



.NET 大局观

Understanding .NET
— A Tutorial and Analysis —



Understanding .NET

A Tutorial and Analysis

David Chappell



David Chappell 著
侯捷 荣耀 译

华中科技大学出版社



.NET 大局观

Understanding .NET

David Chappell 著

侯捷 荣耀 译

.NET 大局观

Understanding. NET

David Chappell

Copyright ©2002 Addison Wesley Longman, Inc.

Simplified Chinese Copyright 2003 of the first publication by Huazhong Science and Technology University Press and Pearson Education North Asia Limited.

All rights Reserved.

Published by arrangement with Pearson Education North Asia Limited, a Pearson Education Company.

版权所有,翻印必究。

本书封面贴有华中科技大学出版社(原华中理工大学出版社)激光防伪标签,封底贴有“Pearson Education”激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

.NET 大局观/(美)David Chappell 著;侯捷 荣耀 译

武汉:华中科技大学出版社,2003年7月

ISBN 7-5609-2949-4

I. N...

II. ①D... ②侯...

III. 计算机网络-程序设计

IV. TP393.09

责任编辑:周筠

出版发行:华中科技大学出版社 (武昌喻家山 邮编:430074)

<http://press.hust.edu.cn>

录排:华中科技大学惠友科技文印中心

印刷:湖北新华印务有限公司

开本:787×1092 1/16 印张:22.75 插页:2 字数:450 000

版次:2003年7月第1版 印次:2003年7月第1次印刷

印数:1—8 000 定价:42.00元

ISBN 7-5609-2949-4/TP·506

侯捷 译序

我把这本中译书命名为《.NET 大局观》。虽然这个名称并没有与原书名《Understanding .NET》直接对应，却清楚地说明了书籍的特性和定位。

是的，.NET 是一个如此庞大的架构，上至商务模型、软件技术、类库，下至编程语言、型别系统（types system），乃至更底层的运行系统、虚拟机……全包揽了。要了解这么庞大的机制和技术，非一蹴可成，但许多不同领域的从业人员基于种种因素，需要认识 .NET，理解 .NET，这时候他们需要巨大的 .NET 不论就其上层应用或其底层技术先有一个通盘认识。这时候，他们需要一本“大局观”。

我也是“亟欲通盘认识 .NET”的这群人中之一个。我的时间紧凑，我的专长在软件技术和编程技术；我不但希望在短时间内抓住 .NET 的技术全貌，也希望在短时间内抓住 .NET 的应用全貌。自从知道 David Chappell 写了这部作品，我就放心了。我知道我找得到我要的东西。Chappell 曾经写过同类型书籍，包括《Understanding ActiveX and OLE》和《Understanding Microsoft Windows 2000 Distributed Services》，都有很

好的口碑。就我的观点，Chappell 是书写这种大局观、全览式、图文并茂、观点纷呈的书籍的第一人选。虽然，他的焦点只在微软相关领域，但是公平地说，那已是涵盖整个软件领域的大半版图了。此外，我也必须提醒你，Chappell 的众多观点对微软带有明显的善意©。

关于这本书，我的经验是，即使囫圇吞枣，以三两天时间浏览一遍，也能够对 .NET 全貌获得一个不错的梗概。阅读它并非全无门槛，我毕竟在 Windows 环境下以技术谋生多年，对相关各领域都有或多或少或精或浅的认识。如果你对本书所涉一切都很陌生，可能你无法接受这本书。

写这种书很困难，需要既广泛又深入的知识。读这种书很容易，只要具备一些常识和（或）概念。我们不会因为这种书很难写而买它，我们因为这种书很容易读所以掏钱！容易读？呃，“读者基础”从来都是一个说不清楚、难以量化的东西。不妨试读一二，如果满意，它就适合你。当然，你也可以看看 Chappell 在本书前言贡献的一段“本书为谁而写”。

对这个译本，我为非常多的术语和专有名词提供了中英并陈的表现方式，频率之高不厌其烦。我也保留了许多全英文术语和专有名词。这么做有多重用意，一是为了让读者熟悉国际通用的术语和名称——你早晚要接触它；二是为了术语突出性——夹杂在中文之间的一些简短的英文术语，可读性很高，比全中文化好看得多；三是为了避免混淆——property 和 attribute 不论译为什么都容易相互混淆，干脆不译，又能让读者将来面对英文读物或文档时没有滞碍。然而，尺度的拿捏和取舍向来是很大的难题，在这个题目上，我的取舍标准也因书籍的层次和定位而不同。本书适宜保留许多英文术语和名词，我这么认为！对于不习惯以及不愿习惯“带有英文术语之中文技术书”的读者，这本书的处理方式恐怕不适合你。

本书译稿由我和荣耀先生共同完成。荣耀负责初译，我负责其后工作。荣耀涉猎的技术领域宽广，企业级项目经验尤其丰富，非我能及。我很高兴和他共同完成这部译作。本书由我定稿，再由荣耀协助术语之覆阅修整。

侯捷 2003/02/16

jjhou@jjhou.com

<http://www.jjhou.com> (繁体)

<http://jjhou.csdn.net> (简体)

Understanding .NET — A Tutorial and Analysis

荣耀 译序

新技术总是伴随着误解一道来临，微软 .NET 也不例外。对于微软环境下的开发人员来说，.NET 的重要性无论如何强调都不为过。因此，怀着平和接纳的心态，全面透彻理解这种新技术，对于我们每一位希望与时俱进的架构师、程序员和技术管理人员来说，都至关重要。

.NET 带来的巨量新技术，足以使任何一名老练的开发人员不知所措。虽然 .NET 相关书籍已经随处可见，但它们要么是代码密集型的开发者指南，缺乏透彻的说理；要么仅限于对某一专门技术的描述，无法让您建立完整的 .NET 观念。结果，直到今天，许多开发人员对 .NET 究竟是什么，仍然不明就里。

您手中的这本书与众不同。它涵盖了整个 .NET 平台的方方面面，又有足够的技术深度。倘若您希望理解 .NET 的内涵、外延及其运作机理，本书正是为您而写。它可以让您全面而深入地领会 .NET 平台，又不会迷失于这个巨大而复杂的技术丛林。我向阁下建议，阅读任何一本 .NET

专项技术书籍之前，请先阅读这本书。本书地位独特，难以替代。

本书以 .NET 概观开始，一一阐述 .NET 各组成要素的原理、目标、彼此相互关系，以及它们如何适应 .NET 整体蓝图，内容涵盖 Web Services、通用语言运行层（Common Language Runtime, CLR）、.NET 语言、.NET Framework 类库、ADO.NET、ASP.NET、.NET My Services 等等。甚至连 .NET 企业服务器本书也有清晰描述——尽管这些软件当前发行版既未使用 .NET 框架，也没有对 Web services 提供太多直接支持。

本书秉承 David Chappell 作品的一贯风格，将文字叙述与技术原理示意图有机结合在一起。为了配合讲解，作者偶尔会给出一些代码样例，但比重甚少。重要概念都被整理成“页边注释”的形式。任何时候您都可以安全跳过不感兴趣的段落，不必担心丢掉对主体思想的把握。

本书高明之处在于作者 David Chappell 不但将艰深复杂的技术以浅显流畅的笔法明晰表达出来，更以多年业界经验和对新技术的敏锐洞察力，写出散布于“分析文块”中极富启迪性的观点。作为一名企业级项目负责人，对于本书，我个人给予极高评价。任何一位需要全面理解 .NET 的架构师、程序员和技术管理人员都应该读一读本书。

感谢侯捷先生。先生为人师表，大家风范，是荣耀学习榜样。感谢周筠女士，是您促成了这次合作。感谢我的爱人朱艳，你永远是我前进的推动力。

感谢本书的每一位读者朋友。您在汲取知识的同时，肯定也会强烈体验到精采和快乐。

荣耀 2002 年 10 月
<http://www.royaloo.com>

前言

编写优质软件并无不二法门，亦不存在十全十美的开发平台。然而创造平台的人们仍然为这个遥不可及的理想奋斗，持续地为我们带来新的软件建构技术。除了极少数例外，每一次变革都比它所取代的东西更好。无一例外的是，每一次创新都使我们脑袋里的知识多少有些过时。是的，和软件互动的根本之道就在于与时俱进，随机应变。

.NET 是迄今微软（也可能是任何厂商）向技术客户展示的最大一套新技术。它所打造的巨大革新几乎提升了开发人员生活的每一方面，与此同时，它也带来了大量需要理解的新技术。本书的目标就是帮助你迁移到这个巨大的新世界中。

本书为谁而写

.NET 规模庞大。肯定会有许多书籍提供这一巨大的“技术水晶”的每一侧面的详尽分析，也会有大量书籍提供核心的、易于动手实践的信息。本书不属此类。我坚信从整体上理解 .NET 将是深入钻研其任何专门技

Understanding .NET — A Tutorial and Analysis

术细节的基本要素。因此，我的目标是提供一个 .NET 主体技术的广泛概观。由于这一软件族系及其提供的服务的最伟大力量之一就是它们可以彼此借鉴，因此本书也试图展示这些技术是如何结合在一起的。

如果你正在寻找关于 .NET 的整体描绘，本书正是为你而写。无论你是位刚开始接触 .NET 的开发人员，还是一位需要对这些技术作出决策的技术管理者，抑或是第一次接触这些观念的学生，本书都将是有益的指南。这儿有足够的、可彻底满足许多人需要的信息细节。上述以外的其他人可以将本书当做了解更多细节知识的“垫脚石”。无论如何，我希望这本书的组织方式和涵盖内容，都可以使你在理解并掌握这一巨量技术时省些力气。

事实和观点

要掌握一种新技术，就非得弄清楚其基本原理不可。它的主要构成是什么？它们如何运作？如何彼此结合？但如果要真正透彻理解一种技术，光是这些远远不够。你不但要弄清楚事情如何运作，还要明白它们为何至关重要，它们与过去的技术相比又如何，下一步可能会发生些什么。

本书提供了所有这一切。在正文中我努力保持严肃的教学形式，完全集中于描述 .NET 是什么。在“分析文块”里，我会从不同方面给出一些关于这种技术的概括性看法。每一个“分析文块”里的分析都表达了我对于“为什么事情会是这个样子，将来又会如何”的观点。通过这种将客观事实和主观看法分开描述的方式，我希望能够让你比较轻易地区分两者；在叙述事实的同时我将表达个人观点，我希望本书因此得以更有趣味，更富启发性。

致谢

如果你曾经写过书籍，你就会明白你从别人那儿得到了多少帮助。如果你没有写书经历，那好，相信我，没有以下这些人的协助，本书将失色不少。我想要向以下人士致以由衷的谢意：Bob Beauchemin, Keith Brown, Cori Day, Ted Demopoulos, Bill Estrem, Jeannine Gailey, Kit George, Greg Hack, Rob Howard, Maxim Loukianov, Juval Löwy, Peter McKiernan, Yahya H. Mirza, John D. Mitchell, Christophe Nassare, Eric Newcomer, David Sceppa, Aaron Skonnard 和 Mike Woodring。他（她）们阅读了本书草稿，给我宝贵的意见，并且时常纠正书中存在的这样或那样的问题。我尤其要感谢 Richard Monson-Haefel，他是一位卓尔不群的技术专家和作家，他阅读了每一章，并给出许多有见地的评论。

我所参与的许多 .NET 研讨会的众多出席者也为本书品质的提升做出了不小的贡献。通过为我提供讲解 .NET 技术的机会，他们帮我判断哪一种途径对于理解 .NET 最富成效；通过向我提出发人深思的问题，激发了我的灵感，使我写出散布于本书各处的许多“分析文块”。

Addison-Wesley 的许多人也值得我深表谢意。如果没有 Kristin Weinberger, *Independent Technology Guide* 系列书籍和本书都不会存在；如果没有 Stephane Thomas，我可能永远都无法完成这个项目；如果没有 Cindy Kogut，我的文字将明显失色；如果没有 Katie Noyes，就不会有如此漂亮的封面。

我还要感谢我的好友 Jim 和 Judy Moffitt，在我最需要的时候，他们亲手送来了成包的巧克力夹心饼干。最后，我要把这一切都归功于 Diana Catignani——没有她，我的生命贫瘠无比。

David Chappell
www.davidchappell.com
2001.12

目录

侯捷译序	i
荣耀译序	iii
目录	v
前言	xi
1 .NET 概观	1
定义 .NET	2
Web Services (Web 服务)	3
.NET Framework (框架)	6
CLR (Common Language Runtime, 通用语言运行层)	11
莫基于 CLR 之上的语言	15
.NET Framework Class Library (Framework 类库)	22
.NET Compact Framework (精简版 Framework)	29
.NET My Services	31
.NET Enterprise Servers (企业服务器)	34
.NET 的一个实例场景	38
结语	39
<i>Understanding .NET — A Tutorial and Analysis</i>	

2 Web Services	43
描述 Web Services	44
应用 Web Services	46
访问 Internet 应用程序	46
B2B 整合	47
A2A 整合	48
Web Services 的一个实例场景	48
Web Services 技术	52
描述信息: XML	52
定义 Web Services: WSDL	55
访问 Web Services: SOAP	58
搜寻 Web Services: UDDI	65
Web Services 的未来方向	71
结语	74
3 通用语言运行层 (CLR)	75
开发受控代码: 通用型别系统 (CTS)	76
CTS (通用型别系统) 简介	77
进一步探讨 CTS Types	80
CLS: 通用语言规范	85
编译受控代码 (Compiling Managed Code)	86
微软中介语言 (Microsoft Intermediate Language, MSIL)	87
Metadata (元数据)	90
组织受控代码 (Organizing Managed Code): 装配件	94
装配件的 Metadata: 清单 (Manifests)	94
装配件的分类	97
运行受控代码 (Executing Managed Code)	99
装载装配件 (Loading Assemblies)	99
编译 MSIL	101
使装配件更安全 (Securing Assemblies)	104
垃圾回收 (Garbage Collection)	110
应用域 (Application Domains)	113
结语	116

目录	vii
4 .NET 语言	119
C#	120
C# 程序样例	122
C# 提供的 types	125
C# 的控制结构	132
其他的 C# 特性	133
Visual Basic.NET	142
VB.NET 程序样例	142
VB.NET 提供的 types	146
VB.NET 的控制结构	154
其他的 VB.NET 特性	155
带有受控扩充件 (Managed Extensions) 的 C++	159
Managed C++ 程序样例	161
Managed C++ 提供的 types	163
Managed C++ 的其他特性	163
结语	165
5 .NET Framework 类库 (Class Library)	167
.NET Framework 类库概观	167
System 名字空间 (namespace)	168
System 的次级名字空间 (sub-namespace) 概观	169
输入和输出: System.IO	180
序列化: System.Runtime.Serialization	183
支持 XML: System.Xml	187
XML 技术家族	188
System.Xml 提供了什么	192
反射: System.Reflection	195
.NET 远程系统: System.Runtime.Remoting	200
远程处理过程 (Remoting Process) 概览	202
将信息传递给远程对象 (Remote Objects)	203
选择一个 Channel (通道)	205
远程对象的创建和销毁 (Creating and Destroying)	207
企业服务: System.EnterpriseServices	214

viii	目录
互操作性: System.Runtime.InteropServices	220
访问 COM 对象	220
访问 Non-COM DLLs	225
Windows GUIs: System.Windows.Forms	227
使用 Windows Forms (表单) 开发你的 GUIs 程序	227
Windows Forms 的控件 (Controls)	232
结语	233
6 访问数据: ADO.NET	235
<hr/>	
.NET 数据供应器	237
直接访问数据	242
以 DataSets 访问数据	246
创建和使用 DataSets	248
访问和修改 DataSet 的内容	253
使用 DataSets 和 “以 XML 定义的数据”	256
结语	262
7 开发 Web 相关应用: ASP.NET	263
<hr/>	
浏览器应用程序: System.Web.UI	264
浏览器相关应用程序的工作机理	266
Web 控件 (Web Controls)	270
将用户界面 (UI) 从代码中分离出来	274
其他主题	276
Web Services 的应用: System.Web.Services	289
Web Services 服务器	289
Web Services 客户端	292
Web Services 应用程序的选项	293
微软对 Web Services 应用程序的特定支持	297
结语	298

8 .NET My Services	299
定义 .NET My Services	300
应用 .NET My Services	304
确保隐私	306
.NET My Services 商务模型：软件业就是服务业	308
描绘 .NET My Services	310
认证服务（Authentication Service）：Passport	311
XML-Based Services	313
.NET My Services 的一个实例场景	319
结语	324
关于作者	325
索引	327

1

.NET 概观

软件开发世界变化无常，新思想不断涌现并迅速体现于新技术之中。有时候新知一点一点地被吸收进来，变成软件人员的专业知识。另一些时候，由于变化如此之大，以至于不能说它们仅仅只是现有技术的改良。在这种情况下，软件开发人员、软件架构师和企业管理者，都将有如体验一次基础结构上的大地震！

微软 .NET 的出现，就是这种地震。它将震撼 Windows 环境下工作的任何人，同时也将在范围更广的世界里产生余震。微软一次性地带给我们那么大的变化，要我们适应它，短期来看，将使我们的日子更加难过，毕竟要学的东西太多！然而一旦我们把握了这套新工具和新技术，大多数 Windows 开发人员将会发现，他们有能力在更短时间内构建出更具威力、更有用的软件。

对 Windows 开发人员来说，.NET 是一个结构性的变化。

定义 .NET

.NET 是一个施用于一系列技术上的商标

Web services 是 .NET 中的一个核心技术

微软将 .NET 视为数字化未来的一个远景和平台。如果更具体更准确地看待这种创新，则是把 .NET 视为一个商标，一个微软已经施行于数种不同技术上的商标。这些技术有些是全新的，提供新的服务和新的可能性，另一些则允许我们以最新的方式来创建我们今天已经知道的各类 Windows 应用程序。当然，也有一些 .NET 家族成员只不过是装饰着 .NET 牌子的现有技术的新版本而已。

.NET 之中最重要的新技术首推 Web Services。如其名称所示，Web Services 提供了某些功能，让我们得以通过网络加以调用。大多数顶着 .NET 商标的技术都可以在某种程度下直接支持 Web Services。然而 .NET 绝非仅仅是 Web Services 而已，微软置于 .NET 商标伞下的技术荦荦大者包括：

- **.NET Framework**: 包括通用语言运行层 (Common Language Runtime, CLR) 和 .NET 框架类库 (.NET Framework class library)。CLR 是建造一系列新应用程序的标准基础，.NET Framework 类库则为许多基于 CLR 的应用程序提供一个新的标准开发环境。这个类库包含的技术有：ASP.NET, 最新一代的 ASP (Active Server Pages) 技术；ADO.NET, 最新一代的 ADO (ActiveX Data Objects) 技术；以及对“构建和使用 Web services”的支持等等。微软还发行了一个 .NET Framework 精简版，名为 .NET Compact Framework，用于小型设备如个人数字助理 (personal digital assistants, PDAs) 上。
- **Visual Studio.NET**: 支持多种可使用 .NET Framework 的编程语言，包括 Visual Basic.NET，它是新一代的 Visual Basic；一个增强版的 C++；以及一个为 .NET Framework 量身设计的全新语言 C#¹。

¹ 发音为 "C sharp"，和音符中的“升记号”发音一样。