语而非重用术语。

另一项重要改进是术语的自定义属性。在 SharePoint 2010 中,术语的属性包仅能使用分类服务器端 API 进行访问。现在,在 SharePoint 2013 中,可以通过浏览器与 CSOM API 来使远程客户端具备使用术语和术语集自定义属性的能力。自定义属性要么是共享的要么是

是本地的。那些在所有重用和固定的术语实例中都可用的自定义属性称为共享自定义属性。只有源术语集中术语的本地属性才可以访问。

对于双语用户和业务有一个重大消息，那就是无须安装语言包即可在 MMS 应用程序中添加工作语言，如图 2-17 所示。

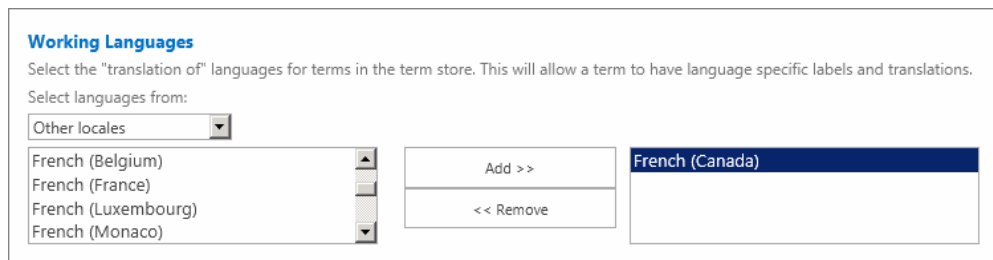


图 2-17

对于那些想要使用多语言分类但其内容主要是以英语编辑和使用的客户，这绝对是一项受欢迎的改变。对于那些不愿意仅仅为了多语言分类而安装和管理多语言包的开发人员，这项改变也非常便利。

在 MMS 应用程序中添加新的工作语言后，每一个术语集都将出现一个新的选项卡，它可以让你选择三种术语翻译选项。这些选项分别是：Machine Translate(机器翻译)，这由 SharePoint 2013 中新的机器翻译服务应用程序驱动；Create Translation Package(创建翻译包) 这术语集导出成 XLIFF 包⁷以便进行专业翻译，或者 Upload Translation(上传翻译)，这基本上将翻译的 XLIFF 包导入 MMS 应用程序。

我们已经简要地了解了 ECM 的新功能；下一项探讨任务是 Web 内容管理。

2.9 Web 内容管理

在 SharePoint 2013 中，微软特别重视 Web 内容管理(Web Content Management, WCM) 方面的工作内容。得益于产品中新的搜索和托管元数据的改进，SharePoint 2013 采用了两种不同的方式来处理内容发布：结构化发布模型与动态发布模型。

现在更详细地介绍这两种模型。

7 XLIFF——即 XML Localization Interchange File Format，XLIFF 是由软件开发商、本地化服务提供商、本地化工具提供商等团体共同倡议和设计，由 OASIS 标准组织发布的用于本地化数据交换的格式标准。它基于 XML 技术制定软件资源文件格式的转换规格，其目的在于提高软件的本地化作业效率，制定可扩展的多语言本地化数据交换的规范，允许任何软件开发商根据该规范创建单一数据交换格式的文件，这些单一数据交换格式的文件能够向任何本地化服务商提交，并且能够被本地化服务商易于理解和有效处理。

2.9.1 结构化发布模型

这是 SharePoint 2010 中发布网站的工作方式。内容作者可以独立地创建内容页面并使其在发布网站中可用。比如,如果需要为某产品创建详细信息的页面,在 <http://www.tailspintoys.com/products> 处打开产品网站并基于预定义模板(页面布局)创建一个发布页面来显示该产品。该页面签入且发布后,它将作为一个详细信息页面,访问者可以在其浏览器中输入以下 URL 来浏览该页面:<http://www.tailspintoys.com/products/pages/foo.aspx>。

这种方式对于需要在 SharePoint 中承载的内容很有用且其具有静态特性。编辑好内容页面后,需要以某种方式将它们汇总到另一个页面上,通常称为汇总页。可以使用内容查询 Web 部件(Content by Query Web Part ,CBQ)或者其他自定义汇总技术来以主/从方式显示产品。

在结构化模型中,使用内容部署可以将发布内容移动和本地化到变体网站。或者,可以使用内容部署将内容进行跨越当前编辑和发布的服务器场以及跨越内部网站、外部网站和互联网网站之间的网络边界的移动。

2.9.2 动态发布模型

这是 SharePoint 2013 新增的,汇总和详细页面可以从索引内容中自动生成。例如,大多数组织放在非 SharePoint 的外部系统中的产品目录,可以通过搜索对其进行索引并自动包括到 SharePoint 2013 发布网站中。

得益于新的托管元数据功能,可以使用更简洁且更易于搜索的地址来访问产品页面,如 <http://www.tailspintoys.com/foo.aspx>。

在动态发布中,可以使用 SharePoint 2013 中称为跨网站发布的新功能来让其他网站可以使用某网站的内容。接下来我们看看背后的动态发布模型是如何工作的。

2.9.3 分类驱动导航

SharePoint 2013 中的导航基础结构使用分类来生成发布页面的 SEO 友好的 URL 与路径。如果查看发布网站中的导航设置,就可以知道有两种方式来设计网站导航:结构化和托管。

结构化导航在 SharePoint 2010 就已经出现了。而托管导航是全新的且由网站分类所驱动。这是一个重要理念,因为从现在起,你无须更改网站的基础结构即可依照业务操作来提取网站导航。这同样使得网站所有者能够通过修改术语集来便捷地重新组织导航。

图 2-18 显示了 SharePoint 2013 发布网站导航设置里全新的托管导航选项。

不只是导航可以基于分类。下一节将介绍 SharePoint 2013 WCM 中的分类驱动页面。

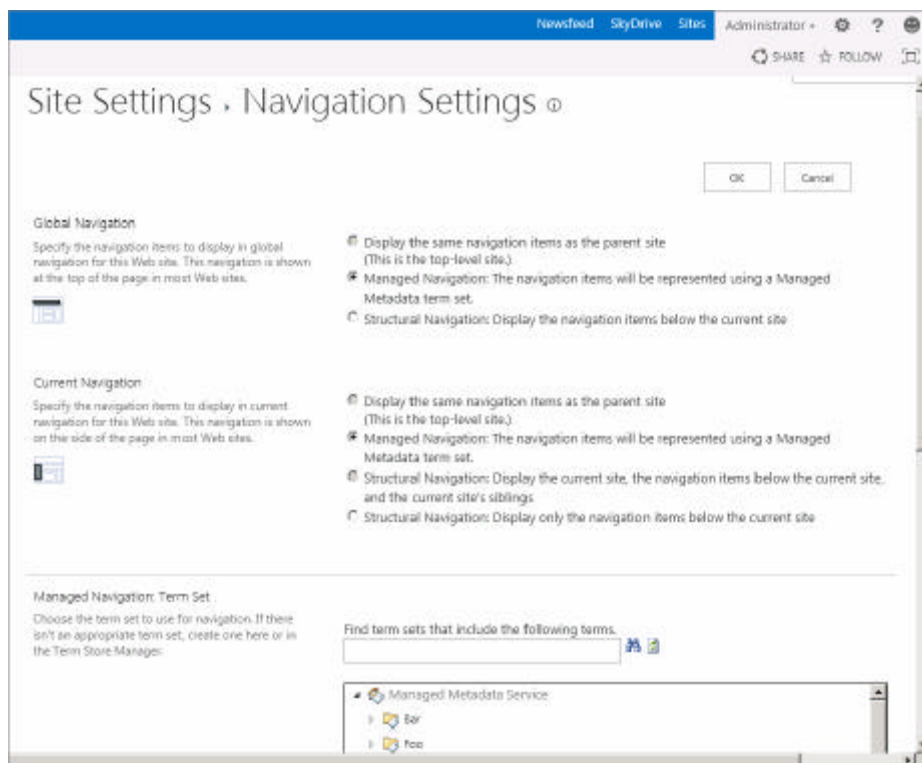


图 2-18

2.9.4 术语驱动发布页面

无论使用结构化还是动态模型在 SharePoint 2013 中创建发布页面，SharePoint 都会自动在网站导航术语集中添加一个指向该新页面的新术语。此外，SharePoint 还将为该术语生成一个主页，就像社会性标签配置文件一样。

术语主页就是一个简洁的页面布局(.aspx)，用来显示发布页面的内容。像往常一样，开发人员和设计人员有办法定制模板来满足特定的展现需求了。

通过术语库中称为 Intended Use(预期目的)的新选项卡，可以选择加入哪些术语集来参与分类驱动导航并且稍后可以进一步定制。选择此选项启用 Navigation(导航)和 Term-Driven Pages(术语驱动的面页)选项卡，它使你能够定制友好的 URL、SEO 选项、目标页面设置和许多其他功能。

2.9.5 跨网站发布

就算你从事 SharePoint 编程只有很短的一段时间，你也应该知道要跳出网站集边界和跨多个网站集来聚合内容并不是一项简单的任务。有一些方式和技术可以用来进行跨网站集聚合，但它们都需要额外的开发工作且均受其自身的某些限制。

SharePoint 2013 允许开发人员对列表和文档库中的内容进行设置以供其他网站集使用。实现起来很简单且只需要几个高级操作步骤。

- (1) 创建一个包含网站栏和内容类型的列表(或文档库)。



提示 :只有网站栏自动包含托管属性并且无需额外配置工作就能在搜索索引中显示。如果使用列表栏,则必须创建托管属性并将这些托管属性映射到列表栏的已爬网属性上。切记,跨网站发布高度依赖内容索引。

(2) 将该列表标记为目录(Catalog)。在列表设置页面有一项新的设置可以完成此任务。这样列表中的内容就可以通过 MMS 应用程序来让其他网站集使用。一个目录至少有一个主键(最多 5 个)来对列表项进行唯一标识。目录还有一栏标记为目录导航(Catalog Navigation)。目标网站集需要将此栏显示在它们自己的导航层次结构中。这个目录导航栏是一个托管元数据字段且与术语集关联,这称为标签术语集。

- (3) 使用与 2.8 节介绍过的相同技术来将目录的标签术语集共享给其他目标网站集。

(4) 运行完全爬网,并确保加入到目录(也即目录导航)中的栏能够通过目标网站集的托管属性顺利查询到。

(5) 在目标网站集中,将导航设定为托管导航(Managed Navigation)。更多信息请参考 2.9.3 节。

- (6) 连接到目标网站集,并在 Site Settings | Manage Catalog Connections 处连接目录。

图 2-19 显示了一个目标网站集中产品目录实现的例子。

在汇总页上单击一项,该页面将从搜索索引中即刻动态生成。这个过程中没有定时器作业参与。

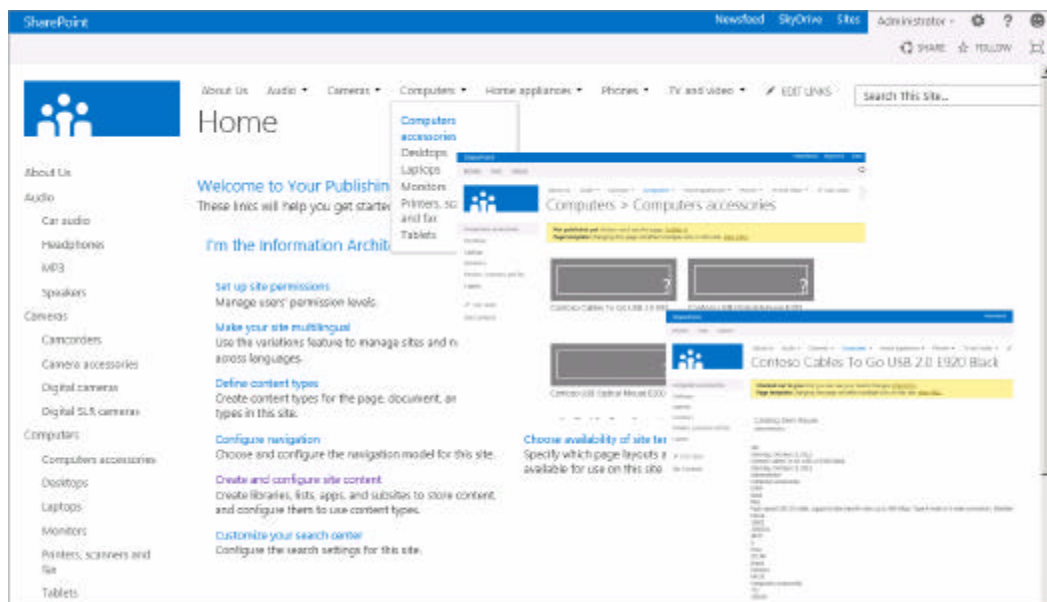


图 2-19

目录标记方式在 SharePoint 2013 中是如此重要以至于微软决定推出一个称作产品目录(Product Catalog)的开箱即用模板。该模板自带一个标记列表,用作称为产品(Products)的目录。这是为了给你提供一条起跑线,它能引导你通向创建你自己企业产品目录系统的大道。



提示：当你将引入到 ECM、WCM 和搜索的新功能组合在一起时，希望你能明了有极大的可能性来实现有意义的开发方案，如跨网站集导航、发布和部署。

在进入下一节之前，有两件重要的事情需要特别说明。

第一，WCM 中新的跨网站发布功能并不是为了替代传统的内容部署。在许多情况下，你仍然应该选择内容部署来进行跨网站发布。更多信息请参阅 [http://msdn.microsoft.com/en-us/library/jj163225\(v=office.15\).aspx](http://msdn.microsoft.com/en-us/library/jj163225(v=office.15).aspx) 处的产品文档。

第二，结构化发布与动态发布以及这两种模型中使用的技术不是互斥的。为了应对复杂的发布需求，它们可以共存或者混合使用。例如，可以将跨网站发布与变体网站联合使用，以便从共享的创作网站集中启用创作多语言网站。

2.9.6 主机名网站集

SharePoint 2007 支持将 Web 应用程序扩展到多个区域并为每个区域赋予一个唯一主机名(主机头)。由于 SharePoint 对于在单个服务器场中托管的 Web 应用程序有数量上的限制，因此 SharePoint 2010 中引入了主机名网站集(Host Name Site Collection, HNSC)来处理这个扩展性问题。问题是 SharePoint 2010 中的主机名网站集必须在默认(Default)区域中且不能使用可选择的访问映射。此外，每个网站集仅支持一个主机名。

SharePoint 2013 对 HNSC 进行了升级，现在每个网站集可以支持不限数量的主机名并且可以在 Web 应用程序级别将每一个主机名映射到一个区域。不过依然需要扩展 Web 应用程序，且每个 Web 应用程序只能支持 5 个区域：Default(默认)、Intranet(内网)、Internet(互联网)、Extranet(外网)以及 Custom(自定义)。但是，不同之处在于，SharePoint 2013 在不同区域中启用主机名的方式。



提示：SharePoint 2013 对软件边界和限制进行了修改。更多信息请参阅 [http://technet.microsoft.com/en-us/library/cc262787\(v=office.15\)](http://technet.microsoft.com/en-us/library/cc262787(v=office.15)) 处的产品文档。

下列代码段在 URL 为 <http://foo> 的 Web 应用程序中创建一个 URL 为 <http://www.bar.com> 的 HNSC。该 Web 应用程序有两个区域：Default 和 Internet。

在该段代码中随后还为该新的 HNSC 添加了额外的路径；该 Web 应用程序 Default 区域的路径 <http://foo.bar.com> 以及 Internet 区域的路径 <https://foo.bar.com>。


```
#Create a new HNSC
New-SPSite "http://www.bar.com" -HostHeaderWebApplication "http://foo" -Name
"Bar Portal" -Description "Bar Portal" -OwnerAlias "Tailspintoys\administrator"
-language 1033 -Template "STS#0"

# Get a reference to the new HNSC
$site = Get-SPSite 'http://www.bar.com'

# Add an alternative URL and map to Default zone
Set-SPSiteURL -Identity $site -Url http://foo.bar.com -Zone 0

# Add an alternative URL and map to Internet zone
Set-SPSiteURL -Identity $site -Url https://foo.bar.com -Zone 2
```

当创建可选择名称时，可以指定使用哪个区域(0 = Default 区域, 2 = Internet 区域)。如果使用下列代码列出为该新 HNSC 创建的所有区域，会得到如图 2-20 所显示的结果：

```
Get-SPSiteUrl -Identity http://www.bar.com
```

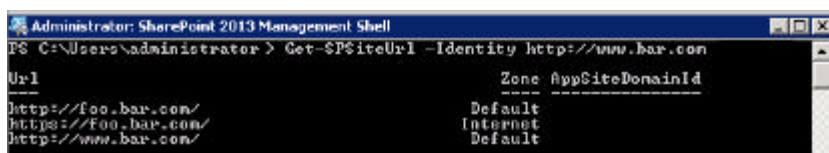


图 2-20

如果该 Web 应用程序的互联网区域支持匿名访问，那么 HNSC 中另外的 URL 也一样支持匿名访问。

2.9.7 多语言功能

如果你生活在像加拿大一样的多语种国家，你大概知道让用户能够切换到另一种语言来呈现他们的内容是多么重要。如果从本章开篇一直阅读到此，就能知道在 SharePoint 2013 中有一些对于跨网站发布和托管元数据的多语言支持功能，但其实还有更多提供多语言支持的功能。

在 SharePoint 中，变体一直是满足多语言需求的主要功能。变体基于下列 4 项原理来将源内容复制成变体标签(目标)：

- ? URL 路径
- ? 语言重定向
- ? 翻译
- ? 内容部署

变体仍然受限于一个网站集；不过，SharePoint 2013 中变体的执行过程变得更快且更加可靠了。你会得到一个更小的导出包，并且有一个副本列表允许快速启用和停用副本内容。这意味着内容部署不再是一个孤注一掷的可怕过程；而且，可以选择一次性复制整个列表或一个或多个变体标签。

类似于托管元数据的术语，SharePoint 2013 中的变体支持发送网站内容到机器翻译服务应用程序的功能。或者，可以导出或导入由第三方以行业标准的 XLIFF 格式翻译的网站内容。在导出内容中可以包括所有变体标签、一个页面或者仅仅一个文档。为了满足开发自有的自定义翻译解决方案的需求，机器翻译服务对象模型与 Word 自动化服务对象模型类似，并且就像 CSOM 和 REST 一样在服务器端可用。



提示：当机器翻译服务应用程序接收到翻译请求时，它会将该请求转发到云端的必应翻译服务。这需要事先与客户沟通清楚。

通过在术语驱动发布页面中使用主机名网站集(HNSC)和友好 URL，多语言资源可映射到 URL 以便能够更加便捷地被流程引擎和最终用户理解。例如，一个法语网站的某发布页面可以映射到 <http://bar.fr/foo> 以替代其原有路径 <http://www.bar.com/fr-fr/Pages/foo.aspx>。

变体的另一大改变是加入了 SEO 优化。页面元数据会将其页面区域设置暴露给搜索引擎。另外，SharePoint 现在使用 HTTP 301 代码替代原来的 HTTP 302 代码用于主页重定向，这对于搜索引擎更合适。

2.9.8 内容搜索 Web 部件

内容搜索 Web 部件(Content By Query, CBQ)在发布网站中一直是一个强大的工具，用于满足内容聚合与汇总的需求。因为发布网站现在严重依赖搜索来运作，所以 SharePoint 2013 提供了新的名为内容搜索(CBS)的 Web 部件。

顾名思义，CBS 允许用户直接从搜索索引中聚合内容。如果你早前浏览了 2.9.5 节，在产品目录例子中的汇总页就用到了 CBS。

不同于 CBQ，CBS 并不局限于一个网站集。因为 CBS 基于搜索，所以它必须超越网站集边界。不过也由于此，CBS 里的查询结构可能不一定是最新的。除了延迟之外，CBS 仅呈现主要版本并且不能查询网站集中被标记为排除在搜索之外的内容。证明 CBS 查询是从索引直接提供服务最简单的办法就是重置索引并且观察返回的结果如何瞬间消失。不过不要在生产环境中进行这种尝试。

CBS 中的查询可以配置为基于页面或 URL 路径下的值的聚合内容。在呈现返回结果之前，可以选择对它们进行样式定义。



提示：CBS 会返回未经处理的 XML 格式的结果。这些结果可以使用替代 XSLT 的 HTML 及 JavaScript 代码段来定义样式。这些代码段在 SharePoint 2013 中称为显示模板且存储在母版页样式库中。

使用显示模板，定制 CBS 将比 CBQ 简单得多，因为可以使用纯粹的 HTML 与 JavaScript。

显示模板还应用到搜索相关的 Web 部件中，本章随后将对这部分内容进行介绍。

2.9.9 设计管理器

随着 SharePoint 2013 的来临，SharePoint Designer 的设计视图正式从该产品中移除了。无论这样做是否明智的决定，也无论继续将“Designer”一词用作产品名称的一部分是否还合理，这都超出了本书要讨论的范畴；不过，显而易见的是，对于高级用户应该有一个更便捷的方式来自定义 SharePoint。

微软建议用户使用 SharePoint 来定制产品。为了达到这个目的，微软引入了另一个称为设计管理器的工具，它能够帮助用户以类似于向导的方式来定制 SharePoint 网站。

开发人员与设计师或设计服务商一起工作来对 SharePoint 进行品牌建设。首先，开发人员从设计代理处获得非 SharePoint 的品牌建设资源(HTML、CSS、图片等)。设计代理可以使用任意 Web 设计工具来创建这些文件，如 Dreamweaver 或 Microsoft Expression。开发人员将接收的文件上传到 SharePoint。然后，借助于一个后台自动处理进程，SharePoint 特有的资源(*.master 与 *.aspx)就生成了。该进程会反复执行直至品牌建设工作全部完成。之后开发人员导出品牌建设资源并且创建一个解决方案包(*.wsp)来将自定义品牌建设部署到生产环境。

图 2-21 展示了替代 SharePoint Designer 的新设计管理器，用于定制 SharePoint。

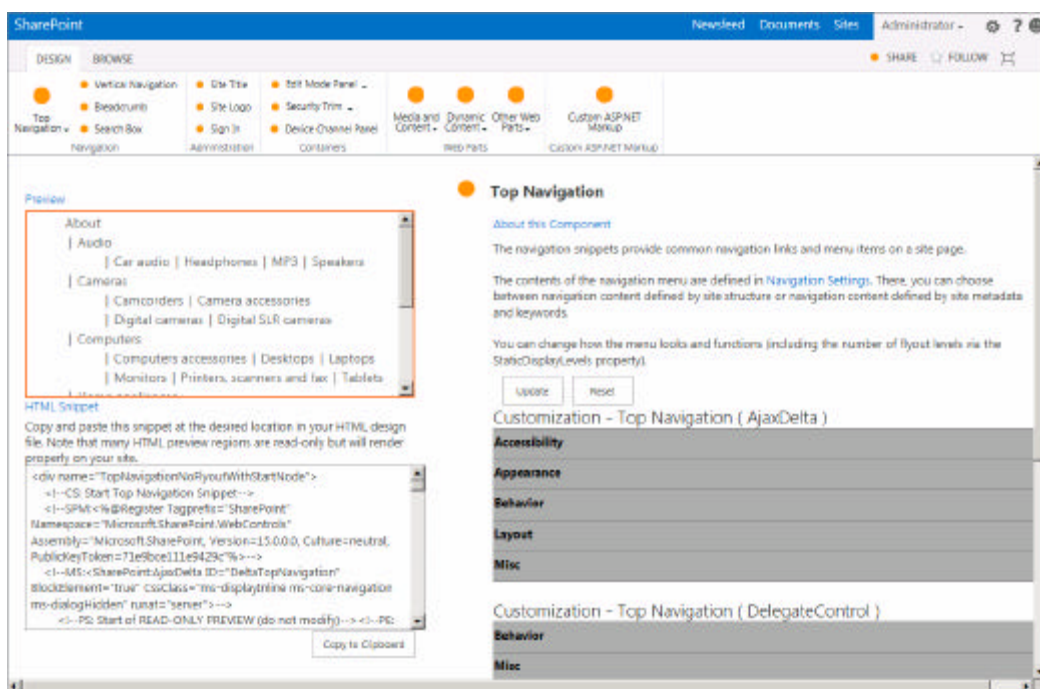


图 2-21

因为设计管理器提供了代码段库，所以 SharePoint 控件可以快速添加到设计页面中(参见图 2-21)。顶部的 Ribbon 菜单有助于查找并添加其他 SharePoint 组件。因为添加新控件

之后，仅会生成该控件的 HTML 表示形式，所以设计代理能够看见该控件在 SharePoint 中的外观如何，即使该设计代理使用的工具集不是 SharePoint 专用的。当设计代理将品牌建设资源发回后，SharePoint 会忽略 HTML 表示形式并呈现 ASP.NET 与 SharePoint 控件。

设计管理器并非完美，但同样，这仅是该工具的一个迭代版本而已。

2.9.10 移动端支持

SharePoint 2010 就已经能够支持移动设备了，但它受到的限制较大且难以定制。在 SharePoint 2013 中对应的新功能是设备通道。设备通道可以针对特定移动设备且与母版页相关联，该母版页允许为每一个目标设备进行特殊的自定义品牌建设实现。此外，网站设计人员还可以决定页面布局中的哪些部分可以包含在通道中。这使得管理移动设备的用户体验变得很简单。

可以在 Site Settings | Look and Feel | Device Channels 页面处配置设备通道。该设置仅在发布网站中可用。

2.9.11 图像呈现形式

SharePoint 2013 允许网站所有者能够通过在其网站中为媒体文件创建不同的呈现形式来优化用户体验。该新特性虽称为图像呈现形式，不过图像和视频都可以使用。

图像呈现形式正确配置后，SharePoint 将图像动态转换成其呈现形式中指定的设置并且随后它将缓存到 Web 前端服务器以便响应之后的请求。由于动态转换成合适的呈现形式设置需要的运算开销很大，因此 SharePoint 2013 借助基于磁盘的 BLOB 缓存技术来提升性能。



提示：直到为 Web 应用程序启用基于磁盘的二进制大型对象(Binary Large Object, BLOB)缓存后，图像呈现形式才能使用。基于磁盘的 BLOB 缓存在默认情况下未启用。关于启用基于磁盘的 BLOB 缓存的信息，请参阅 <http://msdn.microsoft.com/en-us/library/aa604896.aspx> 处的产品文档。

网站所有者可以在其网站的 Site Settings | Look and Feel | Image Renditions 页面开始定义合适的呈现形式。简单来说，图像呈现形式具有三种元素：名称、高度和宽度。

图像呈现形式定义好之后，内容作者能够上传图像，然后单击图像悬停面板查看不同呈现形式的实际效果，或者单击 Click to Change 链接以进一步裁剪图像来确保图像中的重要细节在由指定呈现形式引起的尺寸改变后依旧是重点。到此为止，图像上传后的图像呈现形式设置就完成了。

下一步就是内容作者在添加媒体文件到页面时选择想要的图像呈现形式了。图 2-22 显示内容所有者正如何为上传后的图像选择优化过的呈现形式，以便在 Windows Phone 7 上进行查看的。

带有呈现形式的图像可以结合呈现形式 ID、宽度或高度来进行引用。这里有两个例子：

```
  

```

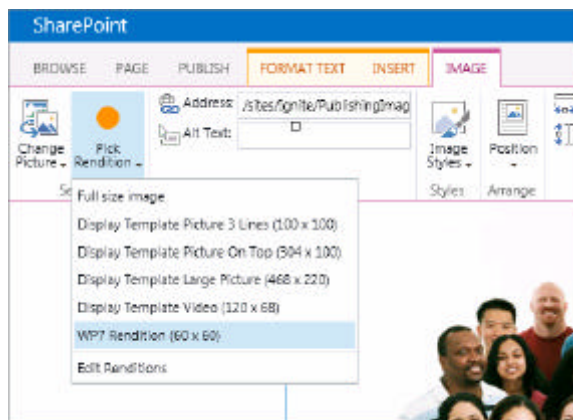


图 2-22

与设备通道一起使用时，图像呈现形式能够在移动设备上提供极佳的用户体验。图像呈现形式还能够帮助受到带宽限制的远程连接用户降低移动设备的带宽消耗。

2.9.12 应用程序驱动发布网站

互联网上大多数面向公众的网站都使用 SharePoint 发布模板构建。如本章所述，WCM 中有许多新的功能和改进用来构建强健的面向公众的网站。除了这些改进外，新的 SharePoint 应用程序也能应用在面向公众的网站中来提升用户体验。例如，可以使用提供程序托管的应用程序来将一个购物卡应用程序添加到网站中，或者可以使用 SharePoint 托管应用程序来为网站主页提供一个股票行情自动接收器。

总而言之，应用程序可以用来从面向公众的网站中抽离出一些为该网站所开发的通用功能，并且将其放入一些协作应用程序的上下文中以便随着这些应用程序一起交付或者供将来网站迭代使用。

2.10 搜索

SharePoint 中的搜索不区分边界。它可以物理遍历 SharePoint 服务器场中的有效内容或者服务器场外部的内容。

自从该服务应用程序出现在 SharePoint 2010 中以来，搜索就变成可以跨越服务器场使用的头等重要的服务。可以在服务器场 A 中发布搜索服务应用程序，然后在服务器场 B 中使用它。SharePoint 2010 的搜索有两种选择：企业搜索和 FAST 搜索。企业搜索包括 SharePoint Foundation 搜索、SharePoint Server 搜索以及搜索服务器。FAST 搜索，作为独立的 SKU 产品进行销售，在微软收购 FAST 之后加入到 SharePoint 产品中。SharePoint 2010

中的每一个搜索产品都有其各自的重点(以及成本)且表现出各自的独特优势和局限性。

随着当前 Office 2013 产品的发布及推广浪潮,微软决定将搜索统一成一个产品,并且对其进行扩展以便支持新的编程模型与整个平台的架构变化。新统一的产品做出了许多改变,后面会重点介绍。

2.10.1 搜索架构

搜索架构是搜索元数据属性的新名称。现在网站设置和网站集设置中都有搜索架构的链接,可以在该页面配置元数据属性映射。也就是说,网站所有者不再必须获取访问权限或者使用搜索服务应用程序来配置元数据属性了。

2.10.2 搜索导航

在 SharePoint 2010 中,让用户在不同搜索体验中快速跳转的唯一方式是创建新的搜索范围及结果页面,并将它们绑定在一起。这种方式下,新的搜索范围会在顶部导航中以范围下拉框的形式显示,以允许用户执行垂直搜索。

在 SharePoint 2013 中,可以在 Site Settings | Search Settings 页面配置每一个网站的搜索导航。这使得我们能够为用户创建不同的搜索体验并且无须使用额外的范围下拉框就能将这些搜索体验直接包含在搜索框中。

图 2-23 显示了 Search Settings 页面中新的搜索导航。

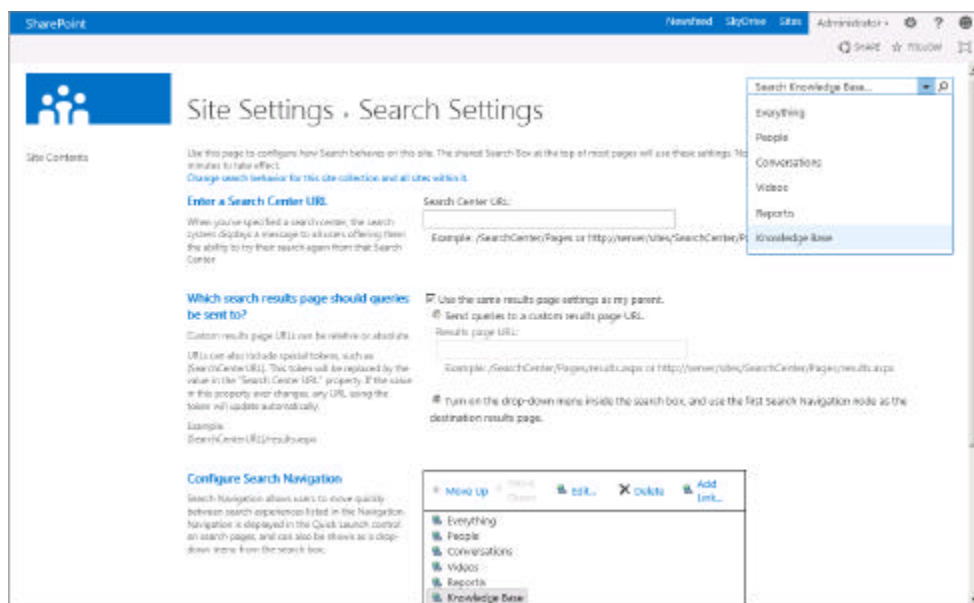


图 2-23

在站点级别执行的搜索设置可以由网站集管理员组在搜索设置页面进行网站集级别的重写。比如,可以在指定网站集中配置所有网站使用一个搜索中心并且完全忽略在网站级别进行的搜索导航设置。

2.10.3 结果源

结果源是 SharePoint 2010 里的范围与联合位置的新名称，它有三项主要改变。

第一，可以从搜索服务应用程序和每个网站的搜索设置中访问结果源。

第二，FAST 不再是信息源协议了；它被两种新协议所取代：Exchange 与 Remote SharePoint。使用 Exchange 协议来获取 Exchange 返回的结果，使用 Remote SharePoint 来获取在独立服务器场中托管的搜索服务应用程序返回的结果。

第三，可以使用新的 Query Transformation(查询转换)设置来指定过滤器以应用到查询中。这替代了 SharePoint 2010 中的搜索范围规则，且在对指定的一些内容(如客户列表或类似销售统计数据库的完整数据源)进行垂直搜索时能够快捷地使用。还有一个优秀的查询生成器，让你能够使用设计器来生成查询转换并且实时对返回结果进行分类查看。

例如，使用查询转换可以将托管属性的查询限定为具有特定值的信息。{searchTerms}author="Andy Au"这一查询转换仅返回包含该用户搜索术语(包含在{searchTerms}语句中)及作者是"Andy Au"的文档。还可以为查询添加前缀。Hockey{searchTerms}这一查询转换每次都会添加 Hockey 作为前缀并将它与用户搜索术语串联(AND)使用。



提示：SharePoint 2013 搜索中不再使用范围这一术语且已经将其从搜索 UI 中移除了。然而，术语范围的概念还是存在的。要执行垂直搜索，可以将搜索导航与结果源结合起来使用，以便完成相同的任务。

在 SharePoint 2013 搜索中，结果源的加入为整个组织中的各种信息(如销售数据或知识库文章)提供了强有力的机制来执行垂直搜索。

2.10.4 显示模板

如前所述，SharePoint 2013 采用了与 SharePoint 2010 不同的方式来定制搜索结果。现在，搜索相关的 Web 部件(即搜索结果 Web 部件)严重依赖于显示模板以控制来自搜索索引的结果的外观及表现形式。



提示：SharePoint 2013 中搜索结果仍旧是未经处理的 XML 格式，需要对其进行解析与提取。不过，可以使用显示模板替代 XSLT 来为搜索结果定义样式。

显示模板是 HTML 与 JavaScript 的代码段。SharePoint 产品中自带了许多预先设定好的显示模板，但是也可以创建自己的显示模板。与定制 SharePoint 的其他模板一样，只需要复制一个类似于你想要的现有显示模板，然后就可以着手开始了。可以使用任何想用的 HTML 编辑器来完成该任务。

显示模板可以在 Master Page Gallery | Display Templates | Search 处找到。如你所见，母版页样式库里还有其他类型的显示模板，以供其他工作任务使用，如 WCM。

2.10.5 结果类型

结果类型将所有事物联系在一起。可以使用结果类型将结果源或者一类搜索结果绑定到显示模板，搜索引擎会用其来展现搜索结果。

就如同显示模板，有许多开箱即用的结果类型，并且可以创建自有的结果类型。每一种结果类型都必须具有一个或多个条件来比较搜索结果，且还需要具有一个指定该搜索结果使用何种显示模板的操作。例如，如果搜索结果来自结果源 KbRs，那么名为知识库的自定义搜索类型将使用显示模板 KbDispTemplate.aspx 来显示结果。

与许多其他搜索配置一样，结果类型也可以同时在网站集级别与网站级别进行配置。

2.10.6 查询规则

在把核心结果呈现给最终用户前，查询规则是修改和加强搜索体验的最后一步。必须满足查询规则的一个或多个条件，该规则才能生效。还可以定义对任意搜索均生效的查询规则(这也就意味着没有规则)。

如果你需要决定是否要完成下列任务中的一项或多项，就要用到查询规则。

- (1) 在核心结果上添加已升级或参考链接。
- (2) 在核心结果上排列出一块额外的结果(称为结果块)。这样做可以将结果块一直显示在核心结果顶部。
- (3) 在核心结果中排列出结果块。如果结果块不是高度相关的，则它不会在第一页上显示出来。
- (4) 修改排列的结果，比如修改它们的排序。例如，可以对排列结果分类排序，以便 PDF 文档能够在搜索结果页面上排序优先。
- (5) 在 CBS Web 部件中配置显示结果块而不是在搜索结果 Web 部件中显示。
- (6) 为查询规则设置过期时间。

图 2-24 揭示了配置为面向本地 SharePoint 结果源(在名为 Everything 的垂直搜索中使用)的查询规则，且在用户的搜索术语包含 SharePoint 一词时触发结果显示。

在规则触发后，会添加一个已升级的链接(<http://sharepoint.microsoft.com/iusessharepoint/landing.aspx>)作为一条横幅(也可以就是一个链接)，并且随后呈现一个结果块，该结果块包含从指向一个外部网站(<http://blogs.devhorizon.com/reza>)的其他结果源中获取的结果。该规则会将所有这些置于核心排列的结果之上。

从概念上讲，能够使用查询规则做的事情有些类似于在 SharePoint 2010 中使用搜索关键字和最佳匹配。不过，查询规则更加强大并且高度灵活，还可设定条件以及定制。

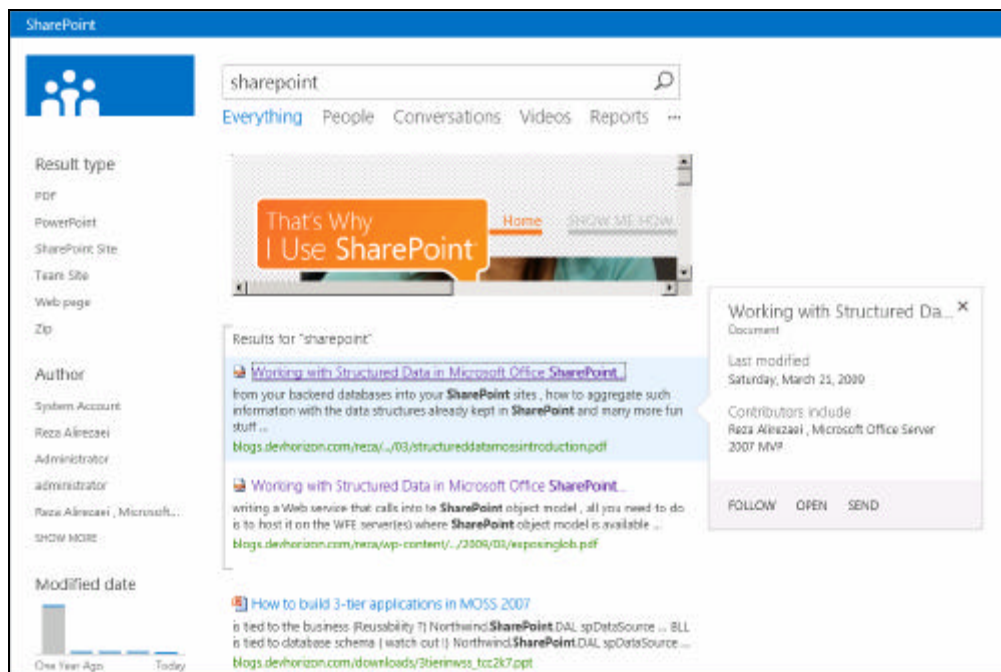


图 2-24

2.10.7 连续爬网

SharePoint 搜索使用完全或增量爬网来对内容进行分析。完全爬网关注内容的几乎所有一切,而增量爬网只关注收集内容中发生了改变的地方或者将更新后的 ACL 推送到索引中受到影响的条目中。由于 SharePoint 2013 严重依赖于在诸如 WCM 与社会化这样一些工作负荷中进行搜索,微软引进了一种新类型的爬网:连续爬网。

为了提供最大限度的新鲜内容,连续爬网会关注那些细小的变化并且使用变更日志来以更快和更高效的方式收集那些变化。连续爬网可以彼此重叠,这意味着一个连续爬网不会阻止另一个对这些变化进行收集。



提示 连续爬网在确保搜索结果最大限度包括最新内容方面扮演着重要角色。在 SharePoint 早期版本中,在内容查询与信息聚合情况下使用搜索,最大的缺陷就是搜索结果内容过于陈旧。

关于连续爬网有两个重要贴士:首先,连续爬网和增量爬网均仅对 SharePoint 内容源可用,可以选择两者之一。其次,连续爬网不能暂停或停止。只能够禁用它。

2.10.8 融会贯通

如果把目前为止介绍过的内容融会贯通,你会意识到 SharePoint 2013 中的搜索功能发生了如此彻底的变化。在图 2-25 中,将新的架构与 SharePoint 2010 中的企业搜索进行了高

层次架构概览的对比。你能看到这些不同之处。

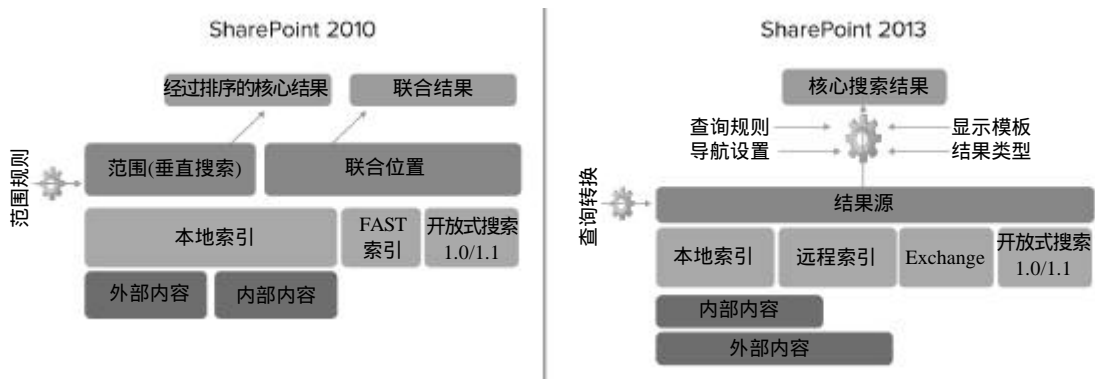


图 2-25

回顾一下所有内容，在 SharePoint 2013 中执行垂直搜索需要以下几个主要步骤。

- (1) 结果页面——使用(欢迎页面)搜索结果页面布局(即 `knowledgebase.aspx`)来创建结果页面。该页面显示垂直搜索的结果，如知识库。
- (2) 搜索导航——在搜索导航设置中添加结果页面的链接。
- (3) 结果源——为知识库网站或数据库这样的严谨内容创建带有查询转换的结果源，该查询转换用于过滤结果。
- (4) 绑定搜索结果 Web 部件——在第(1)步中创建的结果页面上编辑搜索结果 Web 部件，并且将它绑定到结果源。
- (5) 显示模板——创建控制结果显示样式的显示模板，以便搜索结果能够以设定样式在第(1)步中创建的结果页面上显示。
- (6) 结果类型——创建要将结果源绑定到显示模板的结果类型。
- (7) 查询规则——可根据需要创建查询规则来提供已升级的链接及结果块，该结果块来自第(3)步中创建的在其他垂直搜索中的结果源，比如，名为 Everything 的垂直搜索。

2.10.9 查询语言

对于开发人员，搜索一直都是跨网站集访问内容的绝佳方式，不过前提是对内容更新时效性要求不高。在 SharePoint 2013 中，连续爬网在某种程度上能够解决此问题。对于开发人员另一项新变化是可以使用查询语言。

对于许多了解并且喜爱 T-SQL 格式查询的人，有几条坏消息。SharePoint 2013 中不再支持使用 `FullTextSqlQuery` 类的 SQL 查询语言(SQL)。如果你计划迁移到 SharePoint 2013，现在就开始修改你的 SQL 查询吧。

FAST 查询语言(FQL)仍旧可以使用，但是建议开发人员使用关键字查询语言(Keyword Query Language, KQL)和语法来构建面向搜索引擎的查询。KQL 进行了一些增强改进且几乎可以在所有需要构建搜索查询的地方使用，比如，在查询转换和查询规则中都可以使用。表 2-3 列出了 KQL 查询的一些例子。

表 2-3 KQL 查询

| KQL 查 询 | 执 行 结 果 |
|----------------------------------|----------------------------------|
| Hockey | 返回包含 Hockey 或 hockey 的条目 |
| Hockey Soccer | 返回包含 hockey 与 soccer 的条目 |
| Hockey OR Soccer | 返回包含 hockey 或 soccer 的条目 |
| Hockey* | 返回像 Hockey 与 Hockey Jersey 一样的条目 |
| “ Hockey Jersey” | 返回带有 Hockey Jersey 词组的条目 |
| Firstname:A | 返回名字以 “ A ” 开头的所有人 |
| Title:Hockey IsDocument:1 | 返回标题带有 Hockey 的所有文档 |
| Author:Andy IsDocument:1 | 返回作者是 Andy 的所有文档 |
| Hockey FileExtension:pdf | 返回内容包含 Hockey 词语的 PDF 文档 |
| contentClass:STS_ListItem_Events | 返回所有日历中的事件 |
| contentClass:STS_ListItem_Tasks | 返回所有任务项 |

2.10.10 导出和导入搜索设置

如果你配置过搜索或者开发过使用搜索的 SharePoint 应用程序用作数据访问机制，你大概清楚要将你的搜索配置及设置在开发(DEV)、测试(QA)与生产(PROD)环境中彼此迁移不是一项简单的任务。针对不同的部署场景，必须编写 PowerShell 脚本与 XML 配置文档来确保你的配置能够适用于你所有的服务器场。

在 SharePoint 2013 中，可以在网站级别导出或导入搜索设置。该过程会处理查询规则、结果源以及为网站创建的托管属性，但需要迁移风格化的品牌建设组件，如搜索母版页、显示模板及 Web 部件。搜索结果页面的任何定制内容都不会被新的搜索导出与导入功能处理。

可以在 Site Settings | Search | Configuration Export 或者 Configuration Import 处导出或导入搜索设置。

2.10.11 搜索驱动的解决方案

由于搜索中的新改进，你可以构建使用 CSOM 与 REST 来执行搜索查询的自定义解决方案了。例如，下列 REST 查询会返回所有包含词语 “ hockey ” 的结果：

```
http://server/sites/sc/_api/search/query?querytext='hockey'
```

下列 REST 查询会返回所有包含词语 “ hockey ” 的结果，并将其按照最后修改日期和时间降序排列及按照级别升序排列：

```
http://server/site/_api/search/query?querytext='hockey'&sortlist=
'LastModifiedTime:descending,Rank:ascending'
```

下列代码段显示了如何在应用程序中使用 CSOM 来执行相同的 REST 搜索调用：

```
ClientContext ctx = new ClientContext("http://Tailspintoys.com/sites/sc");
var query = new KeywordQuery(ctx, ctx.Site);
query.QueryText = "hockey";
query.ResultTypes = ResultType.RelevantResults;
query.Id = Guid.NewGuid();
var queries = new KeywordQuery[1];
queries[0] = query;
SearchExecutor searchExecutor = new SearchExecutor(ctx);
var rcc = searchExecutor.ExecuteQueries(queries);
ctx.ExecuteQuery();
```

无论在应用程序或工作流中使用何种方式来执行 KQL 查询，返回的结果始终都是未经处理的 XML 格式，所以你需要对其进行解析和提取以便获得想要的内容。在搜索查询中无法获得 JSON 对象。

2.11 BCS

业务连接服务(Business Connectivity Service, BCS)使得 SharePoint 与外部系统的集成工作变得简单。为达成该目标，BCS 将外部系统功能映射到标准接口以定义如何与其进行交互。从核心层面看，有 4 个 BCS 概念必须理解。

- ？ 外部系统——使用开箱即用或自定义连接器来使 BCS 可以访问的任何提供支持的内容源。Web 服务、数据库、业务线(Line Of Business, LOB)系统、Web 2.0 服务，甚或包含数据的.NET 对象都可以称为相对于 BCS 来说的外部系统。
- ？ 外部内容类型(External Content Type, ECT)——外部内容类型定义外部系统的架构和数据访问功能及其特性。外部内容类型通常称为 BCS 的基础。



提示：从概念上理解，可以将外部内容类型看成业务实体。实体是客户、销售订单等真实事物的抽象概念。实体就是在 BCS、C#类、实体框架或数据库架构中使用的对象。每一个实体都有名称、元数据、关联以及固定的操作——如创建、读取、更新与删除——这使得访问外部数据成为可能。每个实体都能拥有多个实例，如数据表中的行数据、SharePoint 列表项或者同一对象的各个实例。

- ？ BDC 模型(BDCM)——XML 文档描述一个或多个外部内容类型及其资源，如区域性字符串、元数据、权限及连接信息。有了 BDC 模型，开发人员不再需要了解每个外部系统的细微差别了。

- ？ 外部列表——外部列表使用户可以像原生的 SharePoint 列表那样使用现有的 LOB 数据。在外部列表中，数据并非实际位于 SharePoint 中；相反，它在每次该外部列表被访问时从后端外部系统中直接抽取显示。

幸好，BCS 核心概念在 SharePoint 中并未发生改变。相反，倒是有了一些强化。现在来看看这些变化。

2.11.1 OData 连接器

SharePoint 2013 自带了供 BCS 使用的 OData 连接器。这使得开发人员能够在 BCS 解决方案中使用从兼容 OData 的服务中获取的数据。在 Windows AzureStore 中，微软发布了许多免费或收费的 OData 数据源，这些数据源可以在你自己的 BCS 解决方案中使用。比如，可以在 <http://services.odata.org/Northwind/Northwind.svc/> 处使用免费的 Northwind OData 数据。下列代码段显示了围绕 Northwind OData 源构建的 BDCM 的一部分：

```
<LobSystem Name="ODataNWModel" Type="OData">
  <Properties>
    <Property Name="ODataServiceMetadataUrl" Type="System.String">
      http://services.odata.org/Northwind/Northwind.svc/$metadata</Property>
    <Property Name="ODataMetadataAuthenticationMode"
      Type="System.String">PassThrough</Property>
    <Property Name="ODataServicesVersion"
      Type="System.String">2.0</Property>
  </Properties>
  <LobSystemInstances>
    <LobSystemInstance
      Name="http://services.odata.org/Northwind/Northwind.svc">
      <Properties>
        <Property Name="ODataServiceUrl" Type="System.String">
          http://services.odata.org/Northwind/Northwind.svc
        </Property>
        <Property Name="ODataServiceAuthenticationMode"
          Type="System.String">PassThrough</Property>
        <Property Name="ODataFormat"
          Type="System.String">application/atom+xml</Property>
      </Properties>
    </LobSystemInstance>
  </LobSystemInstances>
  <!-- Code Omitted for brevity -->
</LobSystem>
```

如前所述，SharePoint 列表数据已经可以作为 OData 源公开使用了。结合所有内容，恰当地说，SharePoint 能够同时作为 OData 源的生产者和消费者。

2.11.2 BCS 驱动的应用程序

SharePoint 2010 中的外部内容类型有一个问题，就是必须在服务器场级别且需要服务器场管理员权限才能进行配置。SharePoint 2013 引入了新的编程模型且其重点是在云计算

和应用隔离上，该问题就不复存在了。

在SharePoint 2013 中，开发人员可以将外部内容类型包含在其应用程序中，并且无须租户管理员参与即可直接从其应用程序中访问来自如 Netflix 或 Windows Azure 的任意外部系统的外部数据。



提示 :只有 Visual Studio 2012 才支持从 OData 源中生成 BDCM。SharePoint Designer 2013 现在还无法提供该支持。

下列代码段显示了如何在应用程序中定义外部列表：

```
<ListInstance Url="$Resources:core,lists_Folder;/BCSEmpoweredList"
Description=" A BCS Empowered External List"
OnQuickLaunch="TRUE" Title="BCS-ECT" Id="BCSEmpoweredList">
  <DataSource>
    <Property Value="" Name="LobSystemInstance"/>
    <Property Value="" Name="EntityNamespace"/>
    <Property Value="" Name="Entity"/>
    <Property Value="" Name="SpecificFinder"/>
    <Property Name="MetadataCatalogFileName"
      Value="BDCMetadata.bdcx" />
  </DataSource>
</ListInstance>
```

可以看到该代码段有一些元素，每一个都代表外部列表的一个属性。比如，Entity 属性应该包含业务实体的名称，该名称就是外部列表指向的后端系统中的实体名称。

2.11.3 CSOM 与 REST 接口

就如同到目前为止介绍过的其他工作负荷，BCS 拥有新的 JavaScript 对象模型以支持远程调用。

比如，下列 REST 查询会返回名为客户的外部列表中的所有客户。该列表由应用程序部署：

```
http://server/sites/sc/_api/lists/getbytitle('Customers')/items
```

下列代码段显示了如何使用面向 CSOM 的 JavaScript 并将返回结果绑定到表格来完成相同的 REST 调用：

```
BCSEmpoweredList.Grid.prototype = {
  init: function () {
    $.ajax({
      url: this.surlWeb + "_api/lists/getbytitle('Customers')/items?$select=BdcIdentity,CustomerID,ContactName",
      headers: {
        "accept": "application/json",
```

```
        "X-RequestDigest": $("#__REQUESTDIGEST").val()
    },
    success: this.showItems
  });
}
```

下列代码段显示了如何在 C#代码中面向 CSOM 使用 BCS 对象模型：

```
var ctx = new SP.ClientContext();
var web = ctx.get_web();
entity = web.getAppBdcCatalog().getEntity(entityNameSpace, entityName);
ctx.load(entity);
lobSystem = entity.getLobSystem();
ctx.load(lobSystem);
lobSystemInstances = lobSystem.getLobSystemInstances();
ctx.load(lobSystemInstances);
ctx.executeQueryAsync(success, failure);
```

代码以建立上下文及引用业务实体开头。一旦引用实体，就加载内容且会注册两个事件处理程序来执行回调。在成功的事件处理程序中，可以编写处理返回结果的代码。

2.12 本章小结

祝贺你完成了本章的阅读！

本章介绍了伴随 SharePoint 2013 到来的许多新功能和增强改进，以及 SharePoint 编程模型的变化。所有这些新功能的介绍都是为了帮助你理解作为 SharePoint 开发人员要在 SharePoint 2013 中构建应用程序大体有哪些内容需要掌握。我们多次提及过可以在 SharePoint 产品中使用相同的 CSOM 与 REST 代码模式且只需修改目标指向即可应对不同的工作负荷。这就是在 SharePoint 2013 中的编程方式。

本章介绍过的每一个工作负荷都会在本书中详尽介绍。请继续读下去吧！