PROGRAMMER TO PROGRAMMER™

Visual Basic .NET Debugging Handbook

VB.NET
调试技术手册

(美) Jan Narkiewicz
Thiru Thangarathinam 等著
申晓旻 译

清华大学出版社

VB.NET 调试技术手册

(美) Jan Narkiewicz 等著
Thiru Thangarathinam
申晓旻 译

清华大学出版社
北 京

内 容 简 介

在软件开发过程中，调试是一个必不可少的重要内容，它是保证一个程序正确无误的有效手段。为了强化您对程序的调试能力，Visual Studio .NET 和.NET Framework 提供了许多调试工具。本书从配置 Visual Studio .NET 开发环境开始，全面介绍了 Visual Basic .NET 语言下的调试，异常、线程和进程，日志记录和程序化调试器的交互， Web 应用程序的调试，以及一些高级的调试技巧。

本书主要针对的是 Visual Studio 开发人员，但它对使用.NET Framework SDK 所提供的命令行工具的开发人员也很有帮助。

EISBN：1-86100-729-9

Visual Basic.NET Debugging Handbook

Jan Narkiewicz，Thiru Thangarathinam et al.

北京市版权局著作权合同登记号　图字:01-2002-5381

图书在版编目(CIP)数据

VB.NET 调试技术手册/(美)纳克维兹，(美)桑哥拉斯曼等著；申晓旻译. —北京：清华大学出版社，2003

书名原文：Visual Basic.NET Debugging Handbook

ISBN 7-302-07008-3

Ⅰ.V…Ⅱ.①纳…②桑…③申…Ⅲ.BASIC 语言—程序设计—技术手册　Ⅳ.TP312-62

中国版本图书馆 CIP 数据核字(2003)第 070814 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：于平

封面设计：康博

版式设计：康博

印 刷 者：国防工业出版社印刷厂

发 行 者：新华书店总店北京发行所

开　　本：185×260　**印张**：17.25　**字数**：357 千字

版　　次：2003 年 8 月第 1 版　2003 年 8 月第 1 次印刷

书　　号：ISBN 7-302-07008-3/TP・5159

印　　数：1～4000

定　　价：36.00 元

前　言

为了提高工作效率，开发人员必须非常熟悉开发环境及开发工具。Visual Studio 和.NET Framework 提供了许多可以调试错误的工具和选项。本书旨在让您熟悉这些调试工具，告诉您如何使用及在什么地方使用它们。

本书读者对象

本书适合所有的 VB 开发人员。软件开发不可能一蹴而就，对所有代码必须不断进行调试。事实上，很多开发人员在调试代码上花费了大量时间，因此即使是一些小技巧和提示，也可能为您节省很多时间。

本书假设您主要使用 Microsoft 的集成开发环境(如 Visual Studio 的某个版本或 Visual Basic .NET 标准版)来开发 Visual Basic 应用程序。幸运的是，由这些版本的 IDE 提供的调试功能的区别不是很大。

虽然本书主要针对 Visual Studio 开发人员，但它还包含了很多非常有用的信息，对使用.NET Framework SDK 所提供的命令行工具的开发人员很有帮助。例如，本书讲述了.NET Framework 中的 Debug 和 Trace 类。

本书主要内容

本书对 Visual Studio .NET 提供的调试功能进行了详尽的阐述。

- 第 1 章——配置 Visual Studio 开发环境

Visual Studio .NET 是一个可灵活配置的开发环境，本章将详细讲述 Visual Studio .NET 中与调试相关的所有特性，以及如何设置它们以实现高效率的调试。

- 第 2 章——Visual Studio 环境下的调试

本章将讲述 Visual Studio 中的实际调试：设置断点、运行调试程序，以及给断点求值。我们还将讲述可用来查看托管应用程序的各类信息的一系列窗口。

- 第 3 章——异常、线程和进程

初看起来，异常只与已发布的程序代码相关，但它也是我们进行调试操作的得力助手。我们可以配置 Visual Studio，以便令它在抛出某个异常时触发调试器。通过链接的

内部异常，有可能利用一个一直收集相关数据的应用程序来跟踪异常的路径。

多线程应用程序的调试比较麻烦，因而 Visual Studio .NET 提供了多种特性来帮助实现在不同线程之间的导航，并将问题分离开来。最后，本章还要讲述调试运行的进程，这对于调试不能从 Visual Studio 中启动或处于生产环境中的应用程序(如 Windows 服务)非常重要。

- 第 4 章——日志记录和程序化的调试器交互

收集和保存应用程序在各个阶段下的状态的信息对捕获错误非常重要。这种调试方法对于已部署的程序代码非常重要，因为通常都不希望在生产系统上安装开发工具，或者在实际应用的应用程序中附加调试器。但通过使用日志记录技术，即便在限制最严格的环境中，也可以使调试应用程序的工作变得非常简单。

- 第 5 章——调试 Web 应用程序

本章将专门讲述如何调试 ASP.NET 运行库中提供的各种应用程序。我们将介绍如何设置 Visual Studio 环境来调试各种类型的 Web 应用程序，以及如何跟踪 ASP 页面。本章还将讨论如何调试 SQL Server 代码和 Internet Explorer 中提供的 Windows Forms 控件。

- 第 6 章——高级调试

本章介绍了一些实际的调试情况，包括远程应用程序，将 VB.NET 转换为 VB6 代码进行调试，以及将 VB.NET 转换为非托管 C++代码进行调试等操作。这些知识有助于我们了解客户-服务器端的调试、多线程调试，和混合模式的调试。

使用本书的条件

只要使用文本编辑器或者命令行工具(.NET Framework SDK 免费携带的)就可开发 Visual Basic .NET 应用程序，但是严肃的开发人员都应该使用下列任意一种集成开发环境：

- Visual Basic .NET Standard Edition
- Visual Studio .NET Professional Edition
- Visual Studio .NET Enterprise Developer Edition
- Visual Studio .NET Enterprise Architect Edition
- Visual Studio .NET Academic Edition

本书的所有示例适合采用 Visual Studio .NET 的各种版本； 但 Visual Basic .NET 标准版总存在一定的使用限制，也就是说本书中的有些代码不一定适用。其主要限制是：您不能自己开发类库——顾名思义，您只能开发 Visual Basic .NET 代码。

本书中有一个示例涉及到和 VB6 应用程序之间的互操作，如果您愿意从头构建该

示例，那么就需要安装 Visual Basic 6 IDE。

如果需要调试 Web 应用程序和一些远程应用程序，您需要安装带有 Microsoft 的 Internet Information Server(IIS)的操作系统。以下系统均可：Windows XP Professional、Windows 2000(各种版本)，和.NET Server (各种版本)。可以在 Windows XP 家庭版上安装 Visual Studio .NET，但是由于它不支持 IIS 服务，因而不能开发 Web 应用程序和 Web 服务。

目　　录

第1章　配置Visual Studio开发环境

本章概述通过使用 Visual Studio .NET 来增强调试和加快开发进度的工作方式。实际上不需要对应用程序进行调试(例如单步执行、设置断点和查看变量的内容等)就可以达到这样的效果。本章讨论了如何更好地组织工具、管理项目/方案和使用帮助。本章还阐述了一些 Visual Studio 的特征，包括维护一个“to do”列表、标记代码、更好地管理剪贴板，以及如何可以在一个相同大小的屏幕区域查看更多的代码。

另外，本章说明了各个不同版本的 Visual Studio .NET(教学版(Academic)、专业版(Profession)、企业开发人员版(Enterprise Developer)，以及企业级体系结构设计师版(Enterprise Architect))的特征，可以安装什么样版本的.NET，.NET Framework 和 Visual Studio .NET 支持哪些操作系统，以及工具、解决方案和项目配置信息的配置和存储的位置。

1.1　运行库、SDK 和调试器

一些开发人员认为每台可用的机器(包括父母和顾客使用的机器)都应该安装 Visual Studio .NET。其实，开发人员需要知道他们在某台机器上可以利用的调试支持功能取决于该机器上安装的软件包(Visual Studio .NET、.NET Framework SDK 或者只是.NET Framework)。为了更清楚地说明这个问题，考虑图 1-1 所示的可下载软件，这些软件可以到网址 http://msdn.microsoft.com/上进行订购。

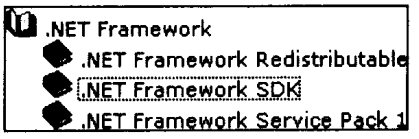

图　1-1

从图 1-1 中可以看到存在不带 Visual Studio .NET 的.NET。可以用来调试应用程序的特定.NET 配置包括：

- .NET Framework Redistributable——这种环境是对应于使用.NET 应用程序的普通用户(非开发人员)的配置情况。通过安装.NET Framework，用户能够运行.NET 应用程序，但他很可能不开发.NET 应用程序(尽管.NET Framework 配备了适用于 C#、VB.NET、Jscript .NET 和 C++的编译器)。用户可能没有安装调试器，而

只是安装了.NET Framework，但是.NET 架构本身仍带有一些调试功能。

- 安装了.NET Framework 软件开发工具包(Software Development Kit，简写 SDK)——.NET Framework SDK 包含.NET 的开发基础构件(如各种 API 和文档等)。SDK 的目的是允许不使用 Visual Studio .NET 的开发商(例如，针对自己的 CPU 特性来开发各自的 JIT 编译器而互相竞争的编译器供应商或销售商)同样可以使用.NET 进行工作和程序开发。SDK 拥有大量的工具，包括一个非常通用的调试器，即 CorDbg.exe。非开发人员的用户很可能不会安装.NET Framework SDK。
- 安装了 Visual Studio .NET——开发人员(并且通常只有开发人员)的机器需要安装 Visual Studio .NET。安装 Visual Studio .NET 的同时将会自动安装.NET Framework 和.NET Framework SDK。这种情况适用于安装任何版本的 Visual Studio .NET(教学版、专业版、企业开发人员版，以及企业级体系结构设计师版)。

这些说明看起来很容易理解，但是开发人员从来不认为他们可以在访问一个顾客站点的同时处理 Visual Studio .NET。作者在一次会议上目睹了一位刚从院校毕业的开发人员正在向某个财富 500 强公司的一位 IT 经理解释，为什么应将 Visual Studio 安装在他们刚生产出的价值 100000 美元的服务器上。财富 500 强的公司保护他们的服务器产品就像 Sanders 上校保护他的肯德鸡配方一样，因此他们(IT 经理而不是 Sanders 上校)不会将未认可的代码放置在他们的高端机器上运行，来帮助某个窘迫的开发人员调试应用程序。在这种环境下部署的应用程序必须要具有内置的不需要调试器的调试技术(例如，使用配置文件来修改应用程序记录的信息以及这种记录输出的目标文件)。

支持安装.NET Framework Redistributable 的操作系统包括：

- Windows 98 第 2 版
- Windows ME
- Windows NT 4.0, Service Pack 6a 或更高版本
- Windows 2000, Service Pack 2 或更高版本
- Windows XP 家庭版
- Windows XP 专业版
- .NET Server (各种形式均可)

以上操作系统均要求至少安装 Internet Explorer 5.01 和 Windows Installer 2.0。Windows 95 和 98 不支持.NET Framework。.NET Framework SDK 和 Visual Studio .NET 可以安装在以下操作系统上：

- Windows 2000, Service Pack 2 或更高版本
- Windows XP 家庭版(有限制)
- Windows XP 专业版
- .NET Server

使用 Visual Studio .NET (或者.NET Framework SDK)，可以开发用于 Windows 98 和

Windows ME 的托管代码，但是，对于这些操作系统而言当前没有调试器支持。本书的后续章节会讨论，当我们调试为 Windows 98 和 Windows ME 开发的应用程序时将面临的一些挑战。但是越来越明显的是，Microsoft 正在逐步表明他今后将废弃 Windows 98 和 Windows ME 操作系统。

1.1.1 Visual Studio .NET 的版本

当前存在多种形式的 Visual Studio .NET。购买的版本不同，软件的功能(包括调试功能)也不同。很可能您得到的是教学版的 Visual Studio .NET 或者只是单语言版本(VB.NET, C# 或者 C++标准版)。这些版本的 Visual Studio .NET 可以较为便宜地购买到，并且非常适合于上夜校或者在职上学的人们。但是我们在此推荐购买全版的 Visual Studio .NET ，而不仅仅是 VB.NET 版，因为这些标准版存在相当大的限制，例如它们不支持 DLL 的开发。

教学版的 Visual Studio .NET 基本上与专业版的 Visual Studio .NET 相同，这些版本包含的功能是相当丰富的。就应用程序而言，可以支持以下类型的应用程序开发：基于 XML 的 Web 服务、Web 应用程序(包含服务器端控件的 ASP.NET)、Windows 应用程序(包含控件的 Windows 窗体)、驻留在 Internet Explorer 中的 Windows 控件、Windows 服务，和各种数据库应用程序。

如果将 Visual Studio .NET 升级到企业开发人员版本，那么将可以获得更好的数据库支持功能，例如，触发器设计、存储过程和用户定义的功能。借助于 Visual SourceSafe，Visual Studio .NET 可以提供源代码控件。从调试的观点看，Visual SourceSafe 是一个天赐之物。 当您正在开发一个应用程序时，可以对该程序设置检查点，并且可以登记不同的版本。如果您不小心犯下了错误或者因为某些原因丢失了代码(硬盘驱动器损坏或者人为错误)，Visual SourceSafe 可以提供一个备份。

Visual Studio .NET 还提供了一个企业级体系结构设计师版本(Enterprise Architect edition)，该版本并不完全是以调试为中心的，它更加重视特定的体系结构和设计。它包含大量的可以用来加快项目设计的企业应用程序模板，另外它还带有 Visio ，可以用来进行软件和数据库建模。它的软件建模特征并不完全与 Rational Rose (一个流行的面向对象开发工具)相同，数据库建模特征也不完全等价于 Erwin (一个流行的数据库开发工具)。但是，企业级体系结构设计师版能够用于大量的面向对象的开发以及数据库开发。

说明：

当安装 Visual Studio .NET 时，应该安装完整的帮助文档而不要通过使用 CD 或 DVD 来访问相关文档。因为，没有什么比由于无法访问 Internet (不能从 msdn.microsoft.com 获

得帮助)，并且将包含帮助文档的 CD 遗忘在家中而坐在计算机旁发呆更让人沮丧。

如果您的机器的硬盘驱动器的空间不够大而不能满足装载完整的 Visual Studio .NET，或许您应该考虑找一份新的工作。

1.1.2 Web 服务器

当开发 Web 服务、ASP.NET 应用程序、ASP.NET 服务器端控件、远程使用 HTTP 的应用程序，以及使用 HTTP 的服务组件时，需要使用 Microsoft 推出的 Web 服务器，即 Internet 信息服务器(Internet Information Server，简写 IIS)。坚持使用 Windows XP 家庭版的开发人员需要知道他们必须舍弃 IIS，而其他任何可以支持 Visual Studio .NET 的操作系统都包含 IIS。需要注意的是，在 Windows XP 或.NET Server 操作系统上，默认情况下 IIS 不是激活的。因此，应该记住在安装 Visual Studio 前启动 IIS。

包含 IIS 并不是选用 Windows XP 专业版操作系统的惟一原因。例如, Windows XP 家庭版不能很好地支持在 Windows 基于角色的安全中使用的 Access 控制列表。

1.1.3 Web Matrix

Microsoft 已经推出了另一个针对 ASP.NET Web 开发的集成开发环境(IDE)，该 IDE 称为 Web Matrix——它可以从网址 http://www.asp.net免费下载。尽管与 Visual Studio 相比，它存在更多的限制，但是它可以用来开发 Web 服务、移动应用程序、ASP.NET 页面，和数据库应用程序。然而，它不能支持 VS.NET 可以支持的许多其他特征，例如 Web 窗体控件的开发。该工具优于 Visual Studio .NET 专业版(相对于企业开发人员和设计师版本)之处在于它可以编辑和修改用于 SQL Server 和 MSDE 的存储过程。此外，它还包含一个可以用来测试 ASP .NET 代码的轻型的 Web 服务器，这意味着使用 Web Matrix 时不需要安装 IIS。

但是，Web Matrix 不能提供比.NET Framework SDK 提供的更多的调试功能。 例如，它不能支持多项目开发和智能识别特征。作为一个工具，Web Matrix 是比较理想的工具，但是缺乏调试支持能力是它的一个明显缺陷。

1.2 配置位置

作为一个工具，Visual Studio .NET 可以管理多种解决方案。一个解决方案只是一个包含多种项目的容器——这些项目可以是控制台应用程序、Windows 窗体应用程序、Web 服务，以及 Windows 服务等。 解决方案的主要职责之一是维护项目之间的相关性

和规定解决方案内的各个项目的构建顺序。当配置和使用 Visual Studio .NET 支持的调试特征时，区分以下的调试特征是非常重要的：

- 针对一个运行 Visual Studio .NET 的特定用户的配置
- 针对一个特定解决方案的配置
- 针对一个解决方案下的特定用户的配置
- 针对一个特定项目的配置
- 针对一个项目下特定用户的配置

1.2.1　Visual Studio .NET 配置

通过使用 Tools | Options 菜单，可以拥有多种配置 Visual Studio .NET 的选择。在此，我们将说明如何使用选项来更好地促进调试。每用户 Visual Studio .NET 的配置信息存储在 Windows 注册表中(如图 1-2 所示)，这意味着这种定制配置对于工作在其他人的工作站上的开发人员不可用，甚至一个以不同的用户身份登录的开发人员也不能使用这种定制。

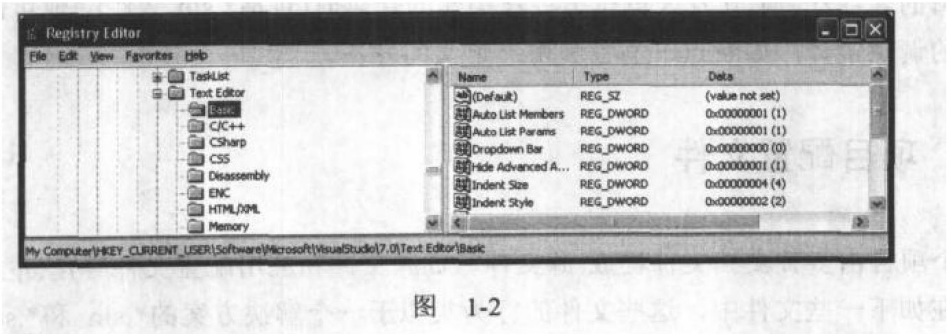

图　1-2

图 1-2 给出了使用 RegEdit(一种 Windows 实用程序)查看的 Windows 注册表的内容，在这里具体是指与当前登录的用户相关联的注册表键(HKEY_CURRENT_USER)。被检查的键值用于 Visual Studio .NET，并且它配置了与文本编辑器(即 Text Editor\Basic)相关的基本设置。Insert Tabs 键值被设为零，这意味着在源编辑器中使用空格而不使用制表符。更重要的是，这一设置以及其他使用 Tools | Options 配置的设置不会自动地被配置，因为使用安装了 Visual Studio .NET 的特定工作站的开发人员必须分别建立(定制)Visual Studio .NET 的每个实例。

注意：

每用户(per-user)设置正在逐渐得到重视，因为大量的开发人员都认为 Tools | Options 只不过是项目的一部分。作者曾经与三位工程师共同开发一个项目，这三位工程师被认为是面向对象的大师，他们在各自的领域都是专家。他们在此项目上花费