

Broadview
www.broadview.com.cn

Microsoft

2005
EDITION



Microsoft®

.NET 2.0

应用程序调试

DEBUGGING MICROSOFT® .NET 2.0 APPLICATIONS

[美] John Robbins 著
陈缘 邹建峰 郑琼 译



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

第2版

第2版

第2版



.NET 2.0 应用程序调试

第2版

第2版

第2版

TP393.09/169

2008

.NET 2.0 应用程序调试

DEBUGGING MICROSOFT® .NET 2.0 APPLICATIONS

[美] John Robbins 著
陈缘 邹建峰 郑琼 译

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是资深调试专家 John Robbins 关于调试技术方面的第 4 本著作。在本书上一个版本——《Microsoft .NET 和 Windows 应用程序调试》的基础上,作者对大部分内容进行了全面的更新。删除了上一版本中的“本机代码的强大工具和技术”部分,剩下了前面的 3 大部分:“调试概述”、“强大的调试技术”和“强大的调试工具”。

在第 1 部分中,作者首先介绍了 Bug 的来源及调试的基础知识,并在该部分的结尾处,对以往读者提出的一些具有代表性的问题——做了解答。而后,在第 2 部分中,作者介绍了 Visual Studio 2005 在调试方面的新特性,以及如何使用 Visual Studio 2005、WinDBG、SOS、ADPlus 等进行应用程序调试。最后,在第 3 部分,作者介绍了如何对 Visual Studio 的 IDE 进行扩展,以及如何编写你自己的代码分析规则。

本书的最佳读者对象是拥有一定开发经验的中高级开发人员和调试人员。

Copyright © 2007 by Microsoft Corporation. All rights reserved.

Original English language edition © 2007 by Microsoft by John Robbins.

All rights reserved. Simplified Chinese edition published by arrangement with the original publisher, Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文简体版专有版权由 Microsoft Corporation 授予电子工业出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字:01-2007-0566

图书在版编目(CIP)数据

.NET 2.0 应用程序调试 / (美)罗宾斯(Robbins,J.)著;陈缘,邹建峰,郑琼译.—北京:电子工业出版社,2008.3

书名原文:Debugging Microsoft .NET 2.0 Applications

ISBN 978-7-121-05970-4

I.N… II.①罗…②陈…③邹…④郑… III.计算机网络—程序设计 IV.TP393

中国版本图书馆 CIP 数据核字(2008)第 016284 号

责任编辑:陈元玉

印 刷:北京市天竺颖华印刷厂

装 订:三河市金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×980 1/16 印张:29 字数:450 千字

印 次:2008 年 3 月第 1 次印刷

定 价:75.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

SQL Server 2005 微软官方权威参考书



《Microsoft SQL Server 2005技术内幕： 存储引擎》

[美] Kalen Delaney 著
葛伟 方磊 揭磊骏 译
刘莹 审校



《Microsoft SQL Server 2005技术内幕： T-SQL查询》

[美] Itzik Ben-Gan, Lubor Kollar, Dejan Sarka 著
赵立东 唐灿 刘波 译
赵立东 审校

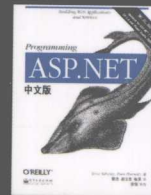


《Microsoft SQL Server 2005技术内幕： T-SQL程序设计》

[美] Itzik Ben-Gan, Dejan Sarka, Roger Wolter 著
赵立东 译

自SQL Server 2005发布以来，该领域公认的经典之作《Microsoft SQL Server 2005技术内幕》也面世了。然而，就像《Microsoft SQL Server 2005技术内幕》的作者Kalen Delaney所说，SQL Server 2005太庞大了，以至于一本书无法容纳这么多的内容。因此，SQL Server 2005技术内幕被分成4册单独出版，博文视点引进了全部4册，以飨读者，其中第4册《Inside SQL Server 2005: T-SQL Tuning and Optimize》将于近期上市。

作为SQL Server领域公认经典，该系列图书不但囊括大量鲜为人知的技术内幕，而且包含作者的丰富经验和一些非常实用的技巧，是每位SQL Server开发者案头必备之书。



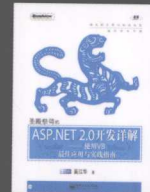
《Programming ASP.NET中文版（第3版）》

[美] Jesse Liberty, Dan Hurwitz 著

瞿杰 赵立东 张昊 译

郝刚 审校

■ Amazon五星名著，技术进阶必读



《圣殿祭司的ASP.NET 2.0开发详解——使用VB》



《圣殿祭司的ASP.NET 2.0开发详解——使用C#》

奚江华 著

■ 循序渐进讲解ASP.NET 2.0技术精要



《Programming C#中文版（第4版）》

[美] Jesse Liberty 著

刘基诚 李愈胜 刘卫卫 译

■ 深入浅出地讲述C#和如何使用C#编写.NET应用程序

■ 连续四版畅销全球

对《.NET 2.0 应用程序调试》的赞誉

“《.NET 2.0 应用程序调试》是众多卓越的调试开发工具与技术的实战手册——而这仅是该书优点的一部分。你还能从中了解到如何解决非常棘手的问题，以及如何让你在一开始就避免那些 Bug 发生的工程实践方面的建议。John Robbins 以一种生动的写作方式，提供了许多实用参考资料，它是一个装满实用工具和代码的百宝箱，所以，你可以马上就开始应用他的建议（而你也应该这么做）。”

——Maria Blees，软件开发工程师，Microsoft Corporation

“《.NET 2.0 应用程序调试》是.NET 调试方面的权威书籍。我觉得我在每一页中都学到了新的东西，并且迫不及待地要一瞥下一章的内容。这本书将会成为我们开发团队的必读之书。”

——Jeff Lewis，资深开发者，Xactware Inc.

“既然我们无法消灭 Bug，我们就得高效地处理它们。这是一个涉及很多方面的问题，包括态度、思考技巧、计划，以及工具。John Robbins 为我们深度覆盖了所有的东西：如何在一开始就避免 Bug 的产生，如何系统地解决 Bug，以及如何最有效地使用所有我们可以使用的工具。《.NET 2.0 应用程序调试》将会拯救你的饭碗！”

——David Douglass，技术顾问 II，ADP, Inc.

“《.NET 2.0 应用程序调试》真是一笔很棒的资源。你能从中获得一整套可以帮助你更好地进行调试的关键技巧，而且 John 对那些技术的激情讲解将确保你编写出最好的代码，并使你在 Bug 发生前就消灭它们。”

——Jim McGregor，HBOS Plc

“拜读完《.NET 2.0 应用程序调试》之后，我只能说，如果 John 生活在 300 多年前，那他肯定会被当作一个巫婆而烧死。”

——Dave Montgomery，Triant

“John 是主代码害虫的灭杀者，而《.NET 2.0 应用程序调试》是每一位.NET 开发者书架上都应有的为数不多的几本书籍之一。这是一本扣人心弦的读物，它的内容从 Visual Studio 中最浅显的技巧到低级别的.NET-Windows 交互 Bug。最棒的是，这本书融入了 John 在多年帮助其他开发人员寻找和清除最坏、最有害 Bug 的过程中所获得的智慧。”

——Don Keily，资深技术顾问

“John Robbins 再一次编著了一本生动且非常实用的调试书籍。通过最新的.NET 2.0 技术,深入探讨如何最有效地使用 IDE、编译器和调试器,以及他那成堆的实用工具源代码,John 继续与我们分享着他具有深度的经验。《.NET 2.0 应用程序调试》通过提供快速消灭,乃至完全避免 Bug 的正确工具、方法和建议,节省了无数的调试时间。”

——Thomas Fejes, 独立顾问, DeltaFusion Pty Ltd

“对于一个软件工程师来说,获得足够拯救一个项目的调试技能,是职业生涯上非常重要的一项任务。阅读《.NET 2.0 应用程序调试》就是获取这些技能的一个重要步骤。这本书中的某些东西适合所有的经验级别。你在读完之后可能会希望能有更多的 Bug 可以跟踪,而这仅仅是为了将你所学的东西付诸使用。”

——Mark Bartosik, www.leakbrowser.com

“软件开发最多可以花去他们一半的开发时间进行调试。John Robbins 的《.NET 2.0 应用程序调试》会帮助所有.NET 程序开发者学习到主动式的调试技能。这些技能可以帮助他们更快地找到 Bug,并大大地减少沮丧感。这是.NET 开发者必备的书籍之一。”

——Roger Orr, OR/2.

“如果你觉得自己是个调试权威,并且已经没有什么可以学习的,在放弃 John Robbins 的《.NET 2.0 应用程序调试》前请你三思。书中有关 Bug 猎杀的提示、工具及信息量令人吃惊,而这些对于帮你搜寻隐藏于某些应用程序内部最棘手和最吓人的 Bug,绝对绰绰有余。这本书绝对是必读的!”

——Roberto A. Farah, 专家级工程师, Microsoft

译者序

俗话说，人无完人。在计算机程序设计领域中，这一点得到了非常好的印证。如果有哪个程序员某日站出来宣称自己从未创造过任何 Bug，那他的名字准会在第二天出现在全世界的 IT 门户网站上。从某种意义上说，软件的规模和 Bug 的出现概率成正比。即软件的规模越大，其包含 Bug 的可能性就越大。例如，编写一个 100%没有 Bug 的 Hello World 程序非常简单，但要编写一个 100%没有 Bug 的操作系统则几乎是永远不可能实现的。因此，就目前而言，没有 Bug 的世界只存在于人们的想象之中。

作为一个凡人，我们虽然不能靠念几句咒语就让 Bug 变成南瓜车，但却可以尽己之力减少 Bug 的产生。而这也正是 John Robbins 在本书中要告诉我们的重要内容之一。对于剩下的顽固 Bug，John 告诉我们：拿起武器，斗争到底。那这武器是什么？武器就是调试工具。

在程序员中，知晓调试者十之八九，但真正深谙调试之道者却有如凤毛麟角。为什么？正如 John 所说的那样，在大学的计算机专业课设置中根本没有专门的软件调试课程，即便偶尔有涉及调试内容的课程，也都是语焉不详，一带而过。同时，在校园之外也鲜有此类书籍的身影。可以说，除却本书及本书的前两个版本，能在书店发现一本介绍软件调试的专业书籍，就好似中了 500 万元的彩票一般。所以，对于很多新手来说，按下 F5，然后祈祷世界和平，是他们唯一能做的事情。

作为 Wintellect 公司的资深专家和 MSDN Magazine 杂志“Bugslayer”专栏的资深作家，John Robbins 用他宝贵的经验为我们照亮了调试征途上原本充满雾霭的漫漫长路。感谢 John Robbins，是他让我们拥有了一个可以站立的巨人肩膀。

本书的翻译过程有些曲折，但在邹建峰和郑琼的帮助下，我终于得以完成此书。8 个章节中，我完成了其中的前 7 个章节，他们协助我完成了最后的第 8 章。在此，我要特别感谢他们为此所付出的努力。还是那句话，人无完人。虽然我在翻译过程中竭尽全力查阅相关资料，但肯定还是会存在某些不足之处，希望各位在遇到这些“Bug”时，能拨冗发个错误报

告给我。我的电子邮件地址是: yuan505@mail.zjgsu.edu.cn。最后, 本书的源代码可以在作者的网站 <http://dti.wintellect.com> 下载到。不过访问这个地址所需的时间有时足够烧好一壶开水, 所以, 为了方便国内的读者, 我将代码下载后也放了一份在 <http://econet.zjgsu.edu.cn/dti/Debugging.Zip> 中。

最后, 衷心感谢您阅读本书!

陈 缘

2008 年元月, 于浙江工商大学

Acknowledgments

致谢

如果您读过本书的前两版，看过我写的“Bugslayer”专栏，听过我在会议上的发言或我讲的课，谨此致谢！您对于程序调试及写出更好的代码的不懈热情支持我战胜了再版过程中的困难。非常感谢大家给予的 E-mail 交流和讨论。你们让我学到了很多。谢谢。

四位不凡的人使得本书得以问世，对于他们我不胜感激：Valerie Woolley（策划编辑）、Christophe Nasarre（技术编辑）、Joel Rosenthal（文字编辑）及 Ben Ryan（组稿编辑）。他们将我杂乱无章的文字编辑成你们手中的书。他们为此做出了巨大的努力。

如同之前的几个版本，我要向以下众多的人致以无尽的谢意——我的“审阅团”。这些勇敢的人们弥补了我英语文字的严重不足，并且贡献了很多惊人的调试技巧。他们是当今该行业中精英的代表。他们分别是：Mark Bartosik、Maria Blees、Derek Cooke、Chad Coon、Stoyan Damov、David Douglass、Richard Dutton、Roberto Farah、Thomas Fejes、Moishe Halibard、Don Kiely、Martin Kulov、Jeff Lewis、Spencer Low、Jim McGregor、Dave Montgomery、Roger Orr、OsirisPedroso、Jeff Scanlon、Austin Thompson 及 Jalil Vaidya。其中特别要感谢 Maria Blees，她对本书做出了不可估量的贡献。

我荣幸而惶恐地将自己与我的 Wintellect 同道们相提并论，他们通过各种途径对本书做出了巨大贡献：Cara Allison、Jim Austin、Jim Bail、Paula Daniels、Sam Easterby、Sara Faatz、Todd Fine、Lewis Frazer、Bethany Jones、Steve Porter、Jeff Prosise、Jeffrey Richter、Kenn Scribner、David Shoots、Justin Smith 及 Gregory Young。

最后，最大的感谢一如既往地归属于我的妻子，Pam。她在我奋笔写书期间牺牲了众多的夜晚和周末。亲爱的，都结束了。你的丈夫可以回到你身边了。

致我生命中最重要的三个人。

我的父母，Rob 和 Doris Robbins，感谢你们给了我
一个充满冒险的美妙童年。

我的成长经历就是一部国家地理特辑。

你们的爱和支持是我的一切。

我的姐姐，Deborah Matthews。我希望，

某一天我能像你一样，

精明、风趣，并且勇敢。

Introduction

介绍

一句话，该死的 Bug，是把你推入工程逾期、无尽的熬夜和被忿忿的同事所包围的绝境的罪魁祸首。Bug 足以导致你的生活悲剧，如果它们过多地侵入你的软件，客户就可能停止使用你的产品，你就可能因此失业。因此，Bug 是需严肃对待的问题。

很多时候，业内人士简单地将 Bug 描绘成让人头痛的麻烦。问题远不止这些。所有的工程师都能指出几个含有失控 Bug 的工程，甚至可以列举出因过多程序错误而导致产品根本不可用以至使公司倒闭的例子。当我写本书第 1 版的时候，美国国家航空航天局就因为在设计阶段溜进了一个 Bug 而损失了一个火星航天探测器。当我写本书第 2 版的时候，因为 GPS 软件中的电池更换所产生的一个程序错误导致了一枚炸弹偏离其原定目标而砸向了美国特种部队士兵。当我写本书第 3 版的介绍的前一周，微软发布了一个程序补丁以挽救之前一个程序补丁造成的微软 IE 6 上的巨大缓冲区溢位漏洞。尽管 Bug 开始逐渐地受到应有的关注，但距离我们形成一种高度重视 Bug，并不再将其视为开发过程中不可避免的小问题的开发氛围，我们还有很长的路要走。

我希望本书提供的信息能够帮助你学会如何从写程序开始就尽可能地避免 Bug——这样，当到了需要调试纠错的时候，你就可以节省很多时间。多数开发小组由于没有意识到这一点，而将平均占开发周期近百分之五十的时间用在了调试纠错上。用正确的方法进行调试将极大地缩短调试所需的时间，这意味着你能够更快地交货。虽然你不能在需求采集和设计过程中节省更多的时间，但你肯定可以在调试过程中采取更加聪明的方法。本书对调试纠错的过程进行了全面的剖析。我不认为调试是一个孤立的步骤，它应该被视为整个产品开发周期中的有机部分。我认为，你应当从需求阶段就开始进行调试工作，并将其贯穿至最终版本的完成。

有两个问题使得在微软 .NET 环境中进行调试时艰难而且耗时。一是调试过程从来都是一门自学的手艺——你几乎一直以来都是靠自己的摸索前进的。即便你有计算机专业的学位，我打赌在你的大学里从没有一门课程是专门针对调试过程的。除了一些诸如设计某些根本没有人用的语言的自动程序校验之类的神秘兮兮的课程，或者为盲目乐观的超大平行进程计算机开发的调试器之外，对应用于商业软件的调试科学似乎从来都不大受教育机构的欢迎。有的教授指出，当你从开始写程序的时候就不应该出现 Bug。尽管这是一个非常好的观点并且是我们孜孜以求的理想境界，但现实却往往不尽如人意。学习系统的、被实践验证过的调试技术虽然不能让你彻底避免写出带 Bug 的程序，但按照本书进行的练习将帮助你减少代码中出现 Bug 的数量，并能更快地追查出程序中出现的错误。

第二个问题是, 尽管市面上有很多优秀的关于特定 .NET 技术的书籍, 却没有哪一本书在调试问题上能够深入到有实用价值的内容。要想对任何技术进行有效的调试, 你必须掌握远远多于一本特定技术专著所提供的內容。知道如何写一个嵌入到 ASP.NET 页面中的 ASP.NET 控件是一回事, 而能够找出那个 ASP.NET 控件中的错误完全是另外一回事。要想对那个 ASP.NET 控件进行纠错, 你首先必须了解 .NET 和 ASP.NET 的细节, 知道 DLL 们是如何被放进 ASP.NET 的临时缓存的, ASP.NET 又是如何着手找到那些控件的。有的书上将远程数据库连接之类的复杂特性的实现描述得很简单, 诸如使用目前最热门的技术就能解决问题, 但当 `db.Connect` (“Foo”) 在你的程序中失灵时——事实上它注定会失灵——你就得靠自己来找出并且修复这条技术链上断掉的环节。尽管个别涉及项目管理的书的确论及调试, 但它们的侧重点多倾向于行政管理事务而不是从开发者的角度出发的。这些书或许包含了一些关于如何计划调试的有用信息, 但当你面对着一个损坏的数据库或一个不停重启的 ASP.NET 工作进程而束手无策时, 它们却帮不上什么忙。

本书的灵感来源于我在作为一个努力按时发布高质量产品的开发者和管理者, 以及作为一个试图帮助其他人按时交付产品的顾问期间所经历的磨难。这些年来我学会的技巧和技术, 足以用来对付这两类助长了基于微软视窗的程序开发的挑战性问题。对于正规程序调试训练上的资源匮乏, 我在本书的第 1 部分中已给出了一个程序调试的速成教程——明确针对商业软件开发。至于第二个问题——对于一本专门针对 .NET 的书的需求——我想我已经提供了一本能够填补专业技术和现实应用之间空白的书。

在过去的 11 年中, 我非常幸运地并能够有机会几乎专门致力于程序调试。其中一些经历帮助我形成了在程序调试方面的独到见解。第一个经历是在 NuMega Technologies (即现在的 Compuware), 当时我是第一批从事 BoundsChecker、TrueTime、TrueCoverage 和 SoftICE 等超酷项目研究的工程师之一。在 NuMega 工作期间, 我开始在 MSDN 杂志上撰写 “Bugslayer” 专栏, 之后我离开 NuMega 并开始撰写本书的第 1 版。当我和其他软件工程师们在开发各种人们能够想到的程序期间进行交流时所发送的那些令人难以置信的 E-mail, 让我了解到更多软件工程师们现今在交付软件产品时所面临的问题。

最后, 决定我观点的最关键经历是 Wintellect 的创建, 这给了我走出去并帮助遍布全球各地的公司解决那些棘手问题的机会。想象一下, 当你在半夜两点的时候坐在某办公室黔驴技穷时, 如果你再不解决 Bug 问题, 你的客户就要终止这笔生意——这场景有些恐怖, 但同时也可以让你的肾上腺素上溢。和诸如微软、eBay、Intuit 等其他众多公司最优秀的工程师们合作, 是我所知的学习解决各种程序调试问题的窍门和技术的最佳途径。

谁应该读这本书

本书是我为那些不想再熬夜调试程序，并且希望提高他们的代码及他们所服务的机构的工作质量的程序开发者们所写的。同时本书也是写给那些希望建设更加高效率、有影响力的团队的管理者和领导者们的。

从技术角度来说，理想的读者是那些有过一至三年在 .NET 或 Windows 平台上的开发经验的人。同时我还希望我的读者是某个现实开发部门的一员，并至少完成过一件产品的开发。尽管我并不太在意这一点，但软件行业将拥有此经验级别的开发者称为中级开发员。

高级开发员从本书中也很可能获益良多。在我收到的关于前几版的热情洋溢的 E-mail 中，有很多是来自于那些起初并没有指望从中学到什么的高级开发者。我很激动本书提供的一些工具能够被用来充实他们的工具箱。再有，和前面的版本一样，被称为审阅团的一群极其优秀的朋友们，在我向微软出版社交稿之前，审阅了各个章节。这些我在本书前面的致谢部分里所罗列的工程师们，可谓是开发精英中的精英，正是他们保证了所有阅读本书的人都能有所斩获。

如何阅读本书及第 3 版中的新内容

第 1 版侧重于 Microsoft Visual Studio 6 和 Microsoft Win32 的调试。第 2 版侧重于 Microsoft Visual Studio .NET 2003 和一个全新的编程范式、.NET，以及 Windows 本机开发。而这一版则全力关注 .NET 2.0 和 Microsoft Visual Studio 2005。对于仍然大量从事本机开发的读者来说，本书将在未来的另一个版本中集中透彻地讨论本机 C++ 开发。

第 1 版中仅本机开发的内容就占了 512 页，第 2 版中关于本机和 .NET 的内容占了约 850 页，而本版则有将近 480 页的内容论及 .NET。当我着手撰写这部 .NET 专版的时候，本设想仅是对部分内容的改写，但结果却成了至少百分之八十的内容被改写。在第 6 章“WinDBG、SOS 和 ADPlus”（本书最长的章节）中，总共仅有 4 段文字沿袭自上一个版本。缩小本书的聚焦范围，以及 .NET 2.0 和 Visual Studio 2005 的重大改变是本书大量改写的主要原因。而另一个主要原因则是第 2 版的读者们与我交流的 E-mail 和展开的讨论，以及他们对于如何将此书写得更好的出色批评和建议。

代码范例

在之前的版本中，本书所含的大量代码和公用程序是广受欢迎的。第 1 版有 2.5MB 的文本源代码，而第 2 版中则有 6.9MB。尽管因为将重点转移到 .NET 上后数量有所减少，但 Windows 资源管理器上的显示告诉我，本版至少有 9.7MB 的代码！请记住，这仅仅是文本和所支持的文件，不是编译指令。审阅团的一些成员说，本书包含的代码比目前绝大多数的开发类书籍的全文还要庞大。同时，所有这些代码都是最新的，因而能够充分利用 .NET 2.0 的特性。

本书中，我将作为代码范例的一部分的源文件用“\”在目录前标出。例如：当讨论将源代码装入 WhoAmI 目录下的 WhoAmI 外接程序中的 Connect.cs 文件时，我将文件参数标示为 \WhoAmI\Connect.cs。鉴于本书文本中有众多的代码引用，这样可以免去每次都要读一遍 *Book_Source_Code_Installation_Directory\WhoAmI\Connect.cs* 的麻烦。

本书的组织结构

我将本书分为 3 大部分。前两部分（第 1 章至第 6 章）您应该按顺序阅读，因为该部分的信息是按逻辑顺序来组织的。

在本书的第 1 部分“调试概述”（第 1 章至第 4 章）中，我定义了不同种类的 Bug 并提出了一个所有出色的开发者都应遵循的调试程序，还讨论了正确的团队调试所需的基础构造要求。我强烈建议您特别关注第 2 章“为调试做好准备”中关于建立符号服务器和建立源服务器的介绍。因为你可以（并且应该）在编写代码的阶段就做大量的调试工作，我在书中谈及如何在写代码的过程中进行前瞻性的程序调试。第 3 章“边编码边调试”中有关断言的讨论应该作为 .NET 这个话题的总结陈词。为了帮你获得成为一个优秀的程序调试员所需知识的广度和深度，在第 4 章“常见的 .NET 调试问题”中，我回答了从撰写最近的这个版本以来我得到的所有常见程序调试问题。我同时还提及了关于如何解决 Visual Studio 中的问题的窍门，这样你在和这个工具的角力中就不至于拔自己的头发了。

在本书的第 2 部分“强大的调试技术”（第 5 章至第 6 章）中，我深入阐述了你将在 .NET 开发中用到的两个主要调试工具的方方面面：Visual Studio 和 WinDBG。Visual Studio 较之以前的版本发生了本质上的改变，而你唾手可得的方便度令人难以置信。对我来说幸运的一点是，各类文献对功能特性的描述不免有失偏颇，因此，我得以深入挖掘并向你们展示运用这个出色的工具更聪明更迅速地进行程序调试的各种神奇技巧。WinDBG、Son of Strike (SOS) 扩展及 ADPlus 将是你用来对付仅生产过程中才产生的那些最恶劣 Bug 的工具。在第 6 章中，我尽可能地向你提供了有效使用这些工具所需的指导，即便你之前从未见过它们。

在和来自这个行业各级别的开发者们的合作中，我发现一点——从入门级到炉火纯青级，他们用到的仅仅是 Visual Studio 调试器的冰山一角，并且几乎没有涉足过 WinDBG、SOS 和 ADPlus 中的任何一个。尽管这一观点出自一个程序调试类书籍的作者有些怪异，但我希望你们能尽量远离调试器。当你读本书的时候，会发现我对你们的期望很大程度上并不仅仅是学会如何修复 Bug 和崩溃，而是如何从一开始就避免它们的出现。我同时希望教给你们如何最大程度地利用调试器，因为某一天你会被迫而必须使用它们。

最后一部分“强大的工具”（第 7 章和第 8 章），将探讨扩展 Visual Studio 2005 和构建代码分析/FxCop 规则。在本书之前的版本中，Settings Master 外接程序非常受欢迎，于是我再版的时候将此功能更加发扬光大，以使你仅需点击一个按钮就能将工程设置的变化应用到你所有的工程中。而创建新的设置文件甚至更加简单，因此，对生成版本进行大规模修改的操作是轻而易举的。另外，我提供了一个新的外接程序 Hidden Settings，它列出了 IDE 中未明文标示的设置，以便你修改它们的值。在第 8 章中，我继续延伸了未记载代码分析/FxCop 规则的话题并对比进行介绍，提出了我认为应当作为所有设计准则中一部分的大量规则。

在之前的版本里，我讲述了一些我亲历的有关各种程序调试战役的花边栏。在本版中，我加入了更多你可能会感兴趣的战斗故事。我希望通过与你们分享这些我曾经帮助他人解决的绝“好”Bug（有的还是我帮忙制造的！），你将能理解我所推荐的那些方法和技巧的实际用途。同时我也希望能够帮助你们避免重蹈我的覆辙。

系统要求

使用本书需要如下的配置：

- Microsoft Windows 2000 SP4 或之后的版本，如 Microsoft Windows XP Professional Edition、Windows XP Tablet PC Edition、Windows Server 2003、Windows XP Professional x64 Edition、Windows Server 2003 x64 Edition 或 Windows Vista。本书的源代码中没有本机代码，所以应该能够在 Windows XP Professional IA64 Edition 和 Windows Server 2003 IA64 Edition 中运行，不过我没有条件在基于 Itanium 的机子上测试。
- 为了能够使用全部的功能，你将需要 Microsoft Visual Studio 2005 Team Developer Edition 或 Microsoft Visual Studio 2005 Team Suite。如果只有 Visual Studio 2005 Professional Edition，你将可以生成这些实用工具，但无法生成或运行任何测试代码。Visual Studio Express Edition 或许可以生成代码，不过我没有对此进行过测试。还有，因为 Visual Studio Express Edition 不支持任何扩展性，它们将无法使用第 7 章中的宏和外接程序。