

Osborne 计算机专业技术丛书

Professional  
DEVELOPER'S  
LIBRARY

# COM+

## DEVELOPER'S Guide

# COM+ 开发指南

[美] John Paul Mueller 著

天宏工作室 译



- 构造企业级 Windows “分布式网间应用程序” (DNA)
- 学习使用 “Microsoft消息队列” (MSMQ)和 “Microsoft事务服务器” (MTS)
- 充分利用COM+的特性和各种应用程序类型
- 参照纯粹的功能代码示例编写实用的COM+应用程序

CD-ROM!



清华大学出版社  
<http://www.tup.tsinghua.edu.cn>



麦格劳-希尔国际公司  
<http://www.mcgraw-hill.com>

Osborne 计算机专业技术丛书

# COM + 开发指南

[美] John Paul Mueller 著

天宏工作室 译

清华大学出版社

---

|

# (京) 新登字 158 号

COM + 开发指南  
COM + Developer's Guide  
John Paul Mueller  
ISBN: 0-07-212086-X

Copyright © 2000 by The McGraw-Hill Companies, Inc.  
Authorized translation from the English language edition published by McGraw-Hill, Inc.  
All rights reserved. For sale in the People's Republic of China only.

北京市版权局著作权合同登记号 图字 01-2000-2903 号

JS485 / 14

本书中文简体字版由美国麦格劳-希尔国际公司授权清华大学出版社在中国境内出版发行。未经出版者书面许可, 任何人不得以任何方式复制或抄袭本书的任何部分。

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。

书 名: COM + 开发指南  
译 者: 天宏工作室  
出 版 者: 清华大学出版社 (北京清华大学学研大厦, 邮政编码 100084)  
<http://www.tup.tsinghua.edu.cn>  
责任编辑: 丁朝欣 许振伍  
印 刷 者: 清华大学印刷厂  
发 行 者: 新华书店总店北京发行所  
开 本: 787 × 960 1/16 印张: 27.25 字数: 602 千字  
版 次: 2000 年 12 月第 1 版 2000 年 12 月第 1 次印刷  
书 号: ISBN 7-900630-75-9  
印 数: 0001 ~ 6000  
定 价: 56.00 元

# 致谢

**感**谢我的妻子 Rebecca，正是因为她与我的共同努力才使这本书得以完成。如果没有她来帮助我对本书中出现的一些信息进行研究和编辑，那么我真的不知道自己写出的书将是什么样子。而且她还很好地校对了我的手稿并为最终的书稿校对了对页码。

应该感谢 Greg Guntle，感谢他为本书所做的技术编辑工作。他大大地提高了你在这里所见的资料的精度和深度。此外，他还非常努力地测试了你在本书中见到的许多步骤——一遍又一遍。如果没有他在分析 COM+ 的错综复杂的技术中所给予的帮助和支持，我真的不知道自己写出的书将是什么样子。

特别要感谢的是 Karen Watterson。她花费了许多时间来审查我的数据库设计方案并确保我的 SQL Server 信息的准确性。许多变动都是在她的建议下做出的，而且在我完成编写书中有关数据库的内容这种有难度的任务的过程中，她总是不遗余力地鼓励我。

Matt Wagner 是我的经纪人，他应该受到赞誉——他帮助我率先获得了合同，而且对大多数作者都不会真正考虑的所有细节问题都作出了考虑。我真的很欣赏他采取了体贴而又友善的方式来处理我在编写这本书的过程中所遇到的某些家庭问题。毫无疑问，他使我的工作变得更容易进行，比起可能会发生的情形要好得多。

Microsoft 的技术支持人员也应该受到赞誉——他们回答了有助于填补空白的诸多问题，并且使 Visual C++ 的学习经验传播得更快。同时，我还想感谢自己曾经在各种各样的 Internet 新闻组中访问过的人，他们提供了对 C++ 编程技术的深刻见解。尤为重要的是关于如何使用 OLE DB 和 ADO 的一些提示。

最后，我还要感谢 Wendy Rinaldi、Ann Sellers、Carolyn Welch、Dennis Weaver，以及 Osborne 产品部的其他工作人员——感谢他们为使本书出版而给予的帮助。我尤其要感谢 Wendy，她在事情不像预想的那样顺利时表现出了良好的耐心。

## 简介

Windows 2000 是许多人在激动与苦恼参半的情况下所期盼的操作系统。在编写本书之时，Windows 2000 仍旧是 Microsoft 眼中的一束微光，但它不会永远是微光。随着 Windows 2000 发布日期的临近，媒体中的大肆宣传也在不断增加——Windows 2000 即将成为 Windows 操作系统的主要升级版本。毫无疑问，许多公司将会立即升级，而其他公司会在做出这种飞跃之前抱以一种观望的态度。因为 Windows 2000 提供了相当多的新特性，所以，无论公司决定何时升级，最终都将这样做。从网络管理员的角度来看，仅 Active Directory 自身所提供的优点就使得 Windows 2000 成为必须进行的更新。当然，在大多数公司的升级决策中，新的安全特性——如 Kerberos 和“公钥基础结构”（Public Key Infrastructure, PKI）将会扮演重要的角色。

对于大公司的程序设计人员或者独立的开发人员而言，Windows 2000 激动人心的部分就是全新的“应用程序编程接口”（Application Programming Interface, API）特性，比如它引用的媒体流。当然，在这些特性中，最重要的就是 COM +。很显然，还要考虑 Active Directory 和新的应用程序专用 API。总之，程序设计人员将可以使用大量的新“玩具”，其中的许多“玩具”将使用户可以更容易地创建健壮的企业级应用程序。

在 Windows 2000 中，令人苦恼的部分是 COM + 以及其他这些新技术所需要的相当艰难的学习过程。对于许多开发人员来说，COM + 所需要的一切首先是花费许许多多多个夜晚阅读有关新技术的资料，并耗费大量时间实验新技术。（很显然，所有这些工作的成果将是一些网络或 Internet 上可扩展的应用程序，并最终可以缩短开发时间。）Microsoft 暗自承认 COM + 将会引起许多开发问题，他们在整个 Windows 2000 测试版中提供了大量开发人员教程来尽量避免其中的一些问题。

这就是这本书如此重要的原因之一。它可以帮助你了解 COM + 将会为你做些什么，以及怎样才能使学习变得更轻松——至少是那么一点点。我们将探索 COM + 世界，并帮助你将它的各种技术分解为更易于处理的部分。你将会发现，COM + 的大多数表面上的复杂性不过是现在单独的软件包中多项技术的某种组合，以及将使 COM + 开发作为一个整体比现在更容易进行的一些增强措施。

许多书籍都会提供所有我们在本书中将要谈及的理论背景。本书的与众不同之处在于，我在演示 COM + 时采用了实际的方法。当然，那就意味着使用实际的编程示例，这些示例将能帮助你更快地使用 COM +。在企业级上，正在使用的一种

更为重要的应用程序类型是数据库——这就是我们将首先创建它，然后以各种方式来增强它的那种应用程序。本书中的数据库示例使用了多个表，而且我们将介绍创建数据库以及管理数据库的应用程序所需要的一整套设计步骤。很显然，这个过程的某些部分将涉及到编写一个服务器端组件，这种组件不仅使查询过程变得更容易，而且还会减少网络的通信量。

COM + 不应该成为一个潘多拉盒子，其中充满了理解技术及实用工具的障碍，使你的每一次努力都受挫。本书将帮助你学习这个令人激动的 Windows 2000 新特性。同时，我们还可以了解 COM + 能够完成的一切事情，以使桌面和 LAN 应用程序能够扩展到 WAN 和 Internet 上。

## 本书的内容

从目标的角度来看——看看本书的目录会有所帮助——本书将帮助你完成哪些任务以及如何完成？本书有三个主要目标：介绍 COM + 理论，展示如何在常规环境和企业编程环境中使用 COM +，最后是提供功能齐全的代码示例。第一章至第六章将介绍 COM + 理论以及使用细节，而第七章至第十章将提供功能齐全的代码示例。

第一章的目的是使你熟悉 COM +。它是通过提供有关 COM + 自身的一些直接信息并比较 COM + 及其前身 COM 来实现这一点的。对比是很重要的，因为它会帮助你了解：假如熟悉了今天的 COM，那么 COM + 的学习过程将会比最初想像的要容易得多。而且，还有两个专门的小节来讲述 COM + 环境中的新特性：“Microsoft 消息队列”(Microsoft Message Queue, MSMQ)和“Microsoft 事务服务器”(Microsoft Transaction Server, MTS)。MSMQ 已被增强为支持“断开的应用程序”(disconnected application, 我们将在本章后面介绍这个术语)这样的特性，而 MTS 已被增强为为数据库以及类似对象提供事务处理。

第二章提供了 COM 自身的一个基本概况。这一概述代替了许多书用 4~5 章的篇幅所介绍的内容，这意味着我并没有花太多时间来介绍所有细节。这一概述不足以讲授有关 COM 的知识，所以，如果对今天的 COM 一无所知，那么本书的这一部分将不适合于你。本章提供了足够的信息来确保我们双方对基本概念都有同样的理解。另外，这一概述将为以前曾经使用过 COM 但最近没有使用 COM 的人提供一次复习的机会。最后，本章将帮助你了解什么是 Microsoft 用来创建 COM + 的基础。在我们开始后面的 COM + 学习旅程之前，拥有一个好的起点是很重要的。

第三章是一个导言，介绍与 MTS 或者 MSMQ 不直接相关的具体的 COM + 特性。这一章介绍基本的 COM +。与 MTS 以及 MSMQ 相关的信息分别包含在第四章和第五章中。从这三章中，将了解到 COM + 在本质上是 COM、MTS、MSMQ 和 DCOM (它仍然用于通过网络传送数据)的一个增强和组合版本。这三章还包括了如何使

用以及何时使用 COM + 信息的大部分内容。在结束这三章时，应该掌握了足够的理论信息以理解后面的编程示例。

第六章专门在实践水平而不是理论水平上讨论重要的 COM + 问题。它概述了你可以使用 COM + 来创建的、COM 不支持——至少不是直接支持——的应用程序类型。是的，COM + 是 COM 的一个扩展，是一个能帮助你比以前更多地利用组件技术的扩展。COM + 确实允许创建一些新的应用程序类型，这些应用程序将会从整体上缩短开发企业级应用程序套件所需的时间，理解这一点是很重要的。

第七章至第十章提供了 4 个功能齐全的编程示例。我尽量选择一些例子来展示 COM + 在中型的公司级至企业级的广泛应用。这些示例专注于新的 COM + 用法而不是从 COM 简单复制的 COM + 功能。例如，数据库示例介绍了如何使用一个服务器端组件来减少网络通信量。这个服务器端组件利用 MTS 来确保所有事务都按照预期情况发生——其中包括那些实际上并非数据库专用的事务。

## 适用范围

本书专供专业（富有经验的）程序员使用。虽然某些资料明确地面向企业，但它并没有区分顾问程序员和企业程序员。开发人员需要对 Visual C++ 非常熟悉，并且至少有一些使用 COM 的经验。我们确实提供了使用 SQL Server 的逐步说明，但是了解“数据库管理系统”（Database Management System, DBMS）的工作方式仍然是必需的。我并不打算花费大量的时间来指出 DBMS 是如何工作的，但是我将介绍为什么采用某种方式（其中一些方式似乎与当今的惯例相反）来实现它们。因此，本书将只提供多种 COM 概念的一个概况，以便专注于 COM + 的编程细节。

因为 MTS 与 MSMQ 是 COM 编程环境的新生事物，因此，对这两个主题的介绍将稍多一些。不过，我假定用户已经知道了如何在用户一级上使用这些产品。换句话说，本书将要提供的是技术细节，而不是用户级信息（示例程序所需的用户级信息除外）。

## 必要条件

COM + 是一项新技术，Microsoft 还没有开发出一种统一的工具（至少在编写本书的时候还没有）供用户使用 COM +。最起码，将需要 Visual C++ 6.0 专业版（企业版是任何数据库示例所必需的）、Windows 2000 Platform SDK，以及 Microsoft Data Access Components (MDAC) 版本 2.5。或许还需要 COM + SDK。对这种产品的需求取决于在作者完成本书到用户从书店买到本书这段时间里 Microsoft 所做的工作。从理论上讲，COM + SDK 将被合并到 Windows 2000 Platform SDK 之中。



作为 Visual C++ 安装的一部分，一定要安装 Service Pack 3。这个升级非常重要，因为当使用 Windows 2000 时，如果试图调试一个应用程序但没有安装 Service Pack 3，则 Visual C++ 几乎一定会停止。这个升级包含在较新的 Visual C++ 6.0 中，并且是该软件包的一部分。还可以从 Microsoft Visual Studio Web 站点上下载它。最后，MSDN 预订中将包含该服务包。一定还要安装 SQL Server 所需的任何服务包，以及在本书中使用的其他产品。我将在本书的第七章中说明如何应用 SQL Server 更新。

在大多数情况下，不能使用 Visual C++ 产品的早期版本，这是因为我计划在每一个示例中都使用新特性。（对于某些示例，Visual C++ 5.0 或许也能工作，但是，我不会提供使用这种产品的任何工作环境。）我假定用户将使用 Windows 2000 作为操作系统，而且任何一个示例都不能保证能够在 Windows 操作系统的任何其他版本上运行。事实上，我可以保证它们几乎没有例外。



**注意：**应该在本书中学习的许多概念都没有出现在联机文档中，因为其中的某些内容是最新的，只会在特定的 Web 站点上出现。你将会发现，在本书中，自始至终都会有一个提示或注意来提醒这种信息的位置。另外，Microsoft 还将某些资料设置为只能通过选定的频道（如 MSDN 预订）获得。其他信息则没有相应的文档，不能在任何地方找到它们，除非一个新闻组中的某个人偶然发现了该特性。

还需要一台运行 Windows 2000 Professional（或者 Windows 2000 Server）的计算机来作为工作站，而且我还极力推荐使用一台运行 Windows 2000 Server 的辅助服务器计算机（必须至少有一台运行 Windows 2000 Server 的计算机来提供 MSMQ 支持）。服务器和工作站都需要有足够的 RAM 和其他资源，以完全支持在整本书中都需要工具。在大多数情况下，这意味着将需要一台最低安装了 64MB 内存和至少 4GB 硬盘空间的 350MHz 的 Pentium II 计算机来作为工作站。服务器将需要额外的 RAM 和硬盘空间。在服务器上使用多个处理器是一种有利的手段，但并不是必需的。

虽然可以使用较低的配置，但是，如果使用一台低档的计算机，那么，在编写代码并且进行测试的时候，将发现计算机会迅速陷入停顿状态。我曾经试着在一台安装了 128MB 内存和 3GB 空闲硬盘驱动器空间的 300MHz 的 Pentium 计算机上运行这些示例，其性能非常糟糕。在编写本书时，由于 Windows 2000 施加于这台计算机上的负载，因此实际上我可以观看到屏幕重画。Microsoft 可能会解决其中的某些性能问题，而其他问题则会在你使用产品正式版而不是测试版时消失，但是，提供一种配置以使这些示例在一段合理的时间完成运行是很有意义的。



## 本书中使用的约定

在这一节中，我们将介绍本书中使用的约定。稍后，我们将在介绍匈牙利表示法以及如何使用它的时候讨论编程约定。本书使用以下约定：

- [ <文件名> ] 如果看见某个值、开关或者命令带有方括号，那么，意味着它是一个可选部分。不必将它包括为命令行或者对话框的一部分，除非需要该值、开关或者命令所提供的附加功能。
- <文件名> 尖括号中的变量名是一个需要使用其他数据来代替的值。将会看到的变量名通常提供一个需要何种信息的线索。在这里，需要提供一个文件名。在键入该值时，千万不要键入尖括号。
- 全部大写 将在三种位置看到全部大写的情况，即命令、文件名，以及区分大小写的注册表项。通常将在 DOS 提示符下 PIF 文件域内或者在“运行”对话框域内键入一个命令。如果在其他位置看到全部大写的情况，则可以假设该项目是一个区分大小写的注册表项或者是像文件名这样的其他值。
- File | Open 菜单以及菜单上的选项在出现时都包含一条竖线。“File | Open”的意思是“访问 File 菜单并选择 Open”。
- 斜体 将在两种位置看到斜体文字：多值条目和未定义的数值。无论何时，只要某个实际值是未知的，那么就总会看到一个斜体值。在多个数值都可能正确的地方，本书也使用斜体。例如，可能会在文字中看到 FILE $xxxx$ 。这意味着该值可能是 FILE0000 和 FILE9999 之间的任何值。
- 等宽字体 对于将在某个宏中使用的文本或者将在命令行中键入的文本，将其与解释它的文本加以区分是很重要的。本书使用等宽字体来进行这种区分。每当看到等宽字体时，就会知道所看到的信息将会出现在某个宏中、位于某个系统文件（比如 CONFIG.SYS 或 AUTOEXEC.BAT）中，或者作为将在命令行中键入的内容。在这种文本中，甚至会看到随 Windows 命令使用的开关。在另一种情况下，也会看到等宽字体。为了使文字便于阅读，每一个代码清单也都使用等宽代码。使用等宽代码还使得向代码示例添加缩进这样的格式变得更加容易。

## 图标

本书包含几种帮助用户识别某种信息类型的图标。以下段落说明了每一种图标的用途。



**注意：**它会指出一些有趣的事实，而这些事实不会影响对本书中其他信息的使用。在本书中使用“注意”来提供一些信息，这些信息是我在使用 Visual C++、Windows NT 或者 Windows 95 时获得的。



**提示：**每个人都喜欢“提示”，因为“提示”可以指出一些处理事情的新方法，而这些新方法可能是以前没有想到过的。“提示”还提供了完成某件事情的替代方法，与我所提供的第一种方法相比，你或许更喜欢这种方法。



**小心：**它的意思是要当心！“小心”几乎总是告诉你：如果执行某项操作（或者是未能执行某项操作），那么将会出现哪一类系统故障或数据损坏。在遵照它后面的提示进行之前，一定要确保自己已经透彻理解了其内容。



**Web 链接：**虽然 Internet 上包含了大量信息，但退一步说，查找信息却很困难。Web 链接可以帮助你 Internet 上查找一些新的信息资源，可以使用这些资源来完善自己的编程方法或者学习新的技术。还可以找到一些新闻组 Web 链接，它们指出应该到哪里去寻找其他人，以便与你一起讨论 Visual C++。最后，Web 链接还可以帮助查找一些实用工具程序，这些程序将会使程序设计变得比以往更快而且更容易。

### 附加说明

附加说明包含一些附加的信息，这些信息不是编写完美的程序时所必须了解的。在许多情况下，我将提供一些与当前主题相关的资料，但这些资料的相关程度没有那么强，或者没有那么重要，因此不足以包括在主题中。大多数附加说明将会提供有趣的资料，以供中级或高级程序员增强他们所创建的应用程序，或者增强他们自己的创造力。例如，可能在某个附加说明中找到一个对向导配置问题所做的说明，或者找到一个论题，论点是“为什么某种特定的添加项产品对于开发人员来说很有用”。

读者还将在本书中发现各种页边注释。这些页边注释描述了一些信息，这些信息都是在开始某个过程或者执行其他类型的工作之前应该记住的。页边注释还包含一些有用的资料，比如某个文件的位置或者应该在某个示例程序中查找哪些内容。在大多数情况下，页边注释只是一些可以用来从整体上改善编程的信息。

## 匈牙利表示法概述

秘密代码是侦探影片以及其他各种人士努力探索的东西。第一次看到匈牙利表示法的时候，可能会将它视为另一种秘密代码。它具备了秘密代码的全部要素，同时还包含了不得不对其进行解码的一系列神秘字母，以及当这样做时将会出现的几乎难以破译的结果。不过，你不久就会认识到，真正神秘的是其他程序员所编写的代码而不是本书中使用的匈牙利表示法。

匈牙利表示法可以节省许多时间和精力。在试图理解自己在以前的一次编程中做了些什么工作或者尝试解释别人的代码时，曾经花费足够的时间从事编程工作的人都会认识到编写好的文档所具有的价值。这就是匈牙利表示法将完成的那部分工作——说明代码。

对匈牙利表示法的了解还可以帮助你更深入地理解本书的示例以及 Microsoft（或者其他开发商）的手册。几乎每一个 Windows 编程语言开发商都在他们的手册中使用某种形式的匈牙利表示法。此外，这些相同的概念同样可以应用于其他语言（比如 Visual FoxPro、Delphi 以及 Visual Basic）中。即使语言本身是各不相同的，代码在各种编程语言之间也仍是相似的。

那么，匈牙利表示法究竟是什么呢？它是告诉其他人你将如何处理某个变量的一种方法。了解某个变量应该做些什么常常可以帮助你解释代码本身。例如，假如我指出某个特定的变量包含了一个窗口的句柄，那么就会知道与该变量有关的更多信息，而不仅仅是它是一个变量这个事实。在了解此变量将对窗口产生影响的情况下，就可以解释变量周围的代码。

这个变量命名系统的第一个开发阶段是由 Microsoft 公司的 Charles Simonyi 开始的。他将自己的系统称为“匈牙利表示法”，这就是我们在这里所使用的名称。在许多地方都可以获得他的作品的一个副本，其中包括 BBS 以及 Internet 上的一些 Microsoft 编程 Web 站点。（许多联机服务，如 CompuServe 也包含以各种变化形式出现的匈牙利表示法副本。）Simonyi 的工作由其他开发者做了进一步的增强。例如，Xbase 程序员使用他们自己特有的匈牙利表示法版本，它考虑了 Xbase 所提供的不同类型的变量。Fresh Technologies 的 Robert A. Difalco 发表了匈牙利表示法的一个增强 Xbase 版本，可以在 DBMS 专用的少数几个 BBS 上以及 CompuServe 的 Computer Association Clipper 论坛上找到他的工作。

可以在前面所提及的文档中发现本节所提出的想法之基础的某种形式。在这里公布它们的目的是想让你知道我所使用的约定的确切本质，以及如何才能在自己的代码中充分利用它们。有四个原因可以说明为什么应该在程序中使用这些命名约定：

- ▼ **有助于记忆的值** 这使你更容易记住变量的名称，它是团队项目的一项重要考虑。
- **建议值** 你或许不是惟一一个修改代码的人。如果正在参与一个团队项目，那么团队中的其他人至少会看一看已经编写的代码。使用这些约定可以帮助其他人理解在使用某个特定的约定时所要表达的意思。
- **一致性** 一个程序员的工作不仅要根据其效率或者功能来考察，还要检查它是否容易被其他人理解。使用这些约定可以帮助你在一个项目和另一个项目之间保持统一的代码。其他程序员只根据你所使用的约定就能够预见到一段代码的意义或功能。
- ▲ **决策速度** 在商业环境中，创建和修改代码的速度经常会决定某一笔交易是否成功。在创建一个变量或函数时，使用一致的代码将会缩短试图确定某个人所要表达的意思所花费的时间。这种决策时间的缩短将会增加用于生产性工作的时间。

既然我已经指出了为什么应该使用匈牙利表示法，现在就让我们来看一看我计划如何在本书中实现它。在命名变量时，我将使用下一节中的规则。你还将看到，我在命名数据库字段或者其他与值有关的结构时也使用这些规则。在某些函数和过程中也将使用它们，但只有在匈牙利表示法将使该函数或过程的含义更清晰的情况下才这样做。

### 规则 1：对变量加前缀

始终在一个变量的前面加上一个或者多个小写字母来作为前缀，这些字母用来表示该变量的类型。在大多数情况下，这是变量类型的第一个字母，因此很容易记住应该使用哪一个字母。下面的示例显示了 Visual Basic、Delphi 和 C 语言中最常用的一些前缀。（Windows 中使用了数以百计的组合，但这里没有显示它们。）

a	数组
c	字符
d	日期
dbl	双精度数
dc	设备描述体
dw	双字

f	标志、布尔值或者逻辑值
h	句柄
i	整型
inst	实例
l	长型
li	长整型
lp	长指针
msg	消息
n	数值
o	对象
pal	调色板
psz	指向以零结尾的字符串的指针
ptr	指针（在与 psz 这样的其他变量一起使用时为 P）
r	实数
rc	矩形
rgb	红、绿和蓝（颜色变量）
rsrc	资源
sgl	单精度数
si	短整型
sz	以零结尾的字符串
u	无符号数
ui	无符号的整型或字节
w	字
wnd	窗口

## 规则 2：标识状态变量

有些变量表示一个对象——如数据库、域或者控件——的状态。它们甚至可能用来存储另一个变量的状态。告诉其他程序员一个变量正在监视某个对象的当前状态有助于他们看到该变量在程序中的重要性。你可以使用以下的三字符限定词来标识状态变量：

New	新状态
Sav	已保存状态
Tem	临时状态

### 规则 3: 使用标准限定词

标准限定词有助于其他人立即看出某个变量的用途。这不是该变量所包含的信息类型，而是关于该变量如何与其他变量相互作用的信息。例如，使用 Clr 限定词可以告诉查看者，这个变量是以某种方式与颜色一同使用的。甚至可以组合使用限定词以增强它们的效果并且说明该变量是如何使用的。例如，cClrCrs 是一个可以决定显示器上光标颜色的字符变量。使用 1~3 个这样的限定词通常就足以说明变量的用途了。下面这些限定词是一些比较常用的类型。

Ar	数组
Attr	属性
B	底部
Clr	颜色
Col	列
Crs	光标
Dbf	数据库文件
F	第一个
File	文件
Fld	字段
L	上一个/左边
Msg	消息
Name	名称
Ntx	索引文件
R	右边
Rec	记录编号
Ret	返回值
Scr	屏幕
Str	字符串
T	顶部
X	行
Y	列

### 规则 4: 添加说明文字

一旦清晰地定义了变量的内容和用途，就可以使用一些说明文字来进一步说明其定义。例如，可能有一个长指针指向某个包含雇员名称的字符串：lpszEmpName。

前两个字母 (lp) 指出这是一个长指针。接下来的两个字母 (sz) 指出这是一个以零 (或者 NULL) 结尾的字符串。其余的字母 (EmpName) 指出这是一个雇员的名称。(请注意, 我在例中使用了标准限定词 Name。) 在一段代码中看到类似的变量名可以立即使你知道它所具有的含义。

### 规则 5: 创建一个以上的变量

在许多时候, 使用一个变量将无法满足不同模块中的所有需求。在这些情况下, 可能想要创建一个以上的该类型的变量, 而且只需对它们编号即可。或许还可以使用某种编号指示符来指定其功能, 如下所示:

1, 2, 3	cSavClr1, cSavClr2 等变量中的状态指针引用
Max	nFldMax 中的严格上限, 字段的最大编号
Min	nRecMin 中的严格下限, 记录的最小编号
Ord	某种类型的序号



## 关于随书光盘

**我**相信用户会喜欢本书中的所有示例，并且当用户学习有关 COM + 的知识时希望尽可能地充分利用它们。但是，试图手工键入全部代码将是令人沮丧的，尤其在看到第七章中的示例时更是如此。很显然，需要有一种键入全部代码的替代方法，比如，采用电子方式来达到这一目的，这样就不必重复我所做的工作了。

过去，某些书的源代码可以在 Osborne Web 站点上获得。本书的源代码多达 76MB，这使得那些使用拨号连接的人下载源代码的过程变得非常耗时而且容易出错。

本书的光盘是为用户设计的。它使用户无需键入一行代码或者下载任何内容就可以查看源代码的工作示例。除了包含源代码之外，光盘还包含了每一个示例的编译版本，这就使得这些示例更易于使用。

请记住，光盘是一种只读设备。因此，在将文件从光盘复制到硬盘驱动器上时，这些文件也将被标记为只读的。可以使用“Windows 资源管理器”来清除硬盘驱动器上文件的只读属性，以便可以根据需要更改源代码或者重新编译。

如果你对源代码存在疑问，那么欢迎使用 e-mail (JMuedler@mwt.net) 与我联系。我希望这本书尽可能便于使用和学习。同时，也请一定将自己的意见告诉我。我一直在记录读者的意见并利用这些意见来改进我编写的书籍。对于任何一本书，不论是正面意见还是反面意见，我始终都是欢迎的。

# 快速目录

第一章	COM + ——组件技术中的最新技术	1
第二章	COM 基础——简化版本	13
第三章	COM + 独有的特性	51
第四章	MTS 概述	91
第五章	MSMQ 概述	113
第六章	理解应用程序的类型	147
第七章	事务驱动的应用程序	169
第八章	处理事务失败	285
第九章	发送消息和 COM 对象	331
第十章	在断开模式下工作	371