



开发指南



Visual Basic

客户/服务器

[美] Craig Goren 著

张云明 郝宗伟 梁源松 译

熊桂喜 审校

清华大学出版社



TP393
G350

415628

Que

北京科海培训中心

Visual Basic 客户/服务器开发指南

七伦
[美] Craig Goren

著

张云明 郝宗伟 梁源松 译

熊桂喜 审校

清华大学出版社

(京)新登字 158 号

著作权合同登记号:01-96-0322

内 容 提 要

这是一本介绍用 Visual Basic 来开发客户/服务器应用程序的专业技术指导书。与一般介绍 Visual Basic 的书籍不同,本书重点介绍 Visual Basic 的一些高级技术,如客户/服务器的三级结构、OLE、远程数据对象、远程数据控制及分布式应用程序的开发技巧。书中通过三个有代表性的实例贯穿全书来介绍这些知识。

本书的读者对象是有一定 Visual Basic 及 Windows 编程经验的程序员。对于那些想要将一般应用程序升级为客户/服务器方式的技术人员来说尤其有意义。

本社出版的另一本书——《SQL Server 6.5 开发指南》是本书的姊妹篇,它们对开发客户/服务器应用程序都有很重要的指导意义。

Visual Basic 4 Enterprise Client/Server Development

Copyright ©1996 by Que  Corporation

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher.

本书英文版由 Prentice Hall 出版社属下的 Que 计算机图书出版公司于 1996 年出版。版权为 Que 公司所有。本书的中文版由 Prentice Hall 公司授予北京科海培训中心、清华大学出版社独家出版。未经出版者书面允许不得以任何方式复制或抄袭本书内容。

版权所有,盗版必究。

本书封面贴有 PRENTICE HALL 激光防伪标签,无标签者不得进入各书店。

书 名: Visual Basic 客户/服务器开发指南

作 者: [美] Craig Goren

出版者: 清华大学出版社(北京清华大学校内,邮编 100084)

印刷者: 北京门头沟胶印厂

发 行: 新华书店总店北京科技发行所

开 本: 16 印张: 41 字数: 1010 千字

版 次: 1997 年 12 月第 1 版 1997 年 12 月第 1 次印刷

印 数: 00001~5000

书 号: ISBN 7-302-02786-2/TP·1453

定 价: 69.00 元

译者序

Visual Basic 3.0 版推出以后,就成为 Windows 环境下最主要的开发环境。也正是从 3.0 版开发,Visual Basic 才为中国广大的程序员所接受,并迅速取代了 Microsoft C/C++ 及 Windows SDK,而成为主要的 Windows 开发语言。能用 Visual Basic 开发的应用程序范围很广,从简单的练习 Windows 编程,到复杂的开发企业级的大型应用软件,Visual Basic 都可适应。

从技术角度看,自 Visual Basic 3.0 开始,在 Windows 环境下编程所需的一些主要技术支持,Visual Basic 都能适应。尽管出于效率及其他原因,常常有人将 Visual C++ 与 Visual Basic 结合起来使用(笔者也推荐这种组合方法)。微软公司在 Visual Basic 3.0 后总结了来自公司其他部门及广大的 Windows 程序员的意见,推出了功能更强、更完善的 Visual Basic 4.0。Visual Basic 4.0 最主要的改进是对 OLE(对象链接和嵌入)的支持,即主要是采用构件方式来编程,并且是彻底的面向对象的。在本书即将付印时,微软公司又推出了 Visual Basic 5.0 版。Visual Basic 5.0 的主要增强之处在于对网络,尤其是对 Internet/Intranet 及 Web 的支持。

在用 Visual Basic 编制的应用程序中,非常重要的一类是基于事务处理的管理信息系统软件。因为 Visual Basic 自 3.0 开始,就内带了数据库引擎,并通过 ODBC 及其他方式,可访问其他多种数据库。特别是通过将 Visual Basic 与 Microsoft Office,尤其是与 Access 的有机结合,可让用户迅速在 Windows 环境下编制出很地道的管理信息系统软件来。有相当一段时间,国内中小型的 MIS 软件,几乎都是采用 FoxPro 或 Visual Basic+Access 编制出来的。而在两者的较量之中,Visual Basic 显然更占上风。

随着技术的进步,MIS 软件在迅速升级。局域网技术在我国的大量应用,以及最近两年来风风火火的 Internet/Intranet 网络,将 MIS 软件由原来的单用户型迅速推向网络共享型。原先由单机方式运行的数据库中的信息,迫切需要升级成网络方式的信息,存放在网络服务器上,被网上的用户所共享。这种升级,首先当然是数据库管理系统(DBMS)的升级,即由 Foxbase, FoxPro, Access 升级成 SQL Server, Oracle, Sybase DBMS。随着 DBMS 的升级,原先的 MIS 软件也必须向客户/服务器方式迁移,或改造,或重新编写。新建的 MIS 软件当然更是直接写成了客户/服务器方式的软件。

在客户/服务器方式非常流行的情况下,Visual Basic 再一次大放异彩。因为它的可视化图形界面、内带的数据库引擎(作为客户端的数据管理内核)、OLE 及众多数据库连接(如 ODBC),使得 Visual Basic 迅速成为最主要的客户/服务器开发工具。与 PowerBuilder 等工具相比,Visual Basic 更易于掌握,更富开放性,能获得更多的技术支持及文档资料支持,因而更能被广大的程序员所接受。我曾领导一个课题组,采用 Visual Basic 在较短的时间内开发出了几套企业级的 MIS 系统,开发周期短,软件质量高。

本书基于 Visual Basic 来介绍客户/服务器的开发技术。因此首先必须声明:这不是一本为初学 Visual Basic 的人准备的读物,它要求读者应具有一定的 Visual Basic 基础及软件开

发经验。可以这么说,它是为高级程序员,尤其是为要学习开发客户/服务器应用程序的技术人员准备的。

本书的内容很有特色。这主要体现在以下几个方面:

- 系统而全面地介绍了客户/服务器的概念、方法。这些概念和方法是不同于初级 Visual Basic 程序员所习惯的编程思路的。对应用程序的结构划分,本书提出了“三级客户/服务器”计算的开发方式,这是一种比以前普遍采用的两级客户/服务器计算更符合实际需求的方法。
- 全面介绍了 OLE,包括 OLE 自动化、远程过程调用等技术,并介绍了在 Visual Basic 里实现 OLE 的种种编程技巧。
- 重点介绍了在三级客户/服务器结构下,如何利用数据库技术来开发满足实际需要的大型软件。
- 从软件工程角度介绍了利用 Visual Basic 来开发大型软件(包括分布式应用)程序的一些技术,如版本控制、集体开发的开发和组织。
- 给出了三个有代表性的实例来阐述所介绍的技术。

在介绍客户/服务器开发技术时,本书引用了许多 Visual Basic 4 Enterprise 版的一些支持,如开发分布式应用程序所需的支持。但这并不是说,本书所介绍的知识只局限于 Visual Basic 4 Enterprise 版。采用 Visual Basic 3.0 或 Visual Basic 4 专业版来开发客户/服务器应用程序时,本书所介绍的知识 and 技巧也都是适用的。对于 Visual Basic 5.0,本书的内容同样实用。因为本书不是一般层次上的使用指南及编程入门指南,它介绍的是概念、方法学及一些具体技巧。

必须提醒读者注意的是,目前开发客户/服务器应用程序必须有 DBMS 作为提供数据服务的支持。编程人员必须熟悉数据库编程语言,特别是 SQL。对 Visual Basic 来说,最好最紧密的 DBMS 是 Access 及 SQL Server。这里我推荐另一本介绍 SQL Server 开发技巧的书给读者——《SQL Server 6.5 开发指南》,它同样是学习客户/服务器编程必不可少的参考书。

本书的几位译者都是使用 Visual Basic 及 SQL Server 进行客户/服务器应用程序开发的工程技术人员。在翻译过程中,阅读原书,已使开发人员受益匪浅。由于本书所涉及的技术难度大,范围广,书中难免有错。欢迎读者批评指正。

熊桂喜

1997 年 6 月于北京航空航天大学计算机系

序

从一开始,Microsoft 就把 Visual Basic 设计目标确定为要使之成为通用的 Windows 开发工具。在它的每个版本中,Microsoft 都以广大开发者容易接受的方式展现了最新的软件技术。如 Visual Basic 1.0,它使程序员不用再学习 Windows API(Windows 应用程序编程接口)中的几百个功能调用,就可生成 Windows 应用程序。而 Visual Basic 2.0 更提供了对重要的和热门的技术的使用,如多媒体、笔计算、通信等技术。Visual Basic 3.0 是第一个带有内嵌式数据库支持的 Visual Basic 版本,同时它还是第一个支持通过 OLE 集成其他应用程序的开发工具。而如今的 Visual Basic 4.0,则更进一步。通过一个直观、高效的开发工具,Visual Basic 4.0 使高性能、分布式企业应用程序的开发向我们走来了。

Visual Basic 4.0 不仅仅是对 Visual Basic 3.0 的升级。它是五年多来开发工作的最高级成果的结晶,它对产品结构完全进行了重组。甚至从 Visual Basic 1.0 一发布,Microsoft 就一直在构想和设计开放式的、基于构件的软件开发。好了,现在 Visual Basic 4.0 就完全由 OLE 构件组成。我们用 Visual Basic for Application(VBA)代替了 Visual Basic 3.0 的语言引擎,它们为处理 OLE 对象进行了优化。所有构件,包括控制构件和数据库构件,也都作为 OLE 对象进行了重新设计。

把 Visual Basic 重新构造为 OLE 构件集,给了我们很大的灵活性。我们可在不牺牲向后兼容性的情况下增强构件的功能,或者干脆换掉它,还可以引入满足开发人员特殊需要的一些新功能。一个很好的例子是 Visual Basic 4 Enterprise(Visual Basic 4 企业版)中的 Remote Data Object(RDO,远程数据对象)和 Remote Data Control(RDC,远程数据控制)。这些 OLE 对象设计的目的,是为开发人员提供对远程服务器和 Microsoft SQL Server,Oracle 或其他任何类型的 ODBC 数据源的高性能数据访问。在 Visual Basic 4 中这些功能可以作为 OLE Server(OLE 服务器)和 OLE Control(OLE 控制,OCX)实现。既然 Visual Basic 可以把 OLE 构件作为主构件,那么 Microsoft 和 ISV 均可以扩展 Visual Basic 环境,这只需要插入用来执行开发人员所需要任务的 OLE 服务器对象即可。

到目前为止,很明显的所有的 Microsoft 的操作系统、开发工具和应用程序应该继续使它们结构的核心与 OLE 相一致。描述 OLE 最好的方式就是在对象中把它作为表述接口的标准方式,以允许其他应用程序处理这些对象。OLE 将是 Microsoft 以后多年里每一种主要产品的基本结构。OLE 对象也可能来自第三方。有些 OLE 接口为特定的直接市场如银行业务、卫生保健和保险等而定义。还有一些 OLE 接口为电话业务、消息发送和数据库访问而定义。OLE 结构允许我们把这些新类型的 OLE 接口加入到 Visual Basic 中,只要它们可以使用。

现在,使用 Visual Basic 4.0 对任何可用 Visual Basic 编程的人都可创建可再使用的 EXE(可执行文件)和 DLL(动态链接库),它们表述了 OLE 接口。此特性为 Visual Basic 开发人员提供了自 Visual Basic 1.0 引入 VBX 以来他们一直梦寐以求的功能——创建、编译和发布用 Visual Basic 编写的可再使用构件的能力。与 VBX 不同,Visual Basic 4.0 创建的

OLE Automation 服务器可被所有支持 OLE Automation 的应用程序使用。现在 Visual Basic 开发人员可创建进程内(in-process) DLL 或跨进程的(Cross-process) EXE 形式的可再使用 OLE 构件。本书将详细介绍 OLE Automation 服务器的设计、编码和展开的方法。

现在读者在任意的一种计算机工业贸易出版物中都不难找到提及“三级计算”的文章。Visual Basic 4.0 Enterprise 版使创建 OLE 服务器更进了一步,它允许远程访问这些对象,并可以把它们放在位于整个 LAN 或 WAN 中任何位置的应用程序服务器上。Remote Automation 的好处在于你的对象不用再编译,就可替换调整以利用这些功能。应用程序继续调用同一对象的同一方法。我将永远不会忘记我第一次使用 Remote Automation 远程访问 Visual Basic 应用程序时的情况。当时,把用 Visual Basic 创建的 OLE 服务器之一移到了 Windows NT 机器上,然后由我的 Windows 95 客户调用它。我没有重写该应用程序代码中的任一部分。然而,不经任何修改它就在两台机器间运行。这可能会在对一个非常复杂的过程进行简化后出现,但我并没有过分简化。Remote Automation 的优点在于,它对 Visual Basic 程序开发人员掩盖了创建分布式计算对象的复杂性,而以他们已经了解的规范方式——OLE Automation 来表示。

Visual Basic 4.0 负责 OLE 和分布式计算,就像 Visual Basic 1.0 负责 Windows 应用程序开发一样,你或许认为这有些夸张。但是请回想 Visual Basic 1.0 出现之前的日子。所有的 Windows 应用程序都必须用 C 语言编写,Windows 程序员必须非常熟悉 Windows API。仅编写“Hello World”的简单程序就需要 100 行的 C 代码。而有了 Visual Basic 之后,程序开发人员只需用一行代码来输出“Hello World”。可用的 Windows 应用程序随后就大量涌现。有了 Visual Basic 4.0,用两行代码就可编写分布式的“Hello World”应用程序(即只需一行代码作为客户应用程序,而另一行作为服务器应用程序)。

我第一次听说 Clarity Consulting 是在 Visual Basic 4.0 版测试期间。在 Microsoft 为业界分析家和记者主办的 Enterprise Technical Summit 上,Craig Goren 从 Chicago 来到 Redmond,演示了用 Visual Basic 4.0 开发的一些应用程序。看过他的演示之后,我立即明白了他不仅仅会编写功能强大的应用程序,而且他已掌握了用 Visual Basic 快速创建可重用 OLE 构件的精髓。

在整个测试过程中,Craig 一直在最大限度地强调 Remote Automation。他总是通过 CompuServe 发来邮件,询问我们关于容量调整、对象池管理、安全性等问题以及他在某天突然想到的其他内容的看法。即使在 Visual Basic 4.0 开始交货后,每隔几周我还会收到他的电子邮件,告诉我一些他在为顾客编写商用对象时发现的实现其中功能的新方法。除 Visual Basic 开发小组外,我还真是很少遇到过有哪个人花费过更多的时间和精力来学习理解、设计和实现用 Visual Basic 4.0 Enterprise 来生成分布式应用程序。

Visual Basic 在创建 OLE Automation 对象和使用 OLE 来生成 Visual Basic 自身两方面的优点,使 Microsoft 可以在最新、最重要的技术出现时很快把它们纳入进来。将来的 Visual Basic 版本将在关键领域,如对象生成、增强远程数据访问、分布式应用程序等方面进行加强。当然,新类型应用程序的发布将通过 Internet 网来进行。Visual Basic 将因为它通过 OLE 提供的扩充性而成为 Internet 中有用的工具。

对 Visual Basic 开发人员的调查表明,80%的 Visual Basic 应用程序都以一定的形式访问数据库。无论应用程序是否牵涉到了企业级 LAN 网还是 Internet 网,Visual Basic 都将被

继续加强用来帮助开发人员开发那些使用了最新数据库技术的分布式数据库应用程序。尽管有了 Visual Basic 4.0, 分布式应用程序的生成和推广使其变得简单了, 但通往分布式计算之路依然充满坎坷。

在编写本书时, Craig Goren, Joy Schemelzer 和 Jeff Smith 提供了全面的指导, 以帮助读者了解分布式 OLE 应用程序的设计、生成、推广及支持。当你开始开发可重用构件时, 很快就会明白, 虽然 Visual Basic 4.0 看起来与过去版本的 Visual Basic 类似, 但它们仍有很大区别。一些新概念, 如 Class Modules (类模块), 较早的联编概念与后来的 OLE 对象联编及对象模型设计等, 都还成了要了解和处理的多种新问题。本书中所介绍的内容是基于 Clarity Consulting 和一些已使用 Visual Basic 4.0 Enterprise Edition 开发过这样的应用程序的作者的第一手资料。在我看来, 这本书无疑将很快地在你的案头占有一席之地, 并将很快成为你使用 Visual Basic 4.0 来开发和推广分布式应用程序时经常翻阅和查找的重要参考资料。

Scott Swanson

Microsoft Visual Basic 产品计划经理

1995 年 10 月

导 引

Microsoft Visual Basic 对企业客户/服务器数据库的开发人员来说再熟悉不过了。这可能要归因于它易于使用、自定义控制的扩展性以及它的灵活性。

Visual Basic 是一种通用程序设计语言。而事实上, Visual Basic 1.0 确实没有任何内带的数据库访问方法。作为通用程序设计语言, Visual Basic 与其他的 4GL 语言——如 PowerBuilder 和 SQL Windows 语言——不同, 它非常适合于开发不牵涉任何数据库访问的应用程序。

但这种独有的不同之处也是 Visual Basic 最大的缺陷——它的简单化不如它的竞争对手那样适合于大规模的客户/服务器项目的企业开发。而像 PowerBuilder 那样的工具, 则随客户/服务器数据库的流行而发展起来了。

但最终 Visual Basic 4.0 还是以先进的功能消除了这种差距——如源代码控制、集体开发、高性能数据库访问等非常适合于企业通信网客户/服务器软件开发的功能。而且, Visual Basic 4 作为第一个适合于开发三级的分布式应用程序的主流开发工具, 也在客户/服务器工具大战中加大了赌注。在加入所有这些新功能的同时, 它依然保留了它的传统优点——开放式的、易于使用的通用 Windows 程序设计语言。

Visual Basic: 企业版

Visual Basic 4.0 Enterprise Edition(企业版)在 Visual Basic 3.0 功能的基础上, 加入了使用三级结构的、分布式企业应用程序进行大规模企业开发的功能。

Visual Basic 作为最流行的基于 Windows 的开发工具已卖出了 200 多万份拷贝。而且, Visual Basic 还拥有最多的对开发工具的第三方支持者。这要归因于它的应用程序快速开发能力和它的灵活性, 可进行范围广泛的项目的开发。

Visual Basic 3.0 被专业开发人员广泛地使用, 他们占 Visual Basic 用户的 85% 以上。这些开发人员的绝大部分——约占 Visual Basic 所有销售份额的 50%——均是进行企业开发的。剩余约 35% 的使用 Visual Basic 的专业开发人员则是顾问、VAR 或 ISV 们。

Microsoft 公司也已经意识到 Visual Basic 的成功, 并正把它推广到它所有的商业应用程序中。比如, 在写作本书时, Visual Basic for Applications(VBA)已出现在大部分的 Office 应用程序套件中。Microsoft 已开始把 VBA 应用程序加入它的 Office 生产线的剩余部分和它的 BackOffice 生产线, 如 SQL Server 和 Exchange 中。

Visual Basic 是企业开发人员在创建企业客户/服务器系统所有三级构件和服务中需要了解的唯一一种开发语言。本书展示了如何创建这些服务, 即如何创建“The Visual Basic Enterprise”。

本书的目的

本书的目的在于提供关于怎样用 Visual Basic 4.0 Enterprise Edition 来构造分布式、三级企业客户/服务器应用程序的理论和操作信息。

现在有很多的关于 Visual Basic 语言和“怎样操作”的参考书。也有很多关于客户/服务器理论的书籍。本书融合了这两种概念,并提供了现实世界里三级客户/服务器的有关操作信息,这些信息均是根据坚实的理论和商业实例写成的。

本书主要介绍完成重要任务的应用程序的开发——即能够处理大数据库和许多用户的商业事务处理程序。本书着重介绍了开发人员和其他书籍容易忽略的问题——如出错处理结构、安全性设计、企业开发、版本控制和发布管理及开发策略等问题。

“本地”数据库开发不是本书的介绍重点。比如说,它没有介绍怎样用 Microsoft Jet Engine 或 Data Access Objects 进行数据库访问。相反,本书着重介绍的是新的客户/服务器的数据访问方法,这就是大家已知的 Remote Data Objects(RDO,远程数据对象)。

构造分布式客户/服务器应用程序包括许多选项。本书给出了详细的建议,并展示了怎样用例子代码来构造构件。同时本书还提供了三级客户/服务器应用程序背景理论的深厚知识,以便掌握能在你所在系统环境中作出好的决策所需的理论知识。读完本书以后,能够开发三级、企业级应用程序了,你肯定会感到高兴。

因此,以下三类读者将从本书获益:

- 熟悉使用 Visual Basic 3 的企业数据库开发人员,他们期望了解 Visual Basic 4 数据库应用的开发技术。
- 已经熟悉使用 Visual Basic 4 的企业数据库开发人员,但期望了解使用 Visual Basic 进行三级客户/服务器的开发。
- 企业数据库设计师或经理,期望了解三级客户/服务器结构。

由于本书的对象为企业开发人员,因此书中所有例子和代码均设定于假想的商业环境中。

阅读本书要具备的知识

本书介绍的功能只有 Visual Basic 4.0 Enterprise 版(VB4EE)中才具有。因此,阅读本书时,你首先应该拥有并要使用 VB4EE 的软件拷贝。

本书不是一本语言参考书。它假定你已具备了基本的 Visual Basic 程序设计知识。阅读本书前,你应该有一些使用 Visual Basic 开发应用程序的经验。但是,因为 Visual Basic 4 中大部分由 Visual Basic 3 引入的功能也适用于本书内容,所以你只需要具有使用 Visual Basic 3 的经验即可——Visual Basic 4 中所有适用于企业客户/服务器开发的新功能(如 RDO 和创建 OLE 的自动化服务器)均将在本书中详细介绍。

有许多面向 Visual Basic 语言的书可以参考。如果你不太熟悉 Visual Basic 4 和 Visual Basic 3,建议你先读一下 Que 公司的《Using Visual Basic 4》之类的书籍。但不同的是,本书是

面向应用程序的。它着重于开发三级客户/服务器应用程序的特殊技术和设计理论。本书之所以侧重于介绍数据库和 OLE 开发,主要是因为这些是用 Visual Basic 构造服务应用程序的主要技术。

我们郑重建议你准备一个能够运行范例应用程序的环境。要求如下:

- 客户工作站,运行 Windows 95(或更高的 Windows 版本)。
- 运行 Windows NT Server 3.51(或更高版本)和 SQL Server 6.0(或更高版本)的服务器工作站。
- 客户机和服务器之间的网络连接。

以上为最低配置要求,在此环境下可以测试所有的例子应用程序和 VB4EE 的功能。由于三级结构是逻辑设计而非物理结构,因此可在一台机器上运行所有级别的软件,但这样则使你无法用到 Visual Basic 4 中的一个著名的新功能——远程自动化——因为它需要网络环境。

由于许多 VB4EE 功能(比如远程数据对象)只能在 32 位环境下运行,因此本书将侧重于 32 位应用程序的开发。书的后面部分有例子展示开发 32 位环境下“关键过程”的应用程序的方法。所有例子应用程序全部用 32 位 Visual Basic 4 版生成。

读者应对数据库、关系数据库理论及 Structured Query Language(SQL,结构化查询语言)有基本的了解。大部分三级结构内容的介绍均是假定读者已经有传统的二级或文件服务器数据库应用程序开发的基本知识。

本书的内容安排

本书共分为五个部分二十四章,全面介绍了怎样用 Visual Basic 4 进行三级客户/服务器的开发的知识。本书的阅读顺序应该是从第 1 章到第 24 章。下面将给出五个部分内容的简单介绍。

第一部分:客户/服务器概念、结构和方法学简介

本部分内容给出了本书的概述,同时给出了三级服务模型的理论背景。

第 1 章“Visual Basic 企业版”介绍本书的目的和 Visual Basic 4 Enterprise Edition(企业版)中的新功能。

第 2 章“应用程序结构”先回顾了传统的应用程序结构,如集中式和文件服务器结构。接着从理论上概述了三级结构及其优点。

第 3 章“了解 OLE”给出了 OLE 的一般性介绍和在 Visual Basic 环境中的应用。它使你明白 OLE 就恰似把分布式构件联结在一起的粘接剂。

第 4 章“Visual Basic 应用程序结构”为三级结构的简短介绍。它从技术角度介绍了 The Visual Basic Enterprise 的各个部分。这一章将帮助你理解“大图像”的概念以及在深入了解结构中各专题之前各部分的关系。

第二部分:OLE

本部分从总体上介绍了 OLE 的概念,同时还特别介绍了 OLE 在 Visual Basic 中的杠杆作用。

第5章“控制 OLE 对象”介绍怎样使用 Visual Basic 来控制 OLE 对象。这些对象包括 OLE 文档、OLE 自动化服务器及 OLE 控件。

第6章“创建 OLE 对象”介绍怎样用 Visual Basic 来创建 OLE 对象。Visual Basic 可以创建 OLE 自动化服务器,然后该对象可被其他应用程序控制,比如说在 Excel, Visual Basic 或 PowerBuilder 中创建的应用程序。

第7章“管理 OLE 对象”介绍怎样管理 OLE 对象。其中介绍了远程自动化——通过网络控制 OLE 对象的一种方法。其中还介绍了 Component Manager(构件管理器),这是 Visual Basic 4 中的一种工具,可用于存储和分类各种 OLE 对象。

第三部分:服务层

第8章“第一级结构:客户应用程序”,介绍怎样在用户服务层创建构件。其中包括 GUI 设计和控制商业服务等内容。

第9章“第二级结构:商业服务器”,介绍怎样在商业服务层中创建构件。这是本书中最重要的几章之一——它介绍了怎样创建新的、商业系列的对象。它还介绍了怎样对这样的对象调整性能以进行远程自动化。

第10章“第三级结构:数据服务器”,介绍怎样在数据服务层中创建构件。这些构件中大部分是买来的而非创建的,如 SQL Server 等 DBMS 数据服务,但它也介绍了怎样平衡其他数据服务,如 Internet 和主流应用程序。

第四部分:应用程序的增强功能

第11章“远程数据对象”介绍怎样使用 RDO——Visual Basic 商业服务来访问客户/服务器 RDBMS 的最好方式。

第12章“高级编码技术”,介绍怎样在 Visual Basic 中使用这些高级编码技术——如集合、回调、可重用代码模块等知识。

第13章“安全性设计”,介绍在三级服务模型中的整体安全性设计的目标。然后还给出了怎样用 Visual Basic 4 实现这些目标的建议。

第14章“出错处理结构”,介绍在服务模型中的所有三级结构里的出错处理技术。它还给出了整体出错处理的建议标准。

第15章“在应用程序中添加帮助”,介绍怎样把用户和商业服务链接到帮助文件。它还介绍了创建帮助文件的方法。

第16章“发行应用程序”,介绍怎样使用 Setup Kit(安装工具)发布应用程序。它介绍了关于发布 OLE 对象的内容。

第17章“扩展 Visual Basic”,介绍怎样使用自定义控制、附加程序和 API 调用来扩展你自己的 Visual Basic 应用程序。

第18章“开发国际化应用程序”,讨论了开发要发布到不同场所的应用程序所涉及的问题。它还介绍了怎样使用资源文件来管理文本信息。

第19章“版本控制和集体开发”,介绍怎样使用 SourceSafe 来管理应用程序的版本和大的项目组。

第20章“确定开发原则”,介绍了开发标准的重要性,并提供了一些推荐的途径。

第21章“确定开发途径”,介绍了区别于传统开发途径的分布式、基于构件的服务的开

发途径所面临的问题和机会。

第五部分:实例应用程序

贯穿全书,实例程序很多。这些程序大部分是从与本书配套使用的磁盘中所包括的 3 个实例应用程序中节选的。这些实例应用程序在本书的最后三章中详细列了出来。

第 22 章“第一个应用程序实例:OLE 文档管理器”,这是一个三级结构的应用程序,你可以从一个中心的 DBMS 对 OLE 文档进行输入检查和输出检查。

第 23 章“第二个应用程序实例:实例管理器”,这是一个缓冲池管理器应用程序,它通过网络来管理 OLE 自动化商业服务器。

第 24 章“第三个应用程序实例:三级处理系统”,这是一个三级应用程序实例——GOLF 系统,它集成了本书中介绍的大部分结构特征。

与本书配套磁盘介绍与配置

与本书配套使用的磁盘上存放了书中应用程序实例的所有源程序代码。它按章进行组织,在根目录下每一章都有它自己的子目录。比如,你可以在磁盘目录\CHAP18 中找到第 18 章中的实例程序。第 22 章、23 章和 24 章每章都包括了一个应用程序实例。这些应用程序实例按相应章以文档形式完整存放在磁盘上。

欲购置本书配套磁盘的读者,请与北京科海培训中心联系。

联系电话:62589259 62562449

书中的排版印刷约定

为使本书使用起来更方便一些,本书使用如下排版约定来加强它的可用性:

注意: 这是一个“注意”的例子。“注意”提供一些附加信息,它们可能有助于避免问题的出现或提示你考虑使用它所提到的功能。

偶尔,你会在本书中看到程序行太长在书中一行内放不下的情景。出现这种情况时,将有继续符出现在第二行前:

```
'This is an example of a line of code that should be typed as a  
→ single line but is too long to fit onto one line of the printed  
→ page.'
```

以上几行中的→符号将代码行分成了几部分以便在印刷页的行中放置,此时不要使用 Enter 键。只需忽略该符号并把两行代码合在一起,然后再按 Enter 键。

此外,本书中在程序代码前面还加有描述代码清单的标题,以便于你在本书中容易发现和使用实例代码和应用程序文件。唯一的例外是第 22、23 和 24 章。在这几章中,完整的实例代码和应用程序文件位置都放在表中,并放在靠近每一章的开始部分。清单标题中包含了应用程序或代码实例的文件名称,你可在配套磁盘上找到它们。

目 录

第 1 部分 客户/服务器的概念、结构和方法学

第 1 章 Visual Basic 企业版..... (1)

- 1.1 关键的三级 VB4 企业版特性 (1)
 - 1.1.1 32 位开发环境 (1)
 - 1.1.2 集体开发 (1)
 - 1.1.3 商业规则封装 (2)
 - 1.1.4 发布构件 (2)
 - 1.1.5 高性能的 DBMS 数据访问 (2)
 - 1.1.6 可扩展的集成开发环境 (2)
- 1.2 开放的、基于构件的分布式解决方案 (3)
- 1.3 企业解决方案的三级模型 (4)
 - 1.3.1 用户服务 (4)
 - 1.3.2 商业服务 (4)
 - 1.3.3 数据服务 (5)
- 1.4 为何使用服务模型? (5)
- 1.5 本章小结 (5)

第 2 章 应用程序结构 (7)

- 2.1 应用程序结构 (7)
- 2.2 什么是客户/服务器 (7)
- 2.3 集中式处理结构 (8)
- 2.4 文件服务器结构 (9)
- 2.5 大多数人对客户/服务器的理解 (9)
 - 2.5.1 客户/服务器的优势 (10)
 - 2.5.2 基于文件与基于服务器 (11)
- 2.6 典型的两级客户/服务器应用程序的划分 (12)
- 2.7 两级应用程序的局限性 (12)
 - 2.7.1 难于管理前端的增强功能 (12)
 - 2.7.2 难于共享公用过程 (13)
 - 2.7.3 难于保证安全性 (13)
 - 2.7.4 用户必须使用 MIS 应用程序 (14)
 - 2.7.5 商业模型:隐藏在前端代码和后端设计之后 (14)
 - 2.7.6 无商业处理引擎 (14)
- 2.8 三级和 N 级结构——服务模型 (15)
 - 2.8.1 服务模型 (15)

- 2.8.2 构件构成的服务 (16)
- 2.8.3 客户应用程序 (17)
- 2.8.4 商业服务 (17)
- 2.8.5 数据服务 (18)
- 2.8.6 进程通信 (19)
- 2.9 一个概念性例子 (19)
- 2.10 三级结构的优点 (19)
 - 2.10.1 可管理的系统增强功能 (19)
 - 2.10.2 可共享的商业服务器 (20)
 - 2.10.3 可保证安全的商业对象和过程 (21)
 - 2.10.4 用户可创建他们自己的客户应用程序 (21)
 - 2.10.5 商业模型有它自己的一级 (22)
 - 2.10.6 商业过程引擎有一级 (22)
- 2.11 分布式系统已成为可能 (23)
- 2.12 OLE:功能强大的通信机 (24)
- 2.13 三级结构与 Visual Basic 企业版 (25)
 - 2.13.1 精心设计的系统有竞争优势 (25)
 - 2.13.2 不必每一级都是 Visual Basic (25)
- 2.14 为何采用 Visual Basic (26)
 - 2.14.1 Visual Basic 可创建 OLE 自动化服务器 (26)
 - 2.14.2 Visual Basic 中包括了远程自动化功能 (26)
 - 2.14.3 Visual Basic 可开发 32 位应用程序 (26)
 - 2.14.4 Visual Basic 具有高性能的数据服务器支持 (26)
- 2.15 其它三级结构产品 (27)
- 2.16 三级结构只是一种原则,不是必需的 (27)
- 2.17 VB 不仅仅是三级结构 (27)
- 2.18 本章小结 (28)

第 3 章 了解 OLE (29)

- 3.1 OLE 是什么 (29)
 - 3.1.1 为何要学习 OLE (29)
 - 3.1.2 OLE 与传统的 API (30)
 - 3.1.3 缩略语 (30)

3.2 OLE 的历史	(31)	4.7.2 商业服务器展开	(57)
3.3 构件对象模型	(31)	4.7.3 批处理	(59)
3.4 其它对象模型	(32)	4.7.4 访问数据服务	(60)
3.5 OLE 对象	(32)	4.8 数据服务	(62)
3.5.1 过程内与过程外	(33)	4.9 Visual Basic 库	(63)
3.5.2 对象管理	(33)	4.10 物理设计	(64)
3.5.3 对象出错处理	(33)	4.10.1 操作系统	(64)
3.6 OLE 服务	(34)	4.11 服务的推广应用	(65)
3.6.1 编组	(36)	4.11.1 用户服务	(65)
3.6.2 结构化存储	(36)	4.11.2 商业服务	(65)
3.6.3 标记	(37)	4.11.3 数据服务	(66)
3.6.4 一致性的数据传输方式	(37)	4.11.4 物理推广的例子	(66)
3.6.5 拖放式操作	(37)	4.12 三级安全性结构	(67)
3.6.6 OLE 文档	(37)	4.13 组织系统开发	(68)
3.6.7 链接	(37)	4.14 本章小结	(69)
3.6.8 嵌入	(38)		
3.6.9 原地激活	(38)		
3.6.10 OLE 自动化服务器	(39)		
3.6.11 OLE 控件	(41)		
3.6.12 分布式对象构件模型	(41)		
3.6.13 Nile(尼罗河)	(43)		
3.7 COM 服务支持	(44)		
3.8 Visual Basic 4 和 OLE	(44)		
3.8.1 控制 OLE 控件	(45)		
3.8.2 控制 OLE 自动化对象	(45)		
3.8.3 控制 OLE 文档	(46)		
3.8.4 创建 OLE 自动化对象	(46)		
3.9 OLE 的未来	(46)		
3.10 本章小结	(47)		
第 4 章 Visual Basic 应用程序结构 ...	(48)		
4.1 Visual Basic 对象应用程序	(48)		
4.1.1 程序设计模型转换	(48)		
4.1.2 面向对象的程序设计技术	(49)		
4.2 控制 OLE 对象	(49)		
4.2.1 OLE 控件	(49)		
4.2.2 OLE 自动化服务器	(50)		
4.2.3 OLE 文档	(50)		
4.3 创建 OLE 对象	(51)		
4.4 管理 OLE 对象	(52)		
4.5 将所有的服务集中在一起	(53)		
4.6 用户服务	(54)		
4.7 商业服务	(55)		
4.7.1 商业服务器设计	(55)		
		第 2 部分 OLE	
		第 5 章 控制 OLE 对象	(70)
		5.1 OLE 对象概述	(70)
		5.1.1 Visual Basic 可以控制的对象	(71)
		5.1.2 选择通信方法	(71)
		5.2 了解 OLE 对象接口	(72)
		5.2.1 对象属性	(72)
		5.2.2 对象方法	(72)
		5.2.3 对象事件	(72)
		5.2.4 成员和从属对象	(73)
		5.2.5 对象集合	(73)
		5.2.6 使用对象接口	(73)
		5.3 控制 OLE 控件	(74)
		5.3.1 实例化 OLE 控件	(74)
		5.3.2 操纵 OLE 控件的属性	(75)
		5.3.3 用 OLE 控件的方法完成动作	(76)
		5.3.4 使用 OLE 控制事件	(77)
		5.3.5 使用成员对象	(78)
		5.3.6 使用 OLE 控件的集合对象	(79)
		5.3.7 使用 OLE 控制载体属性	(81)
		5.3.8 释放 OLE 控件使用的资源	(82)
		5.4 控制 OLE 自动化对象	(82)
		5.4.1 实例化 OLE 自动化对象	(83)
		5.4.2 访问自动化对象的接口	(86)
		5.4.3 使用自动化对象的属性和方法	(86)
		5.4.4 使用 OLE 自动化对象的成员对象	(87)

5.4.5 使用 OLE 自动化对象的集合	(89)	6.3.8 创建集合	(116)
5.4.6 响应繁忙的 OLE 服务器	(89)	6.3.9 出错处理	(117)
5.4.7 释放资源	(90)	6.4 OLE 服务器设计标准	(118)
5.5 控制 OLE 文档	(91)	6.4.1 命名约定	(118)
5.5.1 链接对象	(91)	6.4.2 对象标准	(119)
5.5.2 嵌入对象	(91)	6.4.3 属性	(120)
5.5.3 创建 OLE 文档	(92)	6.4.4 方法	(120)
5.5.4 准备 OLE 载体控件	(92)	6.4.5 避免循环引用	(121)
5.5.5 设计时刻创建对象	(93)	6.5 创建一个测试应用程序	(122)
5.5.6 运行时刻创建对象	(95)	6.5.1 设计测试客户	(122)
5.5.7 在运行时刻创建嵌入对象	(96)	6.5.2 测试客户样本	(122)
5.5.8 把频繁使用的 OLE 文档放入工具箱	(97)	6.6 调试 OLE 服务器应用程序	(125)
5.5.9 操纵复合文档的方法和属性	(98)	6.6.1 使用 Visual Basic 的多个实例	(125)
5.5.10 使用 OLE 文档的成员对象	(98)	6.6.2 在错误处设置断点	(125)
5.5.11 使用 OLE 文档的集合对象	(99)	6.7 Visual Basic 开发环境	(126)
5.5.12 保存对象的数据	(99)	6.7.1 兼容 EXE	(126)
5.5.13 释放 OLE 文档对象	(100)	6.7.2 版本的确定	(128)
5.5.14 创建链接对象时的几点考虑	(100)	6.7.3 临时记录	(129)
5.6 对对象执行多种操作	(101)	6.7.4 运行时刻文件	(129)
5.6.1 With...End 语句	(101)	6.8 Visual Basic 结构	(130)
5.6.2 使用 For Each...Next 语句	(102)	6.8.1 DLL OLE 部件与 EXE OLE 部件	(130)
5.7 使用对象浏览器	(103)	6.8.2 实例化	(131)
5.7.1 了解对象浏览器接口	(103)	6.8.3 DoEvents	(132)
5.7.2 把属性和方法粘贴到代码中	(104)	6.8.4 使得 OLE 服务器应用程序 Object Browser 友好	(132)
5.7.3 用其它对象库	(104)	6.9 实现异步处理	(133)
5.8 本章小结	(104)	6.9.1 实现回调	(134)
第 6 章 创建 OLE 对象	(105)	6.9.2 轮询	(135)
6.1 什么是 OLE 自动化服务器	(105)	6.10 本章小结	(135)
6.1.1 定义	(105)	第 7 章 管理 OLE 对象	(136)
6.1.2 OLE 自动化服务器的好处	(105)	7.1 登记表和 OLE	(136)
6.2 设计 OLE 自动化服务器	(106)	7.1.1 登记表配置文件	(136)
6.2.1 定义对象和层次	(106)	7.1.2 浏览 OLE 登记表项	(137)
6.2.2 标识对象接口	(106)	7.1.3 登记对象	(137)
6.2.3 对象设计工具	(108)	7.2 登记表编辑器	(138)
6.3 开发 OLE 对象	(108)	7.2.1 登记表编辑器简介	(138)
6.3.1 工程设置	(108)	7.2.2 修改关键字及值	(139)
6.3.2 类模块	(110)	7.2.3 输入和输出登记表项	(140)
6.3.3 窗体	(111)	7.2.4 备份和恢复登记表	(141)
6.3.4 创建属性	(112)	7.2.5 Windows NT	(142)
6.3.5 从属对象	(114)	7.2.6 Windows 3.x	(142)
6.3.6 建立方法	(114)	7.3 构件管理器	(143)
6.3.7 公共函数与属性过程	(115)		

7.3.1 使用构件管理器的原因	(143)	8.8 32 位应用程序的优点	(172)
7.3.2 浏览构件	(144)	8.8.1 提高了执行速度	(172)
7.3.3 安装构件	(150)	8.8.2 更好的资源管理	(172)
7.3.4 修改构件	(152)	8.8.3 稳定性更高	(172)
7.3.5 修改构件目录	(153)	8.8.4 抢先多任务	(172)
7.4 远程自动化	(155)	8.9 推荐使用 32 位代码	(173)
7.4.1 揭开远程自动化的奥秘	(156)	8.10 访问其他用户服务	(173)
7.4.2 有关远程自动化的性能问题	(157)	8.10.1 OLE 自定义控件	(174)
7.4.3 自动化管理器	(159)	8.10.2 OLE 自动化服务器	(177)
7.5 远程自动化连接管理器	(159)	8.10.3 OLE 文档	(178)
7.5.1 服务器连接	(160)	8.10.4 DLL(动态链接库)	(178)
7.5.2 客户访问	(160)	8.11 从客户应用程序中访问商业服务	(179)
7.6 客户登记实用程序	(162)	8.12 创建图形界面	(181)
7.7 登记 OLE 对象的其他方法	(162)	8.12.1 创建 MDI 应用程序	(181)
7.8 远程自动化的调试	(163)	8.12.2 创建 SDI 应用程序	(184)
7.8.1 第 1 步:注销构件	(163)	8.12.3 合并菜单和工具栏	(190)
7.8.2 第 2 步:登记构件	(163)	8.12.4 选择边框尺寸及控制方式	(191)
7.8.3 第 3 步:本地测试	(163)	8.13 控制 Windows 95 下的系统设置	(191)
7.8.4 第 4 步:为远程自动化配置服务器工作 站	(164)	8.13.1 外观	(192)
7.8.5 第 5 步:为远程自动化配置客户工作 站	(164)	8.13.2 背景色和前景色	(193)
7.8.6 第 6 步:远程测试	(164)	8.14 实现过程中的编码原则	(193)
7.9 Internet	(164)	8.15 开发客户应用程序的途径	(194)
7.10 RegClean	(165)	8.15.1 步骤 1:收集需求	(194)
7.11 本章小结	(165)	8.15.2 步骤 2:应用程序设计	(194)
		8.15.3 步骤 3:给出应用程序第一个原型	(195)
		8.15.4 步骤 4:收集用户反馈	(195)
		8.15.5 步骤 5:改进应用程序	(195)
		8.16 使用非 Visual Basic 客户应用程序	(195)
		8.17 本章小结	(197)

第 3 部分 分级服务的层次

第 8 章 第一级结构:客户应用程序

.....	(166)
8.1 客户应用程序的位置	(166)
8.2 提供一个表示层	(166)
8.3 用户驱动的设计	(167)
8.4 开发一个核心应用程序	(168)
8.5 编写客户应用程序	(168)
8.5.1 第 1 号测试:创建一个不加入商业逻辑 的新应用程序	(169)
8.5.2 第 2 号测试:允许最终用户创建应用程 序	(169)
8.6 挑选一个控制应用程序	(169)
8.7 选择 16 位或 32 位客户应用程序	(170)
8.7.1 条件编译	(170)

第 9 章 第二结构:商业服务器

.....	(198)
9.1 什么是商业对象	(198)
9.1.1 商业对象类型	(198)
9.1.2 了解商业的粒度	(199)
9.1.3 商业对象的优点	(199)
9.2 商业服务的类型	(201)
9.2.1 批处理商业系统	(201)
9.2.2 交互式商业服务器	(201)
9.3 设计商业服务器	(202)
9.3.1 模型化商业服务器	(202)
9.3.2 封装商业规则	(204)
9.3.3 方法	(205)
9.3.4 属性	(205)