

BOOKS FOR PROFESSIONALS BY PROFESSIONALS™

Comprehensive VB.NET Debugging

Complete for
Visual
Studio
2003

VB.NET 调试全攻略

- 全面描述面向 Visual Basic 与 Visual Basic .NET 开发人员的功能强大的调试技术。
- 讲述从 Windows Forms 到 ASP.NET，再到 XML Web 服务的应用程序类型。
- 讨论了有深度但又易于理解的调试场景，包括多线程、继承及资源管理等。

(美) Mark Pearce 著
谢 俊 尹浩琼 译



清华大学出版社

VB.NET 调试全攻略

(美) Mark Pearce 著

谢俊 尹浩琼 译

清华大学出版社

北 京

内 容 简 介

本书系统全面地介绍了各种功能强大的调试技术,包括所有重要的调试工具和策略,以及 Windows Forms、ASP.NET、Web 服务、Windows 服务和 SQL Server 等多种应用程序的调试方法,最后还深入浅出地讨论了多线程、继承和资源管理等调试场景。

本书适合不熟悉调试技术或者希望全面掌握调试技术的 Visual Basic 与 Visual Basic .NET 开发人员阅读。

EISBN: 1-59059-050-3

Comprehensive VB.NET Debugging

Mark Pearce

Original English language edition published by Apress L. P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA. Copyright ©2003 by Apress L.P. Simplified Chinese-Language edition copyright ©2004 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司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: 01-2003-7371

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

VB.NET 调试全攻略/(美)皮尔斯(Pearce, M.)著;谢俊,尹浩琼 译. —北京:清华大学出版社, 2004

书名原文: Comprehensive VB.NET Debugging

ISBN 7-302-08400-9

I. V… II. ①皮…②谢…③尹… III. BASIC 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2004) 第 026833 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦
http://www.tup.com.cn 邮 编: 100084
社 总 机: 010-62770175 客户服务: 010-62776969

组稿编辑: 曹康

文稿编辑: 王军

封面设计: 康博

版式设计: 康博

印 刷 者: 北京密云胶印厂

装 订 者: 北京国马印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 23 字数: 588 千字

版 次: 2004 年 5 月第 1 版 2004 年 5 月第 1 次印刷

书 号: ISBN 7-302-08400-9/TP·6039

印 数: 1~4000

定 价: 45.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话: (010)62770175-3103 或 (010)62795704

前 言

本书主要讲述在使用 Visual Basic (VB).NET 创建桌面、网络和 Web 应用程序时,如何找出、理解、修复以及(最好是)预防 bug。书中探讨了新的跨语言及跨组件调试工具的强大功能,并说明了如何深入或横穿整个应用程序,从而在 bug 所在的任何级别上找出它们。

随着 VB .NET 的问世,许多传统的调试规则都已经改变,预示着一场变革风暴即将来临。

时代的变迁

回顾个人计算的黑暗时代(Dark Age),在这期间,人就只是人而已,而代码都是以鲜血为代价编写出来的,创建一个可行而稳定的 Windows 应用程序需要付出大量非常艰苦的劳动。Windows 本身相对仍然不太成熟,而且因为缺乏可以用于生产程序的简单工具使其发展受到抑制。1991 年,Visual Basic 1.0 及其后续产品(以后统称为 VB.Classic)陆续出现,戏剧性地拯救并改变了软件开发的世界。

在软件开发史上,Windows 程序设计第一次变得易于理解,而从不认为自己是开发人员的人们也能够施展必要的手段将他们的软件思想变为现实。部门业务流程被大批量自动化,而且通常不需借助于内部技术部门的力量。当这些“非主流”程序员与重视 VB.Classic 带给 Windows 程序设计的非凡生产力的无数专业开发人员胜利会师之时,结果真是蔚为壮观。Windows 软件应用程序在接下来的 10 年中蓬勃发展,其动力是开发的速度和逐渐壮大的开发人员队伍。有一种相当有名的观点,即开发人员及其无数应用程序的可用性促进了 Windows 本身的广泛采用。VB.Classic 完全有可能成为 Windows 真正的应用程序。

这种程序设计的爆炸结果之一是,许多管理人员甚至开发人员开始认为,很多软件开发流程相对简单,而且 VB.Classic 在幕后所做的所有困难工作意味着应用程序的设计与开发可以变得更加迅捷。结果导致了另一次爆炸的出现——被 bug 肆虐的软件大量涌现。由于自身的不稳定和不可靠,Windows 应用程序已经变得声名狼藉。最终用户对每隔一小时就重启一次然后停止错误的应用程序已经习以为常。

由于对可能出现的 bug 类型缺乏了解,开发人员已经在他们的程序中引入了许多难以解决而且莫名其妙的缺陷。理解并解决个人计算涉及的许多典型问题已经花费了人们近 10 年的时间。

如果着眼于未来,可以预见历史可能重演。VB.Classic 可以简化 Windows 桌面应用程序的创建,而公共语言运行库(common language runtime, CLR)的设计目的则是简化桌面尤其是网络应用程序的创建及部署。这些新的应用程序全部使用了公共类库,可以使用至少一打不同语言中的任意一种或几种来编写。虽然与 VB.Classic 相比,学习曲线要明显困难得多,但是回报也肯定要大得多。可以肯定的是,只要 IDE 向导的魔杖一挥,从 XML Web 服务与操作系统服务到 Web 页与桌面程序的一切都将变为可用。部署将变得和 XCOPY 命令一样简单,而 DLL 地狱也将会只存在于远古的神话中(尽管它可能被策略地狱所取代)。

一个全新的世界在向我们招手，用其闪亮的新技术来引诱我们投入它的怀抱。CLR 及 .NET Framework 在网络应用程序和 Internet 方面的潜力就像当初 VB.Classic 在 Windows 方面的潜力一样深不可测。一轮关于富有趣味性和革新性的分布式应用程序的全新浪潮已经蓄势待发。了解网络和网络应用程序的开发人员由此将身价大涨。

面临的挑战

上面说得天花乱坠……现在是接受现实检验的时候了。这种很酷的技术前景面临的几个主要问题之一是，网络资源与本地资源的透明是不同的。网络资源一般不会与本地机器上的资源具有同样的可用性、延迟以及可靠性。编写软件时必须将所有这些因素考虑在内，使用 .NET 提供的一些非常方便的编程接口可以隐藏这些烦琐的细节，但是这样做甚至会使这些问题变得更加糟糕。此外，如果将应用程序放到雷阵雨定期肆虐的 Internet 世界中，那么它下沉的速度将会比 Titanic(泰坦尼克号)还要快。

这种技术前景的另一个主要困难是，.NET 的默认安全模型只给从本地 Intranet 下载的代码授予最低的权限，而给基于 Internet 的代码授予的权限就更少了。如果没有从总体上很好地理解这个安全模型及改变安全策略的含义，调试安全与权限问题很快就会变成一个噩梦。

所以，VB.NET 使得开发人员构建软件系统，尤其是网络应用程序与 Web 应用程序，比过去容易很多。然而，它同时也引入了很多产生 bug 的新途径。这给开发人员带来了令人畏惧的挑战，即尝试构建更加复杂的系统，而且使该系统在出错的时候仍然很容易诊断和修复。如果不小心，使我们的软件应用程序变得越分布式化、越复杂，我们自掘坟墓的可能性就越大。

规模更大的系统，尤其是分布式应用程序，必须使用与小型系统不同的方法进行设计与调试。我们必须吃一堑才能长一智。以防火墙、缓存、加速器、转换器、网关以及通信基础设施的形式引入中间件元素，意味着软件应用程序必须在其周围环境合作的过程中克服几道障碍。软件需求的复杂度似乎与日俱增，而可靠性、可用性、安全性和集成性变得对业务更加关键。

“通过钝(blunt)对象进行程序设计”的思维课程在使用 VB.Classic 的时代通常占据主导地位，但是现在已经远远不够。VB.NET 的规则已经改变，而且为了补偿，交付可靠且缺陷低的软件的过程必须相应地改变。如果开发人员不了解这一点，那么天上的乌云将变得更加浓密，更加诡异，最终 bug 将会像雨点一样落在应用程序上。技术的进展只会使我们倒退的脚步加快。

本书旨在讲述与在有趣的 VB.NET 新世界中设计与调试软件相关的陷阱及危险。无疑，它并不是对语言或框架的一种批评。每项技术都有给开始使用这项技术的开发人员带来问题的方面。不应该让这些危险的因素蒙蔽了您的双眼，以至于看不到这种新语言及其类库潜在的好处。如果能够掌握并驾驭乌云，就可以体验到功能强大、效率高而且配备有一些可以在整个应用程序中使用的优秀调试工具的开发环境带来的好处。

本书内容安排

本书分为 4 个部分，共 15 章。建议首先阅读开始的 4 章，因为这 4 章中包含了对本书其余

部分所涉及到的内容的全面介绍。除此之外，建议应该能够自如地阅读每章，而不必参考前面或后面的内容。每一章的最后都有一个篇幅不长的“插曲”部分，讲述的是调试世界中一些有趣的故事。

第 I 部分：VB.NET 环境中的调试

这个部分首先站在战略的高度上概述了 .NET 环境中的调试工作，然后更加详细地讲述了 VB.NET 中功能强大的新语言特性如何使您犯错误。

- 第 1 章讨论了 VB.NET 及新的 .NET 环境战略上的调试难点。演示了一些需要了解的挑战与危险，从而为防止缺陷打下坚实的战略基础。
- 第 2 章介绍了一些由于不小心引起的讨厌的语言意外。表面上，VB.NET 与 VB.Classic 有一些相似之处，而这些相似之处会蒙蔽开发人员，使他低估了转向 VB.NET 语言的危险性。

第 II 部分：调试工具

在研究一些其他有用的调试工具之前，这一部分较详细地讲述了 Visual Studio .NET 调试器。最后以检查 VB.NET 的跟踪与检测工具结束。

- 第 3 章全面介绍了 Visual Studio .NET 调试器的功能。
- 第 4 章更加详细地讲述了 Visual Studio .NET 调试器，包括对各种调试器的设置及它更加高级的功能。
- 第 5 章研究了一些有用的调试工具，比如其他的 .NET 调试器、Ildasm 实用程序和 Performance Monitor。
- 第 6 章详细讲述了桌面及网络 .NET 应用程序的跟踪与检测。

第 III 部分：调试应用程序

这一部分说明了如何对每一种最常见的 VB.NET 应用程序类型配置并实现调试。还讨论了一些高级的 VB.NET 调试技术。

- 第 7 章研究了 Windows Forms 应用程序的调试，包括类和控件库的调试。
- 第 8 章研究了 XML Web 服务的调试。
- 第 9 章研究了 ASP.NET 应用程序的调试。
- 第 10 章研究了 Windows 服务的调试。
- 第 11 章研究了如何使用 VB.Classic 进行跨语言调试。
- 第 12 章研究了如何使用 SQL Server 进行跨语言调试。

第 IV 部分：调试常见情况

这一部分广泛讨论了 VB.NET 应用程序中的错误处理机制，并且还分析了多线程及分布式应用程序的调试。

- 第 13 章研究了 VB.NET 的异常机制和错误处理工具，并且说明了如何有效地保护应用程序。

- 第 14 章研究了多线程应用程序的调试，重点强调应通过完善的设计来预防多线程 bug。
- 第 15 章研究了分布式应用程序的调试与监控，重点强调远程调试、调试远程处理以及提供可靠的错误诊断方法。

本书面向的对象

本书假定的读者对象是正在使用或者已经长时间使用 VB.NET 的中高级 Visual Basic 开发人员。本书同样适用于正在学习高级程序设计课程的学生和正在寻求在现实世界中掌握调试技能的刚刚毕业的学生。开发团队领导者和非技术管理人员应该发现，本书对于理解正在埋头苦干的开发人员所面临的可靠性及性能问题非常有价值。

本书还假定读者以前对 VB.NET 语言、CLR 以及 .NET Framework 有一些认识 and 了解。本书不适合用作这些主题的入门书。尽管我经常讨论某个特定功能的微妙之处，但您可能(可能不)希望了解如何使用它，您应该广泛阅读文档并进行实验，从而掌握所讨论的功能。尽管可以为每个主题提供更多的介绍，但是这将导致本书的厚度变为现在的 3 倍！所以，我们将重点放在如何避免与调试问题上，而对功能的介绍相对简短，只在确实有必要的时候才进行详细介绍。

最后一个假定是开发人员要负责修复自己的 bug，最好是在这些 bug 到达测试人员之前，否则也一定要在软件投入生产之前修复 bug。如果您认为应该由测试人员或最终用户负责找出您的一些 bug，那么希望本书能够改变您的想法，或者至少向您展示了截然不同的观点。

源代码

本书包含了许多示例程序。可以从 Apress Web 站点的 Download 部分下载它们，站点的 URL 是 <http://www.apress.com>。每一章的源代码以 ZIP 文件的形式提供，每个示例程序位于各自的 ZIP 文件中。

本书中列出的代码都使用最新的 Visual Studio 2003 的 Release Candidate 和 .NET Framework 的 1.1 版测试过，而且还使用 Visual Studio 2002 和 1.0 版本的 .NET Framework 进行过测试。Apress 站点上可用的代码还将另外使用 Visual Studio 2003 和 .NET 1.1 的发布版本进行测试，而且将包括最新的 .NET 服务包在本书出版之后进行的修订以及改动。由于本书的编写和出版需要一定的时间，所以这个方法很有必要。

目 录

第 I 部分 VB.NET 环境中的调试

第 1 章 战略性调试问题	3
1.1 应用程序的可靠性	3
1.1.1 理解可靠性	4
1.1.2 可靠性的衡量	5
1.1.3 软件的可靠性设计	6
1.1.4 改善软件的可靠性	7
1.2 应用程序的可用性	8
1.2.1 理解可用性	8
1.2.2 可用性的衡量	9
1.2.3 设计软件的可用性	9
1.2.4 改善软件的可用性	10
1.3 调试复杂系统	10
1.3.1 根据合同构建	11
1.3.2 理解通信问题	12
1.3.3 可能的解决方案	13
1.4 调试开发人员心理学	13
1.4.1 没有了 Edit 和 Continue	13
1.4.2 心理因素	17
1.5 小结	20
1.6 轶闻趣事	20
第 2 章 VB.NET 语言中的新设计	23
2.1 VB.NET 中 True 的实际值	23
2.2 类成员重载	24
2.2.1 不确定的重载	24
2.2.2 C#重载及 VB.NET 重载	25
2.2.3 让 C#开发人员为难的重载	27
2.3 理解继承问题	29
2.3.1 偶然的屏蔽	29
2.3.2 更多的屏蔽问题	30
2.3.3 理解等价性	32

2.3.4	更好的等价性	35
2.3.5	继承和方法的可见性	35
2.3.6	遍历继承树	37
2.4	其他各种问题	39
2.4.1	开发人员和编译器之间的误解	39
2.4.2	VB.NET 与 C# 之间的混淆	40
2.4.3	装箱的危险	42
2.4.4	数字不再是数字的情况	43
2.4.5	关于 NaN 的其他问题	47
2.4.5	关于 Double	49
2.4.6	有关 Double 的问题	49
2.5	小结	50
2.6	轶闻趣事	51

第 II 部分 调 试 工 具

第 3 章	Visual Studio .NET 调试器	55
3.1	Visual Studio 调试器简介	55
3.1.1	统一用户界面	55
3.1.2	各种调试模式	56
3.1.3	分布式调试	56
3.1.4	高级断点	56
3.1.5	应用程序可显示的信息	56
3.1.6	远程调试	57
3.1.7	低级访问	57
3.1.8	调试器自动化	57
3.2	Visual Studio 调试器的工作机制	57
3.2.1	理解 DebuggableAttribute 类	58
3.2.2	VB.NET 语言编译器的作用	59
3.2.3	JIT 编译器的作用	59
3.2.4	Visual Studio 调试器的作用	59
3.2.5	生成配置及其对调试器的影响	59
3.3	3 种调试模式	62
3.3.1	IDE 调试	63
3.3.2	进程调试	63
3.3.3	JIT 调试	66
3.4	使用 Visual Studio 调试器 IDE	68
3.4.1	使用 Immediate/Command 窗口	68

3.4.2	使用 Output 窗口	70
3.4.3	使用 Source 窗口	71
3.4.4	使用断点和 Breakpoints 窗口	72
3.4.5	使用 Watch 窗口	77
3.4.6	使用 QuickWatch 窗口	78
3.4.7	使用 Locals 窗口	79
3.4.8	使用 Autos 窗口	80
3.4.9	使用 Me 窗口	80
3.4.10	使用 Call Stack 窗口	81
3.4.11	使用 Modules 窗口	82
3.4.12	使用 Disassembly 窗口	83
3.4.13	使用 Threads 窗口	84
3.4.14	使用 Running Documents 窗口	84
3.5	Visual Studio 调试器的 IDE 问题	84
3.5.1	窗口行为	84
3.5.2	解释 Chr(0)	85
3.5.3	调试器的单步调试	86
3.6	小结	86
3.7	轶闻趣事	86
第 4 章	使用 Visual Studio .NET 调试器	88
4.1	Visual Studio 的准备工作	88
4.1.1	General 调试选项	88
4.1.2	Edit and Continue 选项	90
4.1.3	Just-In-Time 调试选项	90
4.1.4	Native 调试选项	91
4.1.5	Project 默认选项(只适用于 Visual Studio 2003)	91
4.1.6	其他有用的 Visual Studio 设置	92
4.2	解决方案的准备工作	92
4.2.1	源文件的搜索路径	92
4.2.2	符号文件的搜索路径	93
4.2.3	解决方案的生成配置	94
4.3	项目准备	94
4.3.1	项目的 Build 选项	95
4.3.2	项目的 Start 选项	95
4.3.3	项目的 Optimization 选项	96
4.3.4	项目的生成配置	97
4.4	设置活动调试器进程	98
4.5	调试生产应用程序	99

4.6	调试符号管理	99
4.6.1	应用程序的符号	99
4.6.2	Windows 操作系统的符号	99
4.6.3	Framework SDK 的符号	100
4.6.4	符号服务器	100
4.7	改进 JIT 调试	101
4.8	处理没有 Edit and Continue 的情况	102
4.9	调试公共中间语言	102
4.10	理解代码优化	103
4.10.1	优化是敌人	105
4.10.2	代码优化测试	105
4.11	小结	106
4.12	轶闻趣事	107
第 5 章	其他调试工具	108
5.1	Cordbg: 控制台调试器	108
5.1.1	使用 Cordbg 的理由	108
5.1.2	使用 Cordbg	108
5.2	Dbgclr: GUI 调试器	110
5.3	Ildasm: 查看 CIL	111
5.3.1	Visual Studio 中的 Ildasm	111
5.3.2	从命令行调用 Ildasm	111
5.3.3	研究 Ildasm 代码	112
5.4	性能监视器: 查看性能信息	118
5.5	ADepends: 查看程序集依赖关系	121
5.6	PermView: 查看程序集权限	122
5.6.1	Visual Studio 中的 PermView	122
5.6.2	从命令行调用 PermView	123
5.6.3	研究权限	123
5.7	小结	127
5.8	轶闻趣事	127
第 6 章	跟踪和检测	130
6.1	有用的诊断信息	130
6.1.1	应用程序的最终用户	130
6.1.2	应用程序的支持团队	131
6.1.3	应用程序的开发团队	131
6.1.4	诊断的类别	131
6.1.5	设计建议	132

6.2	VB.NET 跟踪	132
6.2.1	第 1 步: 创建跟踪	133
6.2.2	第 2 步: 激活跟踪	136
6.2.3	第 3 步: 监听跟踪	137
6.2.4	第 4 步: 编译时的跟踪控制	144
6.2.5	第 5 步: 运行时的跟踪控制	148
6.2.6	有关跟踪的小结	150
6.3	使用 Debug 类	151
6.4	使用定制的性能计数器	152
6.4.1	创建定制的性能计数器	152
6.4.2	操纵性能计数器的实例	153
6.5	小结	154
6.6	轶闻趣事	154

第III部分 调试应用程序

第 7 章	Windows Forms 调试	159
7.1	调试 Windows Forms 应用程序	159
7.1.1	生成应用程序	160
7.1.2	中断至应用程序	160
7.1.3	命中断点	161
7.1.4	考查程序状态	161
7.1.5	查找 Bug	163
7.1.6	棘手的调试情形	164
7.2	调试其他的 Windows Forms 应用程序	166
7.2.1	调试类库	166
7.2.2	调试 Windows Forms 控件	167
7.2.3	调试 Visual Studio 插件	169
7.2.4	调试控件设计器	170
7.3	小结	170
7.4	轶闻趣事	170
第 8 章	Web 服务调试	172
8.1	调试 ASP.NET 工作进程	172
8.2	调试准备	173
8.3	使用 Web 浏览器进行调试	174
8.4	使用 SOAP 客户端进行调试	177
8.5	调试客户端代理	179
8.6	调试远程 Web 服务	180

8.7	调试已部署的 Web 服务	180
8.8	处理 Web 服务故障	181
8.8.1	理解 Web 服务的异常信息流	181
8.8.2	改进 Web 服务异常	182
8.8.3	隐藏异常细节	185
8.8.4	诱捕未经处理的异常	187
8.9	跟踪 SOAP 消息	188
8.9.1	构造 SoapMonitor 的 SOAP 扩展	188
8.9.2	创建 SoapMonitor 属性	192
8.9.3	全部组合起来	193
8.9.4	客户端的 SOAP 记录	195
8.10	跟踪和检测	197
8.11	小结	197
8.12	轶闻趣事	197
第 9 章	ASP.NET 调试	199
9.1	IIS 调试考虑事项	199
9.1.1	使用 IIS 5.x 进行同时调试	199
9.1.2	使用 IIS 6.0 进行同时调试	200
9.1.3	IIS 5.x 的进程回收	200
9.1.4	IIS 6.0 中的进程回收	201
9.1.5	URLScan、IIS 和 DEBUG 谓词	201
9.1.6	IIS 身份验证	202
9.1.7	多线程考虑事项	203
9.1.8	生产调试	203
9.2	调试准备	203
9.2.1	用户账户权限	203
9.2.2	远程服务器调试	205
9.2.3	IE 的配置	205
9.2.4	应用程序配置	206
9.3	调试 AspNetDebugDemo 应用程序	208
9.3.1	ASP.NET 错误处理	210
9.3.2	不进行错误处理	210
9.3.3	程序级的错误处理	211
9.3.4	页面级错误处理	212
9.3.5	应用程序级错误处理	213
9.4	使用 ASP.NET 跟踪	215
9.4.1	页面级跟踪	215
9.4.2	应用程序级跟踪	217

9.5 小结	219
9.6 轶闻趣事	220
第 10 章 Windows 服务调试	221
10.1 调试 ServiceAdmin 服务	221
10.1.1 调试准备	221
10.1.2 使用进程附加进行调试	223
10.1.3 从 Visual Studio 附加	223
10.1.4 确保调试设置的正确性	224
10.1.5 调试不需要安装的服务	224
10.1.6 调试 OnStart 方法	225
10.2 调试安装问题	226
10.3 注意事项	227
10.4 小结	227
10.5 轶闻趣事	228
第 11 章 VB.Classic 应用程序调试	229
11.1 VB.Classic 版本	229
11.2 托管代码和非托管代码	229
11.3 使用 VB 6.0 组件的 VB.NET 应用程序	230
11.3.1 VB.Classic 的准备工作	231
11.3.2 VB.NET 的准备工作	232
11.3.3 调试第一个 COM Interop 应用程序	233
11.3.4 调试技巧	233
11.3.5 使用 VB 6.0 调试器	234
11.4 使用 VB.NET 组件的 VB 6.0 应用程序	235
11.4.1 VB.NET 的准备工作	235
11.4.2 VB.Classic 的准备工作	236
11.4.3 调试第二个 COM Interop 应用程序	237
11.5 克服 COM 版本控制问题	238
11.6 小结	241
11.7 轶闻趣事	241
第 12 章 SQL Server 调试	243
12.1 调试需求	243
12.2 调试组件的安装	244
12.3 权限和 sp_ssidebug	244
12.4 远程调试权限	245
12.5 使用 Visual Studio 直接进行调试	246
12.6 使用 Query Analyzer 直接进行调试	248

12.7	应用程序调试	248
12.7.1	应用程序调试安装	249
12.7.2	从 Visual Studio 中进行应用程序调试	250
12.7.3	从 Visual Studio 之外进行应用程序调试	252
12.8	SQL 调试限制	253
12.9	理解 SQL 安全性	254
12.10	小结	255
12.11	轶闻趣事	255

第IV部分 调试常见情况

第 13 章	错误处理和异常管理	261
13.1	异常与错误	261
13.2	错误处理需求	262
13.2.1	最终用户需求	262
13.2.2	操作支持需求	263
13.2.3	开发人员需求	264
13.2.4	对异常进行日志记录	265
13.3	异常和异常管理	266
13.3.1	System.Exception 类	266
13.3.2	Try...Catch...Finally	267
13.3.3	Try...Catch...Finally 机制	268
13.3.4	正确使用 Try...Catch...Finally	269
13.3.5	构建定制的异常	279
13.3.6	调试异常	286
13.3.7	处理未处理异常	289
13.3.8	异常管理应用程序块	299
13.3.9	分析异常行为	300
13.3.10	异常示例解决方案	301
13.4	小结	301
13.5	轶闻趣事	302
第 14 章	调试多线程应用程序	305
14.1	多线程基础	305
14.1.1	为什么多线程如此难	306
14.1.2	多线程的优点	307
14.1.3	多线程的缺点	307
14.2	多线程问题	308
14.2.1	理解数据争用	309

14.2.2	理解进程死锁	313
14.2.3	理解进程活锁	318
14.2.4	理解线程饿死	319
14.3	ThreadMonitor 应用程序	322
14.4	Windows Forms 中的多线程	326
14.5	处理线程失败	330
14.5.1	处理线程异常	330
14.5.2	终止托管线程	331
14.6	小结	332
14.7	轶闻趣事	332
第 15 章	调试分布式系统	334
15.1	理解分布式应用程序	334
15.1.1	处理失败	334
15.1.2	处理状态	337
15.1.3	理解消息的语义	338
15.1.4	处理有漏洞的抽象	338
15.2	远程调试简介	339
15.3	远程调试准备	339
15.3.1	安装远程调试	340
15.3.2	只安装本机远程调试	341
15.3.3	安装完全远程调试	342
15.3.4	远程调试限制	342
15.4	HeartbeatMonitor 应用程序	343
15.4.1	HeartbeatMonitor 简介	343
15.4.2	安装 HeartbeatMonitor	344
15.4.3	调试 HeartbeatMonitor	346
15.5	监控分布式应用程序	348
15.6	小结	350
15.7	轶闻趣事	350

第 I 部分 VB.NET环境中的调试

“历史上，除了手枪和龙舌兰酒之外，就数电脑使人犯的错误最多”。

—— Mitch Radcliffe

- 第 1 章 战略性调试问题
- 第 2 章 VB.NET 语言中的新设计