

PROGRAMMER TO PROGRAMMER™



Understanding the .NET Framework

Written and tested for final release of .NET v1.0

.NET Framework

编程思想

Kent Tegels Tony Baer 等著

袁勤勇 窦蓉 马誉贤 等译



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



.NET Framework 编程思想

Kent Tegels Tony Baer 等著

袁勤勇 窦蓉 马誉贤 等译

清华大学出版社

(京)新登字 158 号

北京市版权局著作权合同登记号: 01-2002-3203

内 容 简 介

.NET Framework 是 Microsoft 最新推出的 .NET 应用程序开发平台,它为开发人员开发应用程序提供了巨大的便利性。本书首先介绍了有关 .NET 和 .NET Framework 的基础知识,然后介绍了利用 .NET Framework 提供的开发环境开发 Windows 应用程序、Web 应用程序、处理数据及创建 Web 服务等知识,通过对这些内容的介绍,深入阐述了 .NET Framework 编程思想。

本书非常适合于 C++、VB 或 Java 程序员阅读。由于本书遵循循序渐进的原则介绍 .NET Framework 的编程知识,所以对于想了解并深入掌握 .NET Framework 编程思想的初学者也有很高的参考价值。

Kent Tegals, Tony Baer et al: Understanding the .NET Framework

EISBN: 1-861007-09-4

Copyright© 2002 by Wrox Press Ltd.

Authorized translation from the English language edition published by Wrox Press Ltd.

All rights reserved. For sale in the People's Republic of China only.

Chinese simplified language edition published by Tsinghua University Press.

本书中文简体字版由英国乐思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。

版权所有,翻印必究。

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

.NET Framework 编程思想/(美)泰格斯等著;袁勤勇等译.——北京:清华大学出版社,2002

书名原文: Understanding the .NET Framework

ISBN 7-302-05886-5

I. N... II. ①泰...②袁... III. 计算机网络—程序设计 IV. TP393

中国版本图书馆 CIP 数据核字(2002)第 071711 号

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 郭东青

印 刷 者: 北京通州大中印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印 张: 15.75 字 数: 373 千字

版 次: 2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

书 号: ISBN 7-302-05886-5/TP·3492

印 数: 0001~5000

定 价: 35.00 元

出版者的话

近年来，国内计算机类图书出版业得到了空前的发展，面向初级用户的应用类软件图书铺天盖地，但是真正有深度和内涵的高端图书不多。已经掌握计算机和网络基础知识的人们，尤其是 IT 专业人士迫切需要“阳春白雪”。IT 图书市场呼唤精品！

为了满足这种市场需求，清华大学出版社从世界出版业知名品牌 Wrox 出版公司引进了受到无数 IT 专业人士青睐，被奉为 IT 出版界经典之作的 Professional 系列丛书。这套讲述最新编程技术与开发环境的高级编程丛书，从头到尾都贯穿了 Wrox 出版公司“由程序员为程序员而著(Programmer to Programmer)”的出版理念，每一本书无不是出自软件大师之手。实际上，Wrox 公司的图书作者都是世界顶级 IT 公司(如 Microsoft, IBM, Oracle 以及 HP 等)的资深程序员，他们的作品既深入研究编程机理，传授最新编程技术，又站在程序员的角度，指导程序员拓展编程思路，学习实用开发技巧，从而风靡世界各地，被 IT 专业人士和程序员视为职业生涯中的必读之作。

为了保证该系列丛书的质量，清华大学出版社迅速组织了一批位于 IT 开发领域前沿的专家学者进行翻译，经过编辑人员的进一步加工整理后，现陆续奉献给广大读者。

读者可以从 www.wrox.com 网站下载所需的源代码并获得相关的技术支持。同时，也欢迎广大读者参与 p2p.wrox.com 网站上的在线讨论，与世界各地的编程人员交流读书感受和编程体验。

前 言

当 .NET Framework 到来时，在用户能够有效地开发应用程序之前有大量的可以学习和必须理解的知识。更有甚者，不像 Microsoft 的老开发工具的新版本那样，.NET 是一种根本性的改变，以致于大多数开发人员（甚至是经验丰富的 Microsoft 程序员）都不得不从头开始学习。Microsoft 突然执著地把每样东西都打上 .NET 的招牌，这可能潜在地带来巨大的混淆。

本书的目标是澄清所有关于 .NET 的混乱认识。不应该轻易选择使用 .NET，所以本书用来对全部的有关 .NET 的混乱认识提供一个概述，以便您能够较好地理解将要作出的决定。

顺着这个思路，我们希望回答几个基本的问题：

- 严格地说，.NET 是什么
- .NET 会带来什么好处
- 为什么我应该选择 .NET
- 我如何使用 .NET
- 有了 .NET，我能够做什么

在本书的结尾，您应该就对您的职业生涯和公司来说，.NET 是否是一个值得的投资有一个清楚的概念，并且准备好开始学习 .NET 开发。

本书读者对象

本书面向那些在 Java、C++、Visual Basic 或者其他的现代多用途的语言方面有经验的开发人员，他们想弄清 .NET 的真相。本书为 .NET 对您自己商业需要的适合度提供了一种评估技术的概述。它也能满足您的商务需求，使得您能够确定进一步研究 .NET Framework 的哪个领域。

使用本书要求

简单地阅读本书，您将学到许多有关 .NET 的知识。然而如果想要从本书得到大部分知识的话，那就需要开始着手进行一些 .NET 编程。为此，就需要：

- Windows 2000 或 Windows XP



- .NET Framework redistributable 或 SDK。

上列二者都可从 MSDN 网站 <http://msdn.microsoft.com/netframework/prodinfo/getdotnet.asp> 免费下载得到。

本书的部分内容也使用下列产品：

- Visual Studio .NET 1.0。我们已经在 1.0 版本中对代码进行了测试，尽管大部分代码必须在最近的预发布版本中运行。
- SQL Server 2000——尽管我们使用的大多数技术能够适用于任何数据库系统，包括 Access，但最好拥有 SQL Server 2000。

本书主要内容

本书开始把 .NET 作为一个整体介绍，接下来概述 .NET 对开发人员的影响。然后，本书继续介绍一些普通的开发任务，以及应该如何用 .NET Framework 来实现这些任务。最后，对介绍的内容做了总结。

下面是本书各章包括的要点：

第 1 章，“正确认识 .NET”。该章包括：.NET 是什么，为什么要在这里讨论，以及它的发展趋势。我们着眼于 Microsoft 的 .NET 策略的组件，促使 Microsoft 创建 .NET 的环境，以及一些最有可能产生的影响。

第 2 章，“用 .NET 开发”。该章研究了 Framework 基础结构，包括运行环境（公共语言运行环境、通用类型系统和通用语言规范）。本书还会介绍 Visual Studio .NET 开发从 .NET 到部署的一些基础知识。

第 3 章，“Web 开发”。该章着眼于 ASP.NET 为建立动态 Web 站点所作的修改。我们将回顾如何在 CGI 和诸如 ASP 之类的脚本语言基础上进行 Web 开发，然后介绍如何用文本编辑器或者 Visual Studio .NET，来开发您自己的 ASP.NET 页面。

第 4 章，“Windows 客户程序的开发”。该章着眼于 Windows Forms、用于创建 Windows 应用程序的 .NET Framework 类。本章介绍如何用内置的控件创建 Windows 应用程序，然后介绍了怎样扩充和组合控件以产生您自己的控件。本章还介绍了如何用可视化继承在一个应用的几个窗体之间维护公共元素。

第 5 章，“处理数据”。该章探讨不同类型的数据，以及如何用 .NET 对它们进行处理。内容涉及关系数据库、XML 以及平面文件，介绍有关 ADO.NET、.NET XML 类以及数据流。

第 6 章，“传统技术和企业系统”，该章着眼于 .NET 与 DLL、COM 组件和 COM+ 中间层服务的集成。

第 7 章，“Web 服务”，该章介绍开发的一个新领域，Web 服务也是 .NET 本身

起源的一大要素。该章探讨了它们的重要性，它们是如何产生的，以及如何用.NET 来开发。

第 8 章，“.NET 技术的未来”，该章从前面介绍的每个概念中得出结论，然后展望未来。我们着眼于.NET 的未来——它的发展趋势，它会有什么含义。我们也介绍了如何在您的组织内部发展.NET 的技能和实践。

如何从 Web 站点下载代码

本书里的代码一般是短小的摘录，只是为了说明概念，而不是完整的应用程序。但是代码例子可从 Wrox 的 Web 站点下载得到。要想得到代码，请访问 www.wrox.com，并导航到 Understanding the .NET Framework，单击 Code(代码)栏里的 Download(下载)，或者在本书的详细页面上单击 Download Code(下载代码)。

下载的文件是 ZIP 格式。Windows XP 能够自动地识别，但是 Windows 2000 需要诸如 WinZip 或 PKUnzip 之类的解压缩程序。

客户支持

我们想接收到读者来信！想知道读者对本书的看法：喜欢什么，不喜欢什么，以及读者认为下次我们怎样做得更好。请把评价意见发给我们，或发 e-mail 至 feedback@wrox.com。在您的反馈消息里请注明本书的名称。

我们会认真听取这些评价意见，并在将来的书中加以考虑。

勘误表

我们已经尽了很多努力来确保文本和代码方面没有错误。如果您确实发现有错误，比如说拼写错误、错误的代码段或者任何不准确的内容，我们会对您的反馈表示感激。通过以勘误表的形式发送给我们，就可以节省其他读者学习本书的时间，并有助于我们提供更高质量的信息。

把您的评价意见用 e-mail 发至 support@wrox.com。我们将查看您的信息，如果正确的话，该信息就会邮递到该标题的勘误表页面，并将在本书以面的版本里得到采用。

要想找到本书有关标题的勘误表，请去站点 www.wrox.com，并找到 Understanding the .NET Framework 的位置。单击 Book Errata(书的勘误表)这个链接，它在有关本书详细页面的封面图形之下。



E-mail 支持

如果您希望直接向详细了解本书的专家询问本书中出现的问题，那么请发送 e-mail 至 support@wrox.com，在您的 e-mail 的主题字段中写上本书的书名和 ISBN 的最后 4 个数字。请在 e-mail 中包括下面一些内容：

- 在 Subject(主题)字段写上：本书的书名、ISBN 的最后 4 个数字以及问题所在的页号。
- 在消息的主体里写上：您的姓名、联系信息以及所提的问题。

我们不会把垃圾邮件发送给您。为了节省您和我们的时间，我们需要详细资料。当您发送一个 e-mail 消息时，它将经过下面一系列的支持：

- 客户支持——您的消息将被递交到我们的客户支持小组，他们是阅读您的消息的第一批人。他们手上有一些有关最常见问题的档案资料，并且会立即回复有关本书或者 Web 站点的任何一般性的问题。
- 编辑——比较深入的质疑转发到负责该书的技术编辑。编辑们有程序设计语言或特定的软件产品的丰富经验，能够回答有关主题详细的技术问题。
- 作者——如果编辑也不能回答您的问题，那么问题将转发给作者。我们要设法保护作者的写作不被分心，但是我们还是乐意把一些特殊的请求转交给他们。所有的 Wrox 作者都对其作品提供支持。他们将就他们的回答发 e-mail 给客户和编辑，所有的读者都将会受益。

Wrox 支持程序只能对那些直接关系到本书内容的问题提供支持。对超出书本范围的疑问由 <http://p2p.wrox.com>/论坛的团体列表提供支持。

p2p.wrox.com

对作者和同行讨论来说，可以参加 P2P 邮件列表。我们独特的系统提供 programmer to programmer™(程序员对程序员)邮件列表、论坛和新闻组方面的联系，它们全部在一对一的 e-mail 支持系统之外。如果查询 P2P 的话，那么可以确信使用邮件列表的许多 Wrox 作者和业界专家将查看它。在站点 p2p.wrox.com，读者将发现对其有帮助的许多不同的列表，不仅在您读本书的时候，而且在您开发自己的应用程序时都将从中获得帮助。

要想订阅一个邮件列表，应遵循下列步骤：

- (1) 浏览站点 <http://p2p.wrox.com/>。
- (2) 从左边的菜单栏选择适当的类别。
- (3) 单击希望加入的邮件列表。
- (4) 根据指示来订阅，并填入用户的 e-mail 地址和密码。

- (5) 答复所接收到的确认 e-mail。
- (6) 使用订阅管理器来加入更多的列表，并设置用户的 e-mail 参数。

为什么这个系统提供了最好的支持

您可以选择加入邮件列表或者可以将其作为每周文摘来接收。如果您没有时间或者工具来接收邮件列表的话，那么可以搜索我们的在线档案室。垃圾邮件和广告邮件将被删除，独特的 Lyris 系统保护了您的 e-mail 地址。有关加入或离开邮件列表的查询，以及有关邮件列表的其他常见查询，都应该发送至 listsupport@p2p.wrox.com。

目 录

第 1 章 正确认识.NET	1
1.1 历史背景.....	1
1.1.1 .NET 的起源.....	2
1.1.2 Web 服务：概念.....	5
1.2 何为 .NET.....	5
1.2.1 .NET Framework.....	6
1.2.2 .NET 产品.....	8
1.2.3 .NET 服务.....	11
1.2.4 .NET 可提供的內容.....	13
1.3 .NET 与 J2EE.....	15
1.4 对.NET 的挑战.....	18
1.4.1 客户机.....	19
1.4.2 企业.....	20
1.4.3 Web 服务.....	21
1.5 小结.....	23
第 2 章 用.NET 开发	24
2.1 .NET 开发基础.....	25
2.2 多种语言支持.....	31
2.2.1 多种语言支持的缺点.....	35
2.2.2 .NET 语言.....	35
2.2.3 .NET Framework.....	41
2.2.4 公共语言运行环境.....	41
2.2.5 通用类型系统.....	42
2.2.6 公共语言规范.....	44
2.2.7 基类库.....	45
2.2.8 .NET 类库.....	46
2.3 中间语言.....	51
2.4 实时编译.....	53
2.5 垃圾收集.....	54
2.6 程序集.....	57



2.7	小结	60
第 3 章	Web 开发	62
3.1	转变中的 Web 开发	63
3.1.1	开发的速度	63
3.1.2	改善性能	72
3.1.3	容错	75
3.2	ASP 开发如何修改	76
3.2.1	Web Form 前端	76
3.2.2	后台代码类	77
3.2.3	前期或后期绑定	77
3.3	理解 Web Form	78
3.3.1	HTML 标记	78
3.3.2	此页面的后台代码类	82
3.3.3	移动控件	85
3.4	小结	86
第 4 章	Windows 客户程序的开发	87
4.1	Windows Forms: 大图片	88
4.2	Visual Studio .NET 里的 Windows Forms	91
4.3	控件的使用	94
4.3.1	拖动、释放、产生代码; 必要时重复	94
4.3.2	事件	97
4.3.3	进入(消息)循环	100
4.4	滚动您自己的控件	100
4.4.1	理解组件	101
4.4.2	控制您的控件	103
4.5	可视化继承	119
4.6	数据绑定	124
4.7	MDI 应用程序	125
4.8	使用 Web 服务	125
4.9	.NET Compact Framework	126
4.10	小结	126
第 5 章	处理数据	127
5.1	被解释的数据类型	127
5.1.1	关系型数据	127

5.1.2	XML 数据	128
5.1.3	平面文件数据	130
5.2	.NET 里的关系数据	130
5.2.1	理解 ADO.NET	131
5.2.2	使用 ADO.NET	136
5.2.3	ADO.NET 小结	150
5.3	.NET 里的 XML 数据	151
5.3.1	XML 处理模型	151
5.3.2	面向流的 XML I/O	152
5.3.3	文档对象模型	158
5.3.4	XPath	159
5.3.5	转换	160
5.3.6	架构	160
5.3.7	XML 串行化	161
5.3.8	XML 小结	162
5.4	.NET 里的平面文件数据	162
5.5	小结	164
第 6 章	传统技术和企业系统	165
6.1	访问传统函数	166
6.1.1	访问传统结构	167
6.1.2	方法的使用	169
6.2	COM 互操作性	172
6.2.1	COM 互操作性的性能	174
6.2.2	错误处理	175
6.2.3	非托管代码访问 .NET 程序集	176
6.3	服务组件	179
6.3.1	COM+ 服务	179
6.3.2	什么是服务组件	180
6.3.3	.NET 和 COM+ 的互操作性	180
6.3.4	配置和部署服务组件	181
6.3.5	安装	185
6.4	远程处理	186
6.4.1	远程处理客户端实现	189
6.4.2	远程处理现实性和复杂性	190
6.5	小结	191



第 7 章 Web 服务	192
7.1 什么是 Web 服务	192
7.2 创建 Web 服务技术程序块	194
7.2.1 HTTP——传输层	195
7.2.2 SOAP/XML——消息传递层	195
7.2.3 WSDL——接口定义层	197
7.2.4 DISCO 和 UDDI——服务发现层	203
7.3 创建 Web 服务	206
7.3.1 创建一个 CD 目录 Web 服务	207
7.3.2 WebMethodAttribute 属性	213
7.4 创建 Web 服务客户程序	214
7.5 .NET Framework 中的其他重要类	220
7.6 .NET My Services	220
7.7 Web 服务的未来	222
7.8 小结	222
第 8 章 .NET 技术的未来	223
8.1 技术集锦	223
8.1.1 越容易, 越不容易	224
8.1.2 共享组件	225
8.1.3 部署选项	226
8.1.4 代码越少, 性能越好	226
8.1.5 可以开发什么样的应用程序	227
8.1.6 标准的冲击	227
8.1.7 永远不变的内容	228
8.2 对企业的冲击	228
8.2.1 对 VB 程序员的文化冲击	229
8.2.2 应该移植吗	230
8.3 对开发商和顾问的冲击	230
8.4 作为服务的软件: 它到底意味着什么	231
8.4.1 进入 Web 服务	231
8.4.2 对应用程序开发人员的影响	232
8.5 下一步	233

第1章 正确认识.NET

Microsoft .NET Framework 和 Visual Studio .NET 的出现只不过是 Microsoft 寻求大规模计算解决方案的最新篇章。首先,这种潜在的技术提供了一些特性,它们不仅使所有 Microsoft 开发语言成为一体,而且集成了许多来自第三方的工具和语言。就 Visual Studio .NET 而言,它提供了一个适用于所有 Microsoft 语言的标准化开发过程的 IDE。就 .NET Framework 而言,它类似地标准化了以任何兼容的语言编写的 .NET 应用程序执行和部署的方式。另外,.NET 铺就了软件作为服务的道路。

为了利用 .NET Framework 的新功能和多功能性,Windows 开发人员必须在某种程度上“再造自我”。VB 程序员需要采用面向对象的方法,C++开发人员需要习惯于 VB 风格的 RAD 功能和高级类库,而 Java 开发人员需要学会珍惜所拥有的 Microsoft 环境。

对开发人员来说,这是一个机会。而且因为 Microsoft 倾向于广泛地扩充其商标,.NET 潜在地非常混乱。当今 Microsoft 正在给操作系统、服务器、个性化 Web 服务、编程语言、组件模型、集成开发环境以及运行时开发环境打上 .NET 商标,随着时间的推移,.NET 将更加有意义。

那么什么是 .NET,并且为什么都在大肆宣传它?理解 .NET Framework 就是要剖析 .NET Framework 的体系结构,使得您能有一个关于它能提供和满足您的需求的客观评价。但是在告诉您如何利用 Microsoft 开发技术里的最新特性之前,首先要介绍一下 .NET,并且讲述一下其重要性。

特别地,本章通过以下内容来介绍 .NET:

- 提供有关 .NET Framework 出现的历史背景。
- 描述 .NET 实际上是什么以及它提交什么。
- 提供 .NET Framework 与 J2EE 的一个大概的比较,以便可以很显然地选择其一。
- 概述对于 .NET 应用的挑战。

1.1 历史背景

Microsoft 的使命一直是渗透——许多人说是支配——从底向上的企业计算。对于应用程序开发人员来说,随着 Visual Basic 的 1991 版本,即 Windows 的第一个集成开发环境(IDE)的发布,这种改革尤为明显。VB 把开发人员从依靠神秘的文本编辑器编写代



码、依靠图形编辑器开发 GUI 元素，以及依靠独立的 C 编译器产生可执行代码中解放出来。它为开发人员提供了使用一种更加容易学习的语言来开发图形化的、事件驱动应用程序的能力，而这种应用程序通常是用 C 语言开发的。

VB 把代码编辑、图形设计以及实时(JIT)编译器全部组合在一个易于使用的拖放式环境中。对于寻求开发功能强大的企业应用程序的高级开发人员来说，Microsoft 也提供了 Visual C++，几年以后紧接着的是模仿 Java 的语言即 Visual J++。Microsoft 提供的这些语言，既是彼此独立的又作为 Visual Studio 软件包的一部分，它们添加了支持工具，如用于版本控制的 Visual SourceSafe 和用于介绍性的 UML 设计的 Visual Modeler。

正当 Microsoft 着眼于将自己塑造成拥有 NT、SQL Server、Exchange 以及其他服务器端产品的企业时，其开发环境日益成为企业的焦点。一般而言，在某种程度上 VB 的发展与客户-服务器技术的发展并驾齐驱。VB 应用起初是简单的、独立的事件，它提供给大量用户快速学会应用程序开发和开发简单高效的应用程序的能力。

然而，很显然，VB 是 Microsoft 用于主宰世界的计划的一部分。VB 3 添加了数据库访问和对象链接和嵌入(Object Linking and Embedding, OLE)；VB 4 添加了基于对象的类模块；VB 5 改善了本机代码的编译和 Web 友好的 ActiveX 控件；以及 VB 6 添加了丰富的 ODBC 数据库连接和其他“重大的”特性，诸如此类的改进使 Microsoft 的这种企图昭然若揭。

这样做，Microsoft 有助于通过组件对象模型(Component Object Model, COM)体系结构把组件开发技术介绍给众多开发人员。几年之后，COM 及其后续的分布式 COM (Distributed COM, DCOM)逐渐增加了中间层，而中间层正是开发更易于扩展的 Web 和企业应用程序所必需的。

由于有了 .NET Framework, Microsoft 取代了 COM 模型，不仅仅移植了诸如继承和重载之类的面向对象的特性，而且用能够自动执行诸如内存和对象管理之类的日常管理功能的一个模块化环境代替了整个运行环境，减少了版本冲突，并且扩展了对非 Microsoft 语言的支持。虽然 Visual C++可能已经具有其中的许多特性，但对 VB 来说则是新特性。

1.1.1 .NET 的起源

直到 1990 年，Microsoft 主宰了整个 PC 市场。正如我们在上面已经看到其对开发产品的态度，Microsoft 计划通过逐渐提升的方法来主宰服务器市场——开始控制工作组计算，然后是小型商业网络，再往后着眼于更高级的应用。

在 20 世纪 90 年代期间，发生了一些冲击 IT 业界的事情：

● Internet 的出现——人们认识到在服务器上使用平台无关的、非专用的标准，网络能够真正很好地工作。您甚至能够免费获得服务器操作系统！突然间 Microsoft 的桌面系统的优势显得没有价值了。

● 向诸如网络计算机、机顶盒以及 WAP 电话之类的非 PC 设备的迁移。这由 Internet 的倍受欢迎的应用——Web 来驱动。可以在任何客户端平台设置一个 Web 浏览器，然后可以看到驻留在任何服务器平台上的站点。

进入 Java 时代。1995 年，Java 的引入带来了 Web 开发的世界风暴，因为 Java 看上去好像是为 Web 而生的。

Java 具有一种重要的、基于 C 语言的、健壮而有效的“编程思想”，而且具备 Visual Basic 的高层次、快速开发的能力。它能够在任何服务器上运行；也能够任何客户机上运行。此时，Microsoft 来了个 180 度的大转弯开始“拥抱”Internet，而对此缺少一种明确选择的编程语言。事实上，可以有论点认为，出现 .NET Framework 的一个重大原因可以归于 Microsoft 与 Java 反复变化的关系。

Microsoft 的反击获得了一些成功。为了抨击与平台无关的客户机，Microsoft 着手生产最好的 Web 浏览器软件——并且致力于 Windows 平台。Internet 支持开始出现于 Office 和 Windows 中，使得 Windows 成为支持 Web 桌面的平台。然后，ASP (活动服务器页面，Active Server Pages) 作为一种方法通过一个简单的 API (应用程序编程接口) 而不是当时提供的其他工具和语言动态产生 web 内容。

由于 Java 已经出现并得到普及，Microsoft 勉强地从 Sun 取得 Java 的许可协议——将 Visual J++ 添加到稳定的 Visual Studio 语言中。

因为 Microsoft 和 Sun 是同床异梦的合作伙伴，他们的 Java 不能维持共存。Sun 的平台无关的努力与 Microsoft 为 Windows 而优化 Java 的想法发生冲突。Microsoft 对 Java 的包容和扩充，以及用本机 Windows 基础类 (Windows Foundation classes, WFC) 和其他调用增强 Visual J++，使 Windows 成为最快的 Java 部署平台之一，这真是具有讽刺意义。

然而，因为 Sun 拥有知识产权，Sun 就控告 Microsoft 侵犯了其 Java 许可证。Sun 最终在法庭上获胜，迫使 Microsoft 和 Sun 分道扬镳，Visual J++ 走向消亡。Microsoft 因预见到这个分离的判决，而开始忙于开发一个 Java 的替代产品，代码命名为“COOL”。

历史证明，COOL 不仅仅是 Microsoft 增加其本身的 Java 的另一个尝试，而且也是 Microsoft 使应用程序开发与语言而不是平台无关的广泛策略的一个基础。这样做，Microsoft 不是以现代的 COM 或 COM+，而是以颠覆运行环境的一个完全取代而告终。当 Microsoft 于 2000 年 6 月宣布新的 .NET 平台时，其响应的轮廓开始明朗化。由于有了 .NET，Microsoft 设计了一种聪明的方法来用其自己的术语包含异类的世界。



Java 杀手或自然的演变

为了了解 .NET 的起源，关键要知道它准备避开的技术。

Java 的“一次编写，随意运行”的原则起初被解释为：编写那些不用移植就可以运行在任何目标平台上的智能客户程序的能力。Java 的最初成功来自于在运行时在 Internet 之上部署小应用程序，或者通过按传统方式安装的客户端软件。然而，Java 真正发现其常驻的地方是在服务器上。

Java 的成就伴随着 Java 2 平台的 1999 版而得以实现，即 Enterprise Edition (J2EE)，它是包含下列特性的一个框架：组件定义、Web 页面生成以及对象/关系数据映射、到目录服务、事务管理、消息传递和数据库访问的 API。J2EE 迅速获得应用程序服务器供应商的巨大规模的赞同，是他们使 J2EE 变成了 Microsoft 的以 Windows 为中心的分分布式网络体系结构(DNA，其包含 COM+)的一个事实上的替代标准。J2EE 比 Windows DNA 具有更好的可伸缩性，不一定因为其部署体系结构，而是由于其运行于强大的 UNIX 平台上的能力。

注意：

当然，J2EE 不是没有问题。举例来说，非常丰富的框架使得许多企业 IT 小组难于实现它。这个问题综合起来就是这样一个事实，每个应用程序服务器供应商都有不同的部署语义学。诸如 BEA 之类的 J2EE 供应商在引入简化部署的工具。部署是一项任务，一旦应用程序服务器部署由负责批准 Java 标准的实体(Java Community Process, JCP)标准化后就变得比较容易了，这个标准化工作有可能在 2003 年完成。

.NET 的这种体系结构是 Microsoft 对 J2EE 的回答。然而，因为它出现在 Sun 的框架之后，所以 .NET 从其竞争对手的平台的缺点中吸取了教训。虽然我们稍后将探究其细节，但是足以见得 Microsoft 采纳了一些 Java 概念，诸如取消多重继承以及其他几种特性，包括内存管理、垃圾收集以及实时编译程序。然而 .NET 的“王牌”包括了有关 Java 虚拟机的概念，以及一个包含 XML Web 服务直接支持的一个设计。

结果如何？.NET Framework 从开始就包含了 XML 技术，把软件部署简化为 Web 服务把各种数据对象重用于众多应用程序或者目标客户机上。一种强大的公共运行环境处理最小化代码需要的管道函数。至于像虚拟机那样的概念，Microsoft 在处理语言和基础平台之间使用了一种中间层，使得 .NET 与语言、而不是平台无关(虽然一些第三方试图使 .NET 技术适应于像 Linux 那样的其他平台)。由于其 Web 服务的强大支持和多种计算语言的强大支持功能，.NET 提供给 Microsoft 一个以 Windows 为中心的方法来与非 Windows 世界共存。