

学校的理想装备

电子图书·学校专集

校园网上的最佳资源

Visual C++6

从入门到精通



Visual C++ 6 从入门到精通

[美] John Paul Mueller 著

希望图书创作室 译

本书配套光盘内容包括：

1. 与本书配套的中文版电子书
2. 送“精通 Visual C++ 5.0”多媒体学习软件

北京希望电脑公司

北京希望电子出版社

1 9 9 9

内 容 提 要

Visual C++ 6 是 Microsoft 公司最新出品的 C++ 6 开发工具。本书以详实的内容、精选的示例全面介绍了 Visual C++ 6 的新特性以及如何在软件开发过程中运用这些新特性。全书共分五部分：第一部分介绍 Visual C++ 6 的基础知识和基本概念；第二部分介绍数据库编程方法；第三部分阐述 Internet 编程；第四部分说明创建应用程序时应该考虑的安全性问题、普通格式和 HTML 格式帮助文件的创建方法以及应用程序安装盘的建立与分发；第五部分是附录，介绍了 ActiveX 和 OCX 组件资源、在线资源以及本书涉及到的一些技术术语。

本书内容覆盖了开发完善的应用程序所需的各种知识，同时，书中给出了大量与 Visual C++ 编程相关的站点和讨论组地址，这些地址是真正的 Visual C++ 6 程序员上网的站点。本书对已经具有 Visual C++ 编程经验的读者是很好的升级指导书。对初学 Visual C++ 的读者来说，利用本书也可以很快入门，迅速达到中级甚至高级 Visual C++ 程序员的水平。该书对从事 Visual C++ 6 开发与应用的广大科研人员、高校相关专业师生均不失为一本重要的自学、教学参考书。

本书配套光盘的内容是：1.与本书配套的中文版电子书；2.送“精通 Visual C++ 5.0”多媒体学习软件。

需购本书和配套光盘或需技术支持的读者，可直接与 010-62562329，62541992，62531267 或传真至 010-62579874 北京海淀 8721 信箱书刊部（邮编 100080）联系。

版 权 声 明

本书英文版原名为“ Visual C++ 6 from the Ground up ”,由 McGraw Hill 公司出版,版权归 McGraw Hill 公司所有。本书中文简体字版由 McGraw Hill 公司授权出版。未经出版者书面许可,本书的任何部分均不得以任何形式或任何手段复制或传播。

Visual C++ 6 从入门到精通

[美] John Paul Mueller 著

希望图书创作室 译

责任编辑 陆卫民

北京希望电脑公司

北京希望电子出版社 出品

北京海淀路 82 号 (100080)

新华书店、新华书店音像发行所发行、各地书店、软件专卖店经销

* * * * *

1999 年 3 月第 1 版

1999 年 3 月第 1 次印刷

开本：787 × 1092 1/16

印张：33.125

字数：756 千字

印数：1~5000

新出音管 [1998] 210 号

ISBN-7-980023-17-X/TP · 17

定价：58.00 元 (1CD, 含配套书)

作者简介

John Mueller 是位自由作家和技术编辑。到现在为止，他已经编著了 40 部著作，撰写了近 200 篇论文。其讨论主题覆盖了从网络到人工智能、从数据库管理到高级编程的广阔领域。Visual Studio 程序员指南和 Windows NT Web 服务器手册是他近期作品中的两部。他在技术编辑方面的经验使他已经帮助了 22 位以上的作者提炼了他们的手稿。除了撰写著作之外，John Mueller 还为 Data Based Advisor 和 Coast Compute 两本杂志做技术编辑服务。作为计算机界认证领域的知名专家，他还在像 Certified Professional Magazine 这样的杂志上发表与认证相关的文章。

当 John 不在计算机旁工作时，你会在他的工作间找到他。他是位干劲十足的木匠和蜡烛制作者，不管在哪天下午，他都会或工作在车床旁，或把做好的物品放置到书架上。他最新的工艺项目之一是制作甘油肥皂，这是制作礼品篮的好材料。读者可以通过 jmueller@mwt.net 在 Internet 上与 John 联系，John 目前正在建立新的 Web 站点：<http://www:mwt.net/~jmueller>。随意到这些站点看一看、并提些改进建议吧！

Osborne/McGraw-Hill
2600 Tenth Street
Berkeley, California 94710
U.S.A.

致 谢

感谢我的妻子 Rebecca，是她与我一起完成了本书。如果没有她查找和编辑本书中的某些信息（特别是索引部分），我真不知道是否能够写出这部书。她还详细校对了本书的初稿和终稿。

感谢本书的技术编辑 Greg Guntle，他大大增强了本书所用材料的准确性和深度。

我的代理人 Matt Wagner 保证我得到了第一份合同，并处理了绝大多数作者都没有深思的许多出版细节。

Microsoft 技术支持人员回答了我许多提问，进一步丰富了本书，同时也使我更快地掌握了 Visual C++。与此同时，我还要感谢访问 Internet 各种各样新闻组的人们，他们协助提供了 C++ 编程技巧的内幕，特别是他们提供了如何使用 OLE-DB 和 ADO 的技巧。

最后，我要感谢 Wendy Rinaldi，Nancy McLaughlin，Kathryn Hashimoto，以及 Osborne 的全体工作人员，在他们的帮助下本书得以问世。我特别要感谢在事情意外不顺利时 Wendy 所具备的耐心。

译者序

Visual C++ 6 是 Microsoft Visual Studio 套装软件的一个有机组成部分。虽然在构造大多数常规应用程序方面，Visual C++ 没有市场上常见的快速开发工具（比如 PowerBuilder，Delphi，Visual Basic）的开发效率高，但在速度和代码大小要求较高的场合（比如设备驱动程序等），它依然是首选的开发工具之一。

Visual C++ 6 在以前版本的基础上又增加或增强了许多特性。除了当今的每个开发工具都不可或缺的 Internet 特性外，ADO 数据绑定、ATL 复合控件、AutoCompletion、编辑和继续特性、OLE-DB 提供者模板、延迟加载移入以及新的调试特性等都给 Visual C++ 6 增色不少。

本书作者在熟练使用 Visual C++ 6 的基础上，充分听取各阶层读者对本书先前版本的意见和建议，阅读了 Internet 上有关 Visual C++ 的大量内容，并与 Microsoft 的技术人员进行了充分交流，最终成就了这部著作。本书分为五个部分：第一部分介绍 Visual C++ 6 的基础知识和基本概念；第二部分介绍数据库编程方法；第三部分阐述 Internet 编程；第四部分说明创建应用程序时应该考虑的安全性问题、两种格式的 help 文件的创建方法以及应用程序的分发；第五部分是附录，介绍了 ActiveX 和 OCX 组件资源、在线资源以及本书涉及到的一些技术术语。

除了详实的技术细节外，本书也为读者提供了大量的与 Visual C++ 相关的 Internet 站点。通过这些地址，读者不仅能够了解 Visual C++ 的更多内容及发展趋势，参加一些讨论组提高自己的技术水平，同时也避免了读者在 Internet 上大海捞针般地艰难搜寻 Visual C++

的相关材料。

本书不仅适用于初学 Visual C++ 的新手，对已经熟练运用 Visual C++ 以前版本的高手来说，也是一本不可多得的参考资料。

本书由张长富、栾开春、彭涛、张建安、黄石海、姚燕、李匀、黄中敏、陈华等翻译，最后由张长富统稿。参加翻译、录入、校对和资料搜集工作的还有：陈聪、崔立模、宋斌、张剑宇、夏桂娟、王志宏、王锋、张兵、李鹰、王丰、冯桂林、严苏娅、龚楚奇、欧阳志、郝晓伟、苏春波、王刚、李林。翻译过程中得到了享受政府特殊津贴的专家张福林先生的热情指导和帮助，在此深表感谢。由于译者水平所限，欢迎读者批评指正。

张长富

1998,11

引 言

长期使用 Microsoft Visual C++的任何人都会感受到该产品的巨大变化。Microsoft 花了很大的力气来提高 Visual C++这一最新版本的易用性。然而，对 Visual C++程序员来说，编写程序依然是艺术成分多于科学成分。就在 Microsoft 创建新的向导把一个领域的隐秘(比如这个 Visual C++版本中改进的 ATL 向导)暴露于世的同时，某些新技术又把新的隐秘加入到了 Visual C++中(比如 OLE-DB 和 ADO 支持等)。

不断的变化也是你需要另一本 C++参考书的原因。Microsoft 不断地向 Visual C++中增加新特性。虽然本书不能从理论上介绍更多的 C++基本编程知识，但它将会告诉你如何使用 Microsoft 新增的新特性。例如，你会在本书中找到 Visual C++新版本中包含的大量 ActiveX 和其它与 Internet 相关的特性。在 Visual C++的每一次版本更新中，Internet 的主导地位都进一步得到了增强，6.x 版也不例外。正是人们对速度的渴求驱使 Visual C++使用某些 Internet 部件。从 Internet 上下载大型应用程序不是有效的用户选择，用户需要小型、快速的应用程序。听起来很熟悉，是不是？我们将花费大量的篇幅讲述如何把 Visual C++的优势最佳地应用到当今 Internet 驱动的编程环境中。

ActiveX 是用户与分布在 Internet 上的应用程序进行交互的方法之一。它为 Web 网主带来了吸引潜在访问者的更多工具。在许多情况下，你可以得到各种功能而无须编写一行代码。很显然，第一个要问的问题是，“谁关心这些事呢”？答案是想开发内联网或 Internet 站点的所有公司。一项最近的调查表明，当今 48%的公司正在部署 Internet 上的应用，14%

的公司声称他们将在未来的六个月中部署 Internet 应用，28% 的公司计划在一年内部署 Internet 应用（总的百分比达到 78%）。显然，人们对 Internet 上的应用兴趣很大。

这个版本的 Visual C++ 还包括了支持应用程序中嵌入浏览器功能。你可以在多种场合使用浏览器支持功能，包括在你的 Web 服务器上放置帮助文件，以使用户在需要的时候就能访问它们。想一想把桌面帮助浏览器置入应用程序中带来的好处吧。单单支持电话的减少所带来的好处就足以说明应该把浏览器置入应用程序中。这一浏览器支持功能还使旅途中的雇员能够向公司上传数据和从公司下载诸如最新的销售统计以及情况分析之类的信息。

为什么 Internet 会成为这样一种炙手可热的环境呢？另一项调查询问商业客户为什么他们要使用 Internet 提供某种类型的增值服务或为雇员提供更好的信息访问。被调查者中 79% 的人认为 Internet 易于使用。访问 Internet 上海量信息所要完成的一切工作就是找个 ISP（Internet 服务提供商）和浏览器。浏览器在安装上并不比任何其它 Windows 应用困难——并且有时候还要容易些。而且，有些 ISP 提供了自动化程度很高的安装程序，用户只需要提供姓名、地址以及诸如信用卡号码这样的付帐信息即可。例如，我最近使用过的 Pacific Bell 安装程序自动创建 Windows 95 下我所需要的连接，然后拨号登录到它们的 Web 站点来帮助我查找最佳的访问号码，并在检查了另一个 Web 站点后完成安装过程。

基于用户使用 Internet 的统计数据，Microsoft 正投资于把分布式数据库支持添加到 Visual C++ 中。本书中介绍的 OLE-DB 和 ADO 将会告诉你，Microsoft 如何把建立与数据库的连接变得更加容易些，而无论这些数据库放置在什么地方。如果 OLE-DB 连接的提供商允许的话，你甚至可以在 Internet 建立与数据库的连接。

显然，如果说易于使用是人们选择使用 Internet 的原因之一的话，那么许多公司都会有许多使用 Internet 的其它原因。然而，调查中发现使用 Internet 还有其它原因，包括增加

了可用性（69%）、利用了现有基础设施（65%）、成本（51%），以及易于维护（36%）。这项调查允许读者进行多项选择，因此会有上述的百分比。Internet 起到了信息中心仓库的作用，并且或许是绝大多数公司为远程雇员提供访问公司数据库的最佳途径。

任何程序员都知道，代码已经能够运行并不足以说明应用程序已经编写完成。还有许多美学方面的考虑。比如，可以编写出运行速度更快的代码，也可以缩减可执行代码的长度。事实上，编写小型、紧凑代码是专家级程序员的标志。这是个极为重要的目标，贯穿于本书的主题就是如何精炼每一行代码。

编程语言的选择决定了程序员在应用程序中都能完成哪些工作以及以怎样的速度完成这些工作。Visual C++一向以灵活著称：理论上使用 Visual C++可以完成任何任务，并且不会让你的代码变得像意大利式细面条那样乱做一团。C++还以正确使用能够创建出运行速度极快的应用程序而著称。然而，这些高度的灵活性和快的速度经常给初出茅庐的程序员带来麻烦，也经常形成微妙的故障。许多 Visual C++的反对者也指出使用这种语言时需要很长的开发周期——这是使用诸如 Visual Basic 之类的 RAD（快速应用开发）语言进行编程的用户注意不多的事实。

有些人走得远了些，声称 C++的日子屈指可数。快速的计算机并不像以前那样需要既小且快的代码。而且，随着应用程序越来越复杂，在合理的时间期限内完成应用程序是基本的要求。C++应用程序过长的开发和调试周期确实有点不应当今的应用程序开发环境。

对 C++程序员来说，幸运的是，有些编程任务 RAD 语言永远也代替不了 C++。即使 RAD 语言已经取代 C++来开发某些类型的应用程序，但对操作系统服务或设备驱动程序来说，程序员依然需要编写简短、紧凑的代码。完成相对底层任务、需要反复执行的代码的编制依然是 C++程序员的工作领域。

本书概要

我已经有点伤害你的好奇心了，现在就让我们看一看本书的主要内容吧。本书分四个重点：通用应用程序编程、与 Internet 相关的编程、数据库编程以及应用程序打包。当然我们也要花点时间讲述编程基础。例如，我们将考察一下最新版的 Visual C++ 都提供了哪些界面元素以及新的编程特性。

本书的通用应用程序编程一节将向你展示如何使用 MFC AppWizard 创建基本的应用程序。我们不花许多时间来研究创建特定类型应用程序的细节，但将讲述创建 MFC AppWizard 提供的各种类型的应用程序的方法。显然，在本节中我假定读者已经具备了一定的 C++ 编程经验。当然，我会介绍创建应用程序本身的详细做法，但读者应该知道一些 C++ 的基本概念，特别是语言元素方面的知识。

数据库编程示例将为已经知道一些数据库基本知识的读者提供丰富的信息。如果你还不知道什么是表、还没有坚实地掌握关系数据库的理论，那么在阅读本书第二部分之前，需要花点时间先熟悉一下上述内容。我们将研究使用 C++ 操作数据库的基本技巧，你也将理解 ODBC (Open Database Connectivity, 开放数据库互连)，OLE-DB (Object Linking and embedding database, 对象链接与嵌入数据库)，ADO (ActiveX Data Object, ActiveX 数据对象) 以及 DAO, (Data Access Object, 数据访问对象) 之间的差别。我们甚至还各花了一章的篇幅来讲述 ODBC 和 ADO 数据库编程。

Internet 编程实际上从本书的第一部分就开始讲述了。这里研究了如何创建基于 HTML 的应用程序。第二部分将讨论如何创建跨越 Internet 的数据库连接，然而这里我们并没有考察任何实际的实例代码。第三部分是介绍 Internet 的中心章节，我们将以中级程序员的水平来研究 ActiveX 编程。如果你只是个初级程序员，那么会发现某些概念有点难以理解。

然而，某些章节，比如第 8 章讲述的 HTML 编程，将会为各层次的程序员提供帮助。另外，第三部分将向读者展示如何使用 ActiveX Document(ActiveX 文档)，ISAPI Filter(ISAPI 过滤器)，以及 ISAPI Extension(ISAPI 扩展)。

在第四部分，我们将讨论从安全性问题到把帮助文件组合在一起的方方面面的内容。实际上，我们还要探讨 Visual C++中使用帮助文件的两种不同技巧：标准的帮助文件和 Microsoft 正在其产品中使用的新型 HTML 帮助文件。你还将学习发行应用程序的方法，即使在企业环境中这也是一项不可或缺的任务。

通过本书，你将会深入理解 C 编程语言，虽然它还不至于让你成为一名 C 编程高手。本书中你不会找到很多初级的编程知识，但具备一定 C 编程技能的任何人都会理解本书的内容。积极建议读者已经具备 Microsoft Visual C++的一定知识，原因在于所有的示例都以 Visual C++作为基础。即使我们把示例控制在基本编程技巧的范围内，也依然需要编写 OCX 或 DLL 的知识。如果你在阅读本书之前已经掌握了一些底层编程知识，那么你就会很快地学会掌握提供的各种信息。再说一句，本书包含了一些“课程复习”风格的示例，但它们也没有提供很多的初级信息。

当然，上面介绍的仅仅是本书的概貌，在这本书中你会找到能够把 Visual C++的功能发挥得淋漓尽致的各种各样的有用技巧和提示。

本版著作的新内容

本书新内容想法的产生主要有三个来源。首先，我花了大量的时间阅读读者发送来的电子邮件。读者的电子邮件对我编著的每一部书籍都显著地丰富了其内容。第二，Visual C++本身为我提供了一些想法。我必须在本书中覆盖 Visual C++的所有新特性。最后，我花了

大量的时间阅读各种 Visual C++新闻组以查找新的消息线索。某些消息线索十分正规，它们指出了人们使用 Visual C++经常遇到的问题。

第 1 章的绝大部分内容都是全新的。我重新编写了第 1 章，以便提供本版 Visual C++的各种新特性的完整概览。另外，本章的某些部分重新进行了排版，以便能够以更便于阅读的形式向用户提供信息。我竭力以更佳的方式安排插文和表格，让读者能够更快速地找到所需信息。

某些读者抱怨说，在本书的先前版本中我过多地讲述语言的新特性而忽略了普通的编程需要。在本版中，你会发现大大增强了普通应用程序编程材料。第 2 章和第 3 章中提供了几节新材料，并提供了一个新的示例程序。

本书的数据库部分已经进行了几个方面的修订。虽然 DAO 依然是一种可靠的技术，然而它已经过时了。我使用新的 OLE-DB 和 ADO 技术取代了先前书中的 DAO 一节。另外，阐述数据库的章节着重介绍如何通过 Web 服务器连接到数据库上，而不是介绍如何通过传统的网络路由器连接到数据库上。

本书的 Internet 部分也包含了与新技术和改进技术相关的信息。例如，在第 8 章中讨论了 COM+。我还提供了一些 Active Template Library (ATL, 活动模板库) 的信息。

本书 Internet 部分最大的变化出现在第 13 章。许多读者表示他们希望看到有关 ISAPI 的更详细材料，希望找出如何利用 ISAPI 更有效地使用 Web 服务器。这一章提供了 ISAPI 的更多信息和两个新的编程示例，其中的第二个 ISAPI 过滤器示例显示了如何给你的 Web 服务器增加安全功能。

本书的第五部分也有许多变化，其中最明显的改变是第 15 章讨论的基于 HTML 的帮助。在这里你不仅能够学会如何创建基于 HTML 的帮助，而且也将学会如何把这种帮助添加到你的应用程序中。除了可以提供更佳的帮助信息外，你还会发现安全方面和打包方法

的更新。总而言之，这部分的更新内容可以帮助你向使用你开发的应用程序的人们提供更多、更好的应用程序支持。

系 统 需 求

如果你只是对本书的第一、第二以及第四部分感兴趣，那么阅读本书所需的一切就是一份 Microsoft Visual C++ 的拷贝。由于我在每个示例中都试图使用 Visual C++ 的新特性，因此，在绝大多数情况下你都不能使用该产品的老版本（对某些示例来说，Visual C++ 5.0 勉强能用）。无论你使用何种版本的 Windows，一定要确保安装了最新的补丁包和服务包。在测试本书的示例时，我使用的是 Windows 95 的 OSR2 版本和安装了服务包 3 的 Windows NT 4。

如果你打算使用第 14 章的 Internet 信息服务器（Internet Information Server，IIS）示例，那么一定要安装 Windows NT Server 4，并使用服务包 3 进行升级。你还需要在服务器上运行 IIS 4.0。某些示例代码根本不能与以前版本的 IIS 一起工作。第 14 章中的所有示例均假定你使用的是功能完备的 IIS，而不是其它的替代品。

注释 本书中你将学习的许多概念在联机文档中根本就没有。某些概念十分新颖，它们只是一些有选择的 Web 站点上出现。对这样的信息，本书将采用技巧或注释的方式告诉读者它们的地址。另外，有些材料 Microsoft 只通过像 MSDN 订户这样的有选择渠道进行发布。其它一些信息则未被公开，除了在新闻组中某些人偶然发现这些特性外，在其它的任何地方你都找不到它们。

你还需要一台运行 Windows 95/98 或 Windows NT 的计算机作为工作站。你的 Web 服务器和工作站都要求有足够的内存和其它资源以支持本书中用到的各种工具。也就是说，在绝大多数情况下，你的计算机的最低配置应该是 200 MHz 的 Pentium MMX、64MB 内存、至少 3GB 的硬盘空间。虽然你可以降低上述配置，在编写和测试代码时低端计算机会很快让你陷入泥潭之中。我已经在 Pentium 166MHz，32MB 内存的计算机上运行过 Visual C++，但性能让人感到恐怖。

本书使用的约定

本节讨论用法约定。在我们研究匈牙利表示法和如何使用这种表示法时，我们将讨论编程约定。本书使用下述约定：

[<Filename>]	当方括号括住某个值、某个开关或某个命令时，表示该项是个可选项。只有当你想用相应的值、开关或命令提供的功能时，才需要把该项包括到命令行或对话框域中
<Filename>	尖括号中间的变量名是个需要使用实际值代替的名称。你看到的变量名通常是提示词语，你需要提供相应的信息。对左边的示例来说，它表示你需要提供一个文件名。在键入对应值时，千万不要键入尖括号

续表

全部为大写字母	在三种情况下你会看到全部是大写字母：命令、文件名以及大小写敏感的登记项。通常在 DOS 提示符、PIF 文件域、Run 对话框域中键入命令。如果你在其它地方看到了全大写字母，你可以安全地假定它们是大小写敏感的登记项或像文件名这样的其它一些值
File Open	菜单和从菜单中进行的选。 “ File Open ” 表示 “ 访问 File 菜单，并选择 Open 菜单项 ”
斜体字	在三种场合你会看到斜体文字：新词、多值项以及不确定的值。无论何时，当实际值未知时，总使用斜体字表示这个值。书中在多个值都可能正确的情况下也使用斜体字印刷，例如，书中你会看到 FILExxxx，它表示相应值可以是 FILE0000 到 FILE9999 之间的任何值
等宽字体文字	将要在宏中输入的文本或在命令行上键入的文本与解释它们的文字区分开来是相当重要的，本书使用等宽字体来区分两者。当你看到等宽字体的文字时，就表示这些信息要出现在宏中，包含在像 CONFIG.SYS 或 AUTOEXEC.BAT 这样的系统文件中或者是需要在命令行上键入的文本。你甚至会看到 Windows 命令中的开关也使用这种风格的文本。还有一种情况使用等宽字体格式的文本。每一个代码列表均使用等宽字体，这样更易于阅读。使用等宽风格的文本还使得易于在文本中增加某些东西，比如代码示例中的缩进

标志

本书采用了许多帮助读者区分不同类型信息的标志。下文描述了各标志的用途。

注释 注释告诉读者一些有趣的事实，它们并不影响读者使用本书的其它信息。我使用注释向读者提供一些在使用 Visual C++、Windows NT 或 Windows 95 时收集到的一些有用信息。

技巧 每个人都喜欢技巧，原因在于它们会告诉读者一些完成某项任务时以前或许从未想到的方法。技巧框还提供了完成某项任务的另一条途径，读者或许会认为这一条途径比正文中提到的方法更好。

警告 它表示你要留神了。警告几乎总是要告诉读者，如果你要完成某项操作（或不能完成其它操作）时，将会引发某些系统破坏或数据损坏。在你执行警告词语之后的任何指令之前，一定要确保彻底地理解了这个警告。

Web 链接 Internet 包含了丰富的信息，但最低限度上说，找到这些信息是比较困难的。Web 链接帮助读者在 Internet 上找到能够提高读者编程能力或能够学到新的编程技巧的新的信息资源。读者还会发现一些新闻组的 Web 链接，它告诉你在什么地方可以找到谈论 Visual C++ 的其他人群。最后，Web 链接还会帮助读者找到一些能够使编程工作比以前更快、更容易的实用程序。

可移植性 保证代码能够在各种情况下和各种平台上运行总是个美好的理想。当你看到这个标志时，它提醒你一些把代码或可执行文件从一台机器上迁移到另一台机器上时可能存在的问题。如果你的代码要在多台不同类型的机器上运行或要在多种操作系统环境中运行的能力对你来说十分重要的话，一定要特别留意这种类型的标志。

高级技巧 高级技巧框中包含了一些额外的信息，即使你不知道这些信息，也并不影响你编写出优质的程序。高级技巧框为中级和高级程序员提供了一些有趣的资料，他们可以使用这些资料来增强其应用程序的功能或提高自己的生产效率。

本书中读者还会看到各种旁注。它们描述了在你开始某个过程或完成其它任务前应该记住的一段信息。旁注还包含了一些诸如文件位置或应该查看某个示例程序的某些内容这样有用的琐碎信息。在绝大多数情况下，旁注是简单实用的信息宝藏，你可以利用它提高你的整体编程水平。

匈牙利表示法概述

密码——间谍电影和其它各种人类活动的基本素材。当你第一次看到匈牙利表示法时，你可能会把它当成另一种密码。该表示法包含了密码的一切要素，其中包括一系列不得了解码的神秘字符以及使用时几乎不能破译的结果。然而，不久你就会明白，这是其他程序员的秘密代码，而不是本书中使用的匈牙利表示法。

匈牙利表示法可以为你节省大量的时间和精力。在编程方面花过大量时间的任何人都

知道，当阅读以前自己编写的代码或阅读其他人编写的代码时，好的文档是无价之宝。这也正是匈牙利表示法要为你完成的任务——文档化代码。

对匈牙利表示法的清晰理解将有助于你从本书的示例以及 Microsoft（以及其他厂商）的手册中学到更多深入的东西。每一个 Windows 编程语言厂商都在其手册中使用某种形式的匈牙利表示法。另外，相同的概念同样适用于诸如 Visual FoxPro, Delphi 以及 Visual Basic 这样的编程语言。即使语言本身完全不同，但使用匈牙利表示法编写出的代码超越了编程语言而具备一定的相似性。

那么，严格讲起来，什么是匈牙利表示法呢？匈牙利表示法是一种告诉其他人你准备如何使用变量的表示方法。知道了变量要干些什么经常有助于解释代码本身。例如，如果我告诉你某个特定变量保存了某个窗口的句柄，那么这就比变量只简单地是个变量提供了更多的信息。理解了该变量将操作窗口后，你就可以解释代码了。

这种变量命名系统第一阶段的工作是由 Microsoft 公司的 Charles Simonyi 完成的。他把这种命名系统称做匈牙利表示法（Hungarian Notation），我们在这里就使用这种称呼。你可以从许多地方获取 Charles Simonyi 著作的副本，包括 BBS 和 Internet 上的某些 Microsoft 编程站点（像 CompuServer 这样的许多联机服务也以各种形式提供匈牙利表示法的副本）。其他开发人员对 Simonyi 的研究成果做了进一步的增强。例如，Xbase 程序员使用他们自己特殊版本的匈牙利表示法。这种表示法表达了 Xbase 提供的不同类型的变量。Fresh Technologies 的 Robert A. Difalco 出版了匈牙利表示法的 Xbase 增强版本。在某些针对 DBMS 的 BBS 上以及 CompuServer 的 Computer Associates Clipper 论坛上你都可以找到这位作者的作品。

本节讲述的基本概念在前面提到的两个文档中你都可以查阅到，表达形式上可能会有所不同。本节旧话重提的目的是为了让你准确地理解我所采用的约定的意义，并说明在你

的代码中应该如何最佳地使用它们。在你的代码中采用这种命名法的原因有四个：

帮助记忆值 这种表示法有助于更容易地记住变量的名称，在团队项目中这是个要认真考虑的重要事情。

提供建议值 你或许并不是唯一要修改你的代码的人。如果你正在开发一个团队项目的話，小组中的其他成员最起码也要看一看你编写的代码。使用这些约定有助于其他开发人员理解你的代码。

一致性 程序员的工作成果通常并不仅仅体现在效率和功能方面，而且也体现在编写出的代码是否能够被其他程序员轻易地读懂。使用这些约定有助于在不同的项目中保持一致的代码风格。利用你所使用的约定，其他程序员也可以轻易地加入到修改或编写代码的行列中。

加快判断速度 在商业化世界中，创建和修改代码的速度经常会决定特定项目的成功程度。使用一致的代码将减少你花在猜测其他人创建的变量或函数意义方面所需的时间。这一判断时间的减少也就增加了你用于有效地开发产品的時間。

上面已经讲述了为什么应该使用匈牙利表示法，现在让我们看一看本书中准备如何应用这一表示法吧。我将按照下节阐述的规则命名变量。在命名数据库字段或其它与值相关的结构时我也将使用这些规则。另外，只要匈牙利表示法有助于更清晰地表达函数和过程的意义，这些函数和过程的命名也将使用下节中介绍的规则。

规则 1：使用变量前缀

变量命名时，总使用一个或两个小写字母作为变量名前缀，以指明变量的类型。绝大多数情况下，变量名前缀使用变量类型的第一个字母，因此，可以轻易地记住应该使用哪个字母。下面的示例显示了 Visual Basic ,Delphi以及 C 语言中普遍使用的前缀字母(Windows

中有成千上万种文字组合没有在这里列出)。下表也提供了几个数据库专用的修饰符：

a	Array
c	Character
d	Date
dbl	Double
dc	Device Context
dw	Double Word
f	Flag, Boolean 或 Logical
h	Handle
i	Integer
inst	Instance
l	Long
li	Long Integer
lp	Long Point
msg	Message
n	Numeric
o	Object
pal	Palette
psz	指向以零结尾的字符串的指针 (Pointer)
ptr	指针 (与其它类型变量一起使用时也可以使用 p , 比如 psz)
r	Real

续表

rc	Rectangle
rgb	Red,Green,Blue (红、绿、蓝 , 颜色变量)
rsrc	Resource
sgl	Single
si	Short Integer
sz	Zero Terminated String
u	Unsigned
ui	Unsigned Integer 或 Byte
w	Word
wnd	Window

规则 2：标识状态变量

某些变量用于指明像数据库、字段或控件这样的对象的状态。这些变量甚至可以保存其它变量的状态。告诉其他程序员某个变量用于监视某个对象的当前状态有助于他们理解这些变量在程序中的意义。你可以使用下面的三字母修饰符之一来指明该变量是状态变量：

New	新的状态
Sav	已保存状态
Tem	临时状态

规则 3：应用标准修饰符

标准修饰符几乎可以让其他人立即看清楚变量的用途。标准修饰符不提供变量的类型

信息，但它们说明了一个变量与其它变量之间的关系。例如，使用 Clr 修饰符将会告诉阅读程序的人该变量以某种方式用于操作颜色。你甚至可以把多个修饰符组合起来以进一步说明变量的作用并描述清楚应该如何使用这个变量。例如，cClrCrs 是个字符型变量，它用于确定光标的显示颜色。使用一到三个这样的修饰符通常就足以描述清楚变量的用途了。下面列出的修饰符是应用最普遍的一些修饰符：

Ar	Array
Attr	Attribute
B	Bottom
Clr	Color
Col	Column
Crs	Cursor
Dbf	Database File (数据库文件)
F	First
File	File
Fld	Field
L	Last/Left
Msg	Message
Name	Name
Ntx	Index File (索引文件)
R	right
Rec	Record Number (记录号)

续表

Ret	Return Value (返回值)
Scr	Screen
Str	String
T	Top
X	Row
Y	Column

规则 4：添加描述性文字

当你清楚地定义了变量的内容和用途后，就可以使用一些描述性的文字进一步精化这些定义。例如，你可以把指向用于保存雇员姓名的字符串的长型指针变量定义为：`lpszEmpName`。这个变量的前两个字母告诉我们它是个长型指针，紧接着的两个字母告诉我们这是个以零（或空）结尾的字符串，其余的字母告诉我们它是个雇员姓名（请注意，在这个示例中我使用了标准修饰符 `Name`）。在代码中看到这样的变量名让人一眼就可以看出其类型和用途。

规则 5：创建多个变量

时不时就会遇到在某个特定模块中使用一个变量并不能够满足每一种需要的情况。多数情况下你会想到创建同种类型的多个变量，并用数字来区分这些变量。当然你也可以使用下面的这种有意义的数字指示符来说明各变量的功能：

1 , 2 , 3	状态指针引用 , 比如 cSavClr1 , cSavClr2 , ...
Max	限制上界 , 比如 nFldMax 指明字段的最大个数
Min	限制下界 , 比如 nRecMin 指明记录的最小个数
Ord	某种类型的顺序号

目 录

引言

本书概要

本版著作新内容

系统需求

本书使用的约定

第 一 部 分

第 二 部 分

第 三 部 分

第 四 部 分

第 五 部 分

