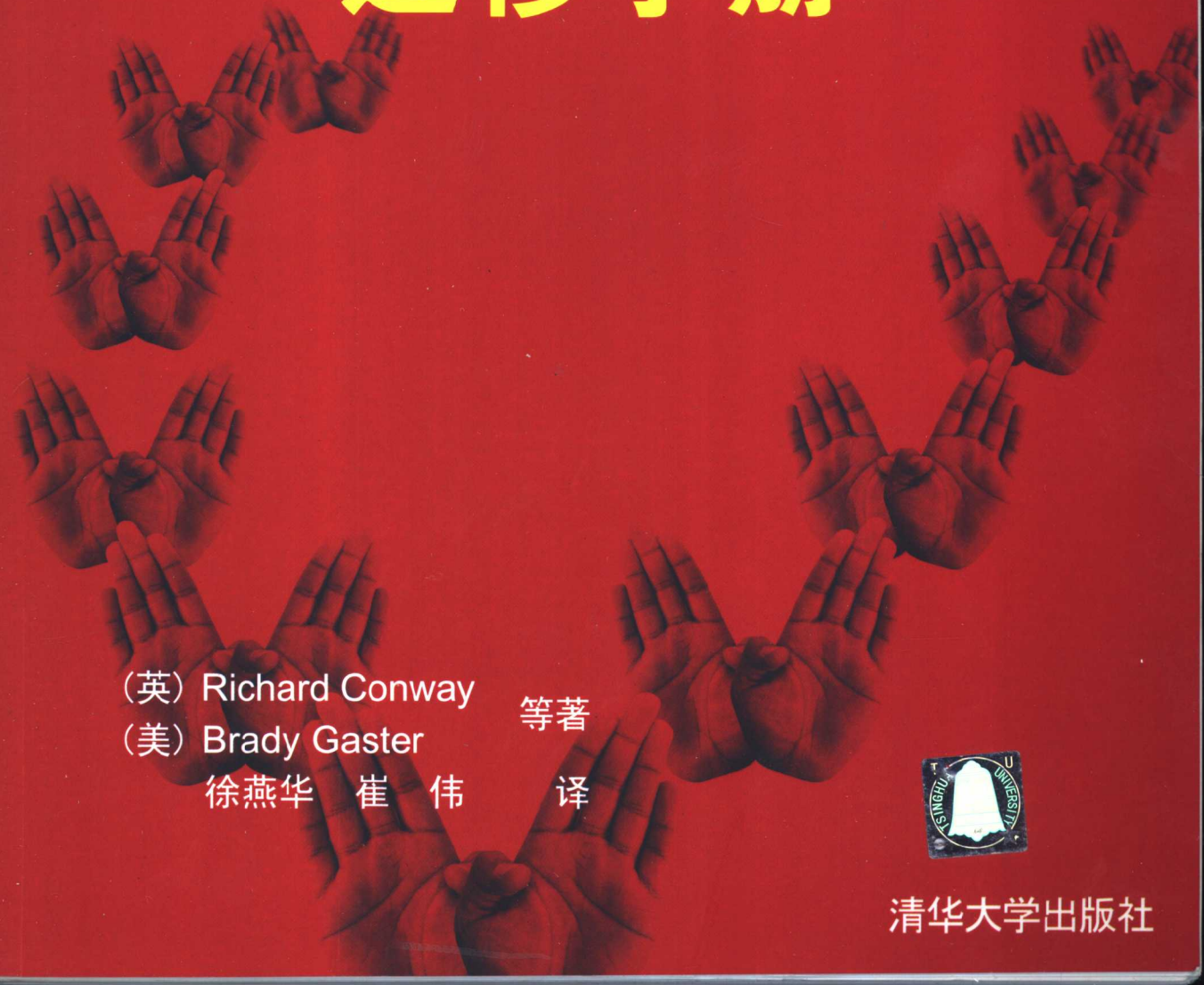
PROGRAMMER TO PROGRAMMER™

ASP To ASP.NET Migration Handbook

ASP 到 ASP.NET 迁移手册



（英）Richard Conway
（美）Brady Gaster
等著
徐燕华　崔　伟　译

清华大学出版社

ASP 到 ASP.NET 迁移手册

(英) Richard Conway
(美) Brady Gaster
等著

徐燕华 崔伟 译

清华大学出版社

北 京

内 容 简 介

本书与您分享那些已经完成从 ASP 到 ASP.NET 迁移的开发人员的宝贵经验，了解他们在迁移到 ASP.NET 时所采用的各项技术。本书介绍如何选择迁移策略、重新设计应用程序来利用.NET 的功能、ASP 和 ASP.NET 在编写代码时的区别、如何将业务对象导出到.NET、如何集成现有的 COM 对象和服务，以及如何充分利用.NET 数据访问和 XML 功能等内容，以便帮助您成功地实现迁移操作。

本书适合于那些曾经用 ASP 构建过 Web 应用程序并希望将 ASP 应用程序迁移到 ASP.NET 应用程序的读者阅读。

EISBN：1-86100-846-5
ASP To ASP.NET Migration Handbook
Richard Conway，Brady Gaster，et al.

北京市版权局著作权合同登记号　图字：01-2002-4606

图书在版编目(CIP)数据

ASP 到 ASP.NET 迁移手册/(英)康威等著；徐燕华 崔伟译.— 北京：清华大学出版社，2003
书名原文：ASP To ASP.NET Migration Handbook
ISBN 7-302-07405-4
I. A… II. ①康… ②徐… ③崔… III. 主页制作—程序设计 IV. TP393.092
中国版本图书馆 CIP 数据核字(2003)第 092469 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：王晓娜
封面设计：康博
版式设计：康博
印 刷 者：北京世界知识印刷厂
装 订 者：北京鑫海金澳胶印有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：15.25　**字数**：316 千字
版　　次：2003 年 11 月第 1 版　2003 年 11 月第 1 次印刷
书　　号：ISBN 7-302-07405-4/TP•5470
印　　数：1～4000
定　　价：30.00 元

前　言

ASP.NET 并不仅仅是 ASP 的一个新版本。从 ASP 到 ASP.NET 的迁移并不同于 ASP 以前版本之间的迁移。这里需要了解的差异和概念更多，其中一个主要原因是，ASP.NET 现在已是.NET Framework 平台的一部分。

在 ASP.NET 中，很多时候需要用不同的方式来考虑 Web 应用程序。为了充分利用该技术，我们需要调整方法，使之适用于所构建的应用程序。

如果要将现有的 ASP 应用程序迁移到 ASP.NET 应用程序，则需要仔细考虑与之相关的操作。如果仅仅是将 ASP 代码一行行地转换为与 ASP.NET 兼容的代码，并不会带来多少好处。但借此机会重新设计并创建应用程序，则可以令我们受益匪浅。

本书并不准备详细讲述 ASP.NET；这种类型的参考书籍很多，而且内容更详尽。因此，我们希望您在学习本书内容时，要结合其他相关书籍的内容。本书将与您共享那些已经完成迁移操作的开发人员的宝贵经验，帮助您轻松实现从 ASP 到 ASP.NET 的迁移。

我们无法对迁移问题提供简单的解答；Web 应用程序种类繁多，因此要提供这样简单的答案是不现实的。我们的目标是帮助您了解实现解决方案中的概念、技术和实践操作。

本书适时提出一些疑问，将您的注意力集中到重要问题上，从而帮助您尽快、顺利地完成迁移。

本书读者对象

本书适用于那些具有用 ASP 构建 Web 应用程序的经验并希望迁移到 ASP.NET 应用程序的读者。如果您当前有一个或多个需要迁移到 ASP.NET 的 ASP 应用程序，那么本书是最好的选择。不过，即使没有这样的应用程序，在从 ASP 转向 ASP.NET 的过程中，本书对您也很有帮助。

在学习本书的过程中，您需要了解 ASP.NET 的相关内容，以便实现到 ASP.NET 的迁移。因此，建议您在学习本书时结合参考其他有关 ASP.NET 的书籍。封底中列出了一些我们建议的读物。

本书主要内容

第 1 章—— 迁移策略

本书从具体操作入手来介绍迁移问题。我们将探讨一些可用于迁移的不同方法，讨论其中的一些主要注意事项。

第 2 章—— 重新设计应用程序

本章介绍到 ASP.NET 的迁移会如何影响整个应用程序体系结构。

第 3 章—— 表示层体系结构

本章介绍 ASP 和 ASP.NET 之间在具体编写代码时的区别。我们将了解 ASP.NET 如何改变表示层的工作方式。

第 4 章—— 重新设计用户界面

本章继续介绍表示层的代码，探讨如何充分利用 ASP.NET 提供的新功能。

第 5 章—— 重写 VB 业务对象

COM 业务对象是许多 ASP 应用程序的重要部分。本章讨论了在将这些组件转向.NET 时需要注意的事项。

第 6 章——COM 互操作性

迁移业务对象的另一种方法就是在 ASP.NET 应用程序中使用已有的 COM 对象。本章讲述了其实现方法并讨论了这样做的后果。

第 7 章—— 服务组件

如果准备向 ASP.NET 迁移的 ASP 应用程序依赖于 COM+服务，则必须可以在 ASP.NET 中使用那些服务。幸运的是，.NET Framework 提供了相关功能，本章将介绍相应内容。

第 8 章—— 关于 ADO.NET

ASP 和 ASP.NET 之间的区别类似于 ADO 与 ADO.NET 之间的区别。本章讨论了数据访问方式改变的本质，并介绍了如何利用 ADO.NET 的各种功能。

第 9 章—— 关于 XML

许多 ASP 应用程序的功能涉及到 XML，但.NET Framework 针对处理 XML 而提供的辅助程序使得 XML 更为有用。本章将介绍在从 ASP 到 ASP.NET 的迁移过程中，如何导出基于 XML 的功能，并讨论如何更好地利用 XML。

目　　录

第1章 迁移策略

在详细介绍从ASP迁移到ASP.NET之前，先看看影响迁移的一般性问题和使迁移过程尽可能简单的策略是很有帮助的。

我们没有承诺要给出完整的迁移解决方案，因为确定采用什么适合给定环境的技术解决方案是迁移成功的重要组成部分。由于不同站点的Web站点和Web应用程序差别甚大，所以没有一种放之四海而皆准的策略可以使用。

根据经验可知，在实际应用中，影响Web站点的决定通常并不明确，经常受到会危及安全的条件和限制条件的影响。例如，理想的迁移策略可能并不是最合适的策略，因为进行迁移工作的团队在特定技术方面没有足够的技能，从而及时优质地制定出迁移解决方案。

通过本章的学习应该能更好地理解可以用于迁移工作的方法。本章将讨论某些影响迁移的关键问题，阐明实现迁移策略要比想象中复杂得多的原因。

1.1 迁移的商业原因

没有人闲着没事将ASP迁移到ASP.NET。迁移必有明确的商业原因。

获得商业中风险承担者的认同是迁移工作中的非常重要的一个步骤。在这个阶段做一些工作，可以确保特定迁移项目可以获得必需的支持，以及整个迁移过程得以成功的适当资源。

从后面就会看到，从ASP成功迁移到ASP.NET需要许多条件，其中包括新软件和培养新的技能。必须确定在迁移过程中可以支配的资源。

迁移的商业原因主要有两个——技术方面和由此引发的财务收益。

1.1.1 技术方面的原因

从一种技术迁移到另一种技术的直接影响当然在技术方面。我们将介绍ASP.NET比ASP更为优越的7个关键领域。当然，也可以考虑其他方面的问题，但这些是ASP.NET带给多数开发人员的主要好处。

考虑各方面的问题时，评估其在您所在环境中的相对价值比较重要。例如，如果ASP.NET并不能带来性能方面的实质性提高，那么在确定商业原因时就很难将包括性

能在内的事务视为问题。

迁移的商业原因方面主要有下面 7 大技术优势：

- 性能　将广泛的高速缓存技术与文件编译结合在一起，ASP.NET 的性能优于 ASP。
- 可伸缩性　ASP.NET 使整个 Web-farm(服务器组)都可以共享会话数据。虽然可以使用为 ASP 提供了 Web Farm 支持的解决方案，但是 ASP.NET 却使得 Web Farm 更容易支持，也更容易管理。
- 可维护性　ASP.NET 鼓励将表示代码(presentation code)和业务逻辑分开，这样使代码维护更容易。ASP.NET 还推广更严谨的 Web 开发方法，这将使应用程序的维护工作更容易。配置已合理化，这样就能使维护工作更方便。
- 统一的开发环境　可以使用一个 IDE(Visual Studio .NET)来开发并调试其逻辑层上的所有应用程序。
- 部署　ASP.NET 使部署工作尽可能简单。例如，它不需要注册与 ASP 开发有关的 COM DLL。设计得当的 ASP.NET 应用程序通过复制其文件并在 Internet Information Services 中设置几个选项应该就可以部署为服务器产品。
- 快速开发　ASP.NET 提供的“开箱即用”功能比 ASP 过去的版本多得多。结合使用 Visual Studio .NET 的拖放式设计模式，就可以非常迅速地构建完善的 Web 接口。
- 可重用性　ASP.NET 面向对象和基于控件的体系结构使得创建复杂、可编程的可重用用户界面元素较以前容易得多。

1.1.2　财务收益方面的原因

任何商业活动的基础当然是资金。为了使企业能接受从 ASP 迁移到 ASP.NET，就必须给它带来财务方面的收益。

如上节所述，ASP.NET 在技术方面的优势也有益于提高多个领域中的潜在财务收益：

- 开发速度更快；
- 代码使用寿命更长；
- 维护成本更低。

1. 开发速度更快

ASP.NET 提供了许多可重用的 Web 应用程序功能。其中包括成熟的用户界面控件、安全功能和高速缓存技术。这样在开发新应用程序时，开发人员就可以比其他人快得多了。

如果开发人员曾接受过 ASP.NET 开发方面的全面培训，使用 ASP.NET 就比使用 ASP 创建通用 Web 应用程序功能的速度要快得多。

2. 代码使用寿命更长

代码的使用寿命就是在重新编写前使用的时间段。ASP.NET 承诺可采用下面两种方式来增加其使用寿命：

- Web Farm 支持这样的可伸缩性功能就意味着，应用程序可以按需要伸缩而不是重新编写。
- 比起使用 ASP 来，可以在未来的项目中更容易地重用代码。

这两种方法都意味着开发优秀的代码可以获得更大的价值。

3. 维护成本更低

维护费用占大型 Web 应用程序所有者总成本中较大的一部分。

从上节可以看到，维护使用 ASP.NET 开发的应用程序应该比使用 ASP 开发的应用程序要难得多。

1.1.3 不迁移的风险

考虑从 ASP 迁移到 ASP.NET 的益处的同时，也应该考虑选择不迁移的后果。

坚持使用 ASP 存在下面许多潜在风险：

支持风险　由于使用 ASP 的人越来越少，就得不到好的支持。开发社团越小，共享信息的机会就越小。

兼容性风险　Microsoft 的新产品一定都与.NET 相兼容。第三方开发人员设计的工具和组件可能供.NET 使用而不是旧技术。坚持使用 ASP 就意味着以后会出现兼容性问题。

落后于他人的风险　从 ASP 迁移到 ASP.NET 是一大进步，从 ASP 迁移到下一个环境 ASP.NET 确实是很大的进步(不管是不是 ASP.NET)。随着新技术的不断开发，都需要确保我们没有落后别人太远，我们自己也要紧跟时代发展的脚步。

1.2 迁移进度计划

迁移到 ASP.NET 所需的时间主要取决于具体情况。很显然，复杂的大型重要商务 Web 应用程序与小而简单的应用程序的进度计划不同。

起草进度计划时，通常会低估其复杂性并忽略不能按时完成的问题。在非正常的原

因以及开发人员没有或参与大型迁移项目的经验很少的情况下，迁移任务就更加困难。

而且，确定可接受进度计划是保持迁移项目成功的关键因素—— 我们必须在用户和管理需求以及常见的需求之间权衡，以避免设计人员、开发人员和测试人员因为时间不充足而将解决方案进行折衷处理。

也会出现这样的情况：由于完成关键任务所需时间不确定而影响迁移策略的选择，帮助实施迁移策略是最低要求，迁移策略可以分解成分散的可管理阶段。

为了合理地估计进度计划，就需要识别并考虑会影响迁移过程的一些问题。

- 站点的大小和复杂程度 站点需要使用复杂的业务逻辑，跨几台服务器甚至几个国家，容纳数千页面。由于存在不兼容问题，迁移站点可能会耗时数年，并且重写代码也会耗费财力。另一方面，系统越小复杂程度就越小。
- 应用程序的当前状态 笨拙的 ASP 站点并不是迁移的优秀候选对象。设计和维护较差的站点也会出现问题。站点迁移后的情况取决于迁移前其自身的“健康”状态。在迁移准备工作中，不“健康”的站点需要维护或重新创建。
- 选用的迁移策略 策略会直接影响迁移应用程序的进度计划(本章稍后将讨论迁移策略)。在某些情况下是由进度计划决定所采用的策略。
- 迁移的准备情况 迁移应用程序前，需要迁移工作团队作一些准备工作。设计人员、开发人员和管理员不能只是熟悉.NET 技术。它们需要全面了解该技术并且是最新的技术。至少迁移团队要评估采用 ASP.NET 编写的现有站点的适用性和兼容性。开发人员需要精通 ASP.NET，至少能运用一种.NET 语言，并且要有在 ADO.NET 方面及在迁移之前要用的其他技术方面的出色工作经验。
- 融资方面的局限 影响进度计划的现实问题就是是否有资金。需要将大型站点的迁移工作作为项目进行融资，它在时间上会跨越一个财年，因此资金流量决定着迁移的时间跨度。采用现有预算分配就可以迁移小型应用程序。
- 资源的可用性 可以采用与迁移准备工作相似的办法来审查资源的可用性。需要有合适的人力资源用于制定迁移计划、实施并监控迁移工作。我们需要一个在.NET 技术方面训练有素的团队。如果站点没有掌握最新.NET 技术的体系结构设计师，就可能失去由全面理解体系结构所带来的益处。

1.3 技能要求

上节简要介绍了一些技能。技能要求会影响迁移策略的选择。技能的熟练程度影响迁移的质量。

下面将详细介绍迁移到 ASP.NET 所必需的技能。

1.3.1 进行迁移需要的技术

我们认为团队应该有 ASP、ADO、COM、HTML、脚本语言和数据库方面的技能。除此以外，迁移还需要下列技能：

1. 语言技能

在 ASP 开发中，对于 UI 开发来说有脚本语言(VBScript 或 JScript)和 HTML 方面的技能就足够了。对于中间层，通常需要 COM 开发语言(VB 或 C++)方面的技能，而数据层则需要 ADO、数据库结构化查询语言(T-SQL 或 PL-SQL)和更新的 XML 方面技能。

在 ASP.NET 开发中，由于这些技术都基于新的范式，所以对语言技能的要求也发生了巨大变化。ASP.NET 页面的体系结构使用后台编码，这是一个使得应用程序开发语言获得更好效果的文件。

在许多使用 ASP.NET 的情况下，有一种.NET 开发语言(VB.NET、C#或 J#)方面的优秀技能并了解丰富的服务器控件库就足够了—— 熟悉 HTML 也很方便。所有应用程序开发项目(Web、Web 服务、Windows、控制台和类库)都可以使用所选择的.NET 语言，所以可以说，.NET 对语言的要求不再那么苛刻。

也需要 XML 方面的技能。.NET 广泛使用 XML 实现各种功能。例如，DataSet 对象(ADO.NET)在本地以 XML 格式保存数据。

ASP.NET 开发不再需要了解脚本语言。对许多开发人员来说，对其语言技能的要求，只是掌握一种.NET 语言，他们在 XML、SQL 和 HTML 方面的工作经验就是.NET 语言所需的知识。

2. 面向对象编程

由于.NET 语言都是面向对象的语言，全面了解面向对象的设计和开发原则是必要的。虽然 VBScript 和 VB 开发人员都想使用 VB.NET，因为他们就可以使用 VBScript 和 VB 了，但是避免出现这种情况的意义是很大的。在 ASP.NET 应用程序中应用面向对象的设计和技术，就可以利用提供类继承的体系结构所带来的益处。例如，不用开发数以百计的分散 ASP.NET 页面，而是考虑使用面向对象的技术，通过扩展一小组模板(ASP.NET 页面或类)来开发页面构成的站点。

ASP.NET 页面是可以被扩展的类。派生类(另一个 ASP.NET 页面)可以继承其功能。

1.3.2 可迁移到.NET 的现有技能

如果选择了可以进行共存的迁移策略(在同一台服务器上运行ASP页面和ASP.NET页面)，那么我们的 ASP 技能对于维护和开发 ASP 页面或应用程序都是很有用的。

不过，如果选择完整的 ASP.NET 迁移策略，许多 ASP 技能的价值就有待商榷。例如，越不重视服务器端和客户端脚本技能，对.NET 语言和服务器端控件就需要越重视。ASP.NET 不再支持 VBScript，它已由 VB.NET 替代。不过，XML 和 HTML 的现有知识仍然很有用。

不仅语言技能不同，而且许多事例中 ASP 的设计概念和开发方法也与.NET 没有相关之处。如上节所述，面向对象设计和开发方法体系创建了不同的方法，这些方法主要强调类继承和其他面向对象的概念(抽象、封装和多态性)。

随着 ADO.NET 和.NET 程序集的流行，ADO 和 COM 技能也将不断贬值——不过，由于过去已在 ASP、ADO 和 COM 方面投入了巨资，所以这个贬值的过程会比较长。

1.3.3　如何使开发团队跟上.NET 的发展步伐

请注意，我们现在在讨论如何使开发团队赶上.NET 的发展，而不只是 ASP.NET。ASP.NET 只是.NET 技术的组成部分，其中包括：

(1) Microsoft 的.NET Framework：

- ASP.NET——Web 服务、Web Forms 和 ASP.NET 应用服务。
- .NET Framework 基类——包括 ADO.NET、NET、XML、安全、线程和诊断技术。
- 公共语言运行库(CLR)——内存管理、通用类型系统和生命周期监控。

(2) .NET 开发语言：

- C#、VB.NET 或 J#等。

除了需要培养面向对象的设计和开发方面的优秀技能外，还要学习如何使用新的 IDE(Visual Studio .NET)。没有 IDE，可以使用文本编辑器和编译器进行开发，Java 开发人员比传统的 Microsoft 开发人员(过去常常觉得使用直观的 IDE 更舒服)更支持这种方法体系。

要赶上这些技能的发展，通常需要比预期的时间更长。也就是说，目前对.NET 或完全面向对象的开发语言不是很了解，并不是因为它们很难掌握，而是要学的东西太多，需要花一些时间才能使技能达到运用自如的水平。一些开发人员需要 6~12 个月或更多时间才能精通.NET 语言以及面向对象的设计和开发理念。

Visual Studio .NET 提供了优秀的集开发和调试为一体的环境，在这种环境下可以开发 ASP.NET 应用程序的所有层。它也是使程序员可以速成的优秀工具。

Visual Studio .NET 的替代方案就是 Web Matrix，它是由 Microsoft 提供的可免费获得的供 ASP.NET 使用的 IDE(可以从www.asp.net下载)。虽然 Web Matrix 适合业余开发人员，却不适合用于企业开发，它不支持一些重要的功能，因此也不能采用 Visual Studio .NET 的方法进行企业应用程序开发。