

WHAT
EVERY
VISUAL C++TM 2
PROGRAMMER
SHOULD KNOW

程序员必读

Peter D. Hipson

海洋出版社

SAMS
PUBLISHING

希望

*What Every Visual C++ 2TM
Programmer Should Know*

Visual C++ 2.0 程序员必读

[美] Peter D. Hipson 著
王 敏 等译
万 博 审校

海洋出版社

1995年·北京

内 容 简 介

本书是针对利用 Visual C++ 进行面向对象程序设计的人编写的,循序渐进地介绍了 Visual C++ 面向对象程序设计技术。本书内容以用 MFC 进行 Windows 应用程序设计为主线安排,把面向对象程序设计技术介绍得由浅入深,通熟易懂,系统全面,适合于从事面向对象程序设计的人员(特别是初学者)使用。

需要本书的读者可直接与北京海淀 8721 信箱书刊部联系,电话 2562329, 邮编 100080。

版 权 声 明

Authorized translation from the English language edition published by Sams Publishing Copyright © 1994.

Chinese language edition published by Beijing Hope Computer Company & China Ocean Press/Simon & Schuster (Asia) Pte Ltd Copyright © 1995.

本书英文版名为《What Every Visual C++ 2TM Programmer Should Know》,由 Sams Publishing 出版,版权归 Sams Publishing 所有。本书中文版由 Simon & Schuster (Asia) Pte Ltd 授权出版。未经出版者书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。

What Every Visual C++ 2TM Programmer Should Know

Visual C++ 2.0 程序员必读

[美] Peter D. Hipson 著

王 敏 等译

万 博 审校

海洋出版社出版(北京市复兴门外大街1号)

海洋出版社发行 双青印刷厂印刷

开本:787×1092 1/16 印张:23.75 字数:570千字

1995年7月第一版 1995年7月第一次印刷

印数:1~5000

*

ISBN7-5027-1778-1/TP·262

定价:45.00元

关于作者

Peten-D Hipson 和他的妻子都在 New HampShire(新罕布什尔州)生活、工作和娱乐。他自 1972 年以来都在进行计算机方面的工作,包括硬件设计和软件开发。他开发了许多 DOS 和 Windows 下的 PC 应用程序。Hipson 拥有 CPU 设计领域内的专利,并从一开始就涉足了微处理器。他是 STARmanager Windows 应用程序的开发者。当他不写书或不编写软件时, Hipson 经常在 Neu England(新英格兰)的乡村骑自行车。

致 谢

作者感谢 Sams 的 Greg Croy, Brad Jones, Greg Gunttle, Gayle Johnson 和 Ryan Rader。同时感谢 Microsoft 的 Brad Silrerberg John Brown 和 DaveBrown。还要特别感谢我的妻子，她牺牲了暑假，以便我能按时完成这本书。

序 言

本书是为不同的两部分读者而写的。第一部分读者是由那些购买了 Visual C++ 2.0 并正在寻求更多了解的人组成。第二部分读者是由那些正在认真考虑要升级到 Visual C++ 2.0 或正考虑要购买 Visual C++ 2.0 和那些在决定购买之前再看一看它的新特性的人组成。

Microsoft 在 1993 年推出 Visual C++ 1.0 时,就首先在 Windows 环境的 C/C++ 编译器方面占据了领先地位。现在,随着 Visual C++ 2.0 的推出,Microsoft 继续占领着基于 Windows 的编译器的领域的领先地位。Visual C++ 2.0 是功能强大的 32 位开发环境,能使开发人员生成 32 位的 Windows 应用程序。Visual C++ 2.0 也允许 Windows 程序员把 Windows 应用程序移植到 Apple Machintosh 68XXX 平台。

Visual C++ 2.0 向程序员提供了许多 Visual C++ 1.x 的 16 位和 32 位版本所不具有的新特性,如 Microsoft Foundation Classes (MFC) 3.0, Object Linking and Embedding (OLE) 第 2 版,OLE Custom Controls 和 Open Database Connectivity (ODBC) 的 32 位支持。Visual C++ 2.0 是 Visual C++ 产品系列的第二次重大改进,大约在 Visual C++ 1.0 之后 18 个月就发行了。

除了这些新特性外,对各种开发工具的改进也是 Visual C++ 的一部分。一般地,有了 Visual C++,就可编写具有许多杰出功能的产品。

Visual C++ 2.0 只能从 CD-ROM 上得到,因为 Microsoft 估计,要把它放在磁盘上,至少需要大约 100 张 3.5 英寸的磁盘,大多数人都不愿处理这么多的磁盘。如果确实愿意,直接在 CD 上运行这个产品比较好,这样对于这个产品只需要很少的一部分硬盘空间。除非有一个确实很快(三倍或四倍速度)的 CD 驱动器,将发现从 CD 运行这个产品比较慢。

在硬盘上完整的安装,典型地需要占据 75~100MB 的磁盘空间。本产品的许多文档和许多例子一样,在完整安装时都被安装了。然而,空间需求还不包括 BooksOnline 或附加的 MFC 或 OLE 例子,这些共有 30MB 的可执行文件;这些都保留在 CD 上。

如果不打算编译包含 Visual C++ 2.0 的示例应用程序,就可以选择来把它们留在 CD 上,这样就能减少对硬盘空间的需求。所有文档都可以用 BooksOnLine 帮助系统来使用,但比较糟糕的是 14 个不同的帮助文件。没完没了地打开帮助文件比较烦人,最好能在其他文件中看到想要的文件的主题!这已不再是一个问题,因为除了对文本和主题题目的更好的搜索,BooksOnLine 还在某一个地方给出了每一个帮助文件。

0.1 为什么越来越大

也许大家都要问一个问题:“为什么应用程序——包括编译器和开发工具——变得越来越大?”在 70 年代后期,我在一个研究生院管理一个巨型计算机中心。我梦想有一天能有一台计算机安放在桌子上,其功能与大学里的 IBM 大型机一样强大。在那时,“个人计算机”这

个词的含义是用电子元件搭起来的比较讲究的玩具。当时的技术水平,没有像图形用户界面这样的东西(只有 80×25 的字符屏幕),在按下 Enter 和对命令作出反应之间的平均字符时间可能是几分钟。

然而,今天的情况并不是很糟。我使用 Visual C++ 2.0 的主平台是由有 8MB 的 RAM 的 200 MB 硬盘和 386/33 组成。这个系统与有几个 GB 的磁盘的服务器联网,但我并不怀疑进行仔细的规划后在单台计算机上也能工作。386/33 的性能水平是最低的。推荐使用有 16MB 的 RAM 和 500MB 硬盘空间的 486。在目前的情况,这样的系统实际上不少见也不昂贵。

运行(1994 年秋季的)Visual C++ 2.0 的最实用的系统是有 32 MB 的 RAM 和 2GB 硬盘的 100MHz 的 Pentium 系统。有几种系统满足这个规范,价格在 \$3 000 至 \$4 000 之间。对于将被零售的产品的专业开发,随着生产率的提高可以以许多倍来返回其投资。在 386/33 上需要两三分钟的编译,在 100MHz 的 Pentium 上只需几秒钟就能完成。

我现在使用的计算机(在我的工作区有三台联网的计算机)在本质上化比 15 年前现代化的 IMB 大型机的功能更强。目前能用到的用户界面也好多了。在我的字处理器中,可以看到 50 到 70 行,每行有 130 个字符。使用计算机可以看到图片、地图、图样、图表、电子数据表和许多其他东西。“东西?”有一个更好的词,就是对象。我们已经对它很狂热了,一点问题也没有!

当然,Visual C++ 2.0 将是更大更好的应用程序的终结。随着对计算机的要求越来越多,为用户工作的应用程序必将越来越复杂。

在本书中,我始终把东西看作对象(现在是定义它的时候了!)。对象有多种:用户的对象,如麻烦的图形、地图、电子数据表等等;系统对象,如程序;和程序员的对象,包括 C++ 类,OLE Custom Control 和数据对象。不管喜欢不喜欢,微型计算机的世界正在变为面向对象的。

处理一个对象,如 OLE 对象,一般对于程序来说比较容易。人不必担心对象的内部格式,只需依靠对象来注意如图样这样的自身的细节。

0.2 OLE 从何而来

Object Linking and Embedding(对象链接与嵌入),一般称作 OLE,是从 Dynamic Data Exchange(动态数据交换)(DDE)发展而来的。DDE 被引入 Windows 的早期版本来允许程序直接交换数据。这种数据交换很复杂,而且很少有程序支持 DDE。Microsoft 用 OLE 实现对象转换的下一代来改进了 DDE 的缺点。OLE 的第一版是对 DDE 的主要改进,但仍然还有许多值得进一步去做。使用 OLE1,还是需要较多的编程上的努力。

OLE 在达到当前的 OLE2 标准的过程中经历了几个发现阶段。有了 OLE2,应用程序可以是客户、服务器或包容器。完整的 OLE 函数性包括如拖放,可视编辑和 OLE Automation(自动化)等支持。

0.3 ODBC

有这样的问題:应用程序对 ODBC 兼容吗?应用程序支持 OLE2 吗?尽管 Windows 应用程序还不支持它们,大多数人对这两个词的含义都有一般性的了解。

Visual C++ 2.0 中包含的一些重要特性是 ODBC SDK,OLE2 SDK 和 OLECustom Control 开发工具。这些开发者的工具就形成了 Visual C++ 2.0 有吸引力的主要部分。

当然,MFC 3.0 的增加也很重要,因为它添加了一些 Visual C++ 1.x 中没有的函数性。

0.4 本书如何组织

本书分为四个部分。第一部分是對 Visual C++ 2.0 所包含的内容的概覽。第二部分包含的是 Windows 编程。第三部分的内容是 OLECustom Control。最后一部分是第四部分,内容是 32 位 ODBC,把重点放在向现有的应用程序中包含 ODBC。

第一部分介绍 Visual C++ 2.0。

第一章介绍 Visual C++ 2.0。内容包含被改变和被添加的一般性术语。

第二章讨论使用 Visual C++ 2.0 的程序员界面。这个界面一般称作 IDE,由于添加了浮动的可定制工具条、可削减的窗口及源代码编辑的集成度增强而得到了极大的提高。由于添加了更好的建立工具,使得开发者能保留正开发的应用程序的调试版和发行版两种版本,也大大地增强了 Visual C++ 2.0 的工程文件管理工具。

第三章讲 Visual C++ 2.0 的 C/C++ 编译器 9.0 版的新特性。这个编译器可使用相似的 CL 命令语法从命令行提示来运行。

第四章考查了对 Windows 程序员的 C++ 类的界面 MFC3.0 来说是新的和改进了的东西。

第五章描述 Visual C++ 2.0 中新的调试技术。所包含的是关于 Visual C++ 2.0 的新特性 Just-In-Time 调试的信息。

第六章的重点是 Visual C++ 2.0 提供的外部实用程序。这些应用程序——如 Spy++、PViewPortTool——大大地为 Visual C++ 2.0 程序提供了便利。

第二部分介绍 Visual C++ 2.0 支持 Unicode,多线索编程和制表框(常称作属性框)。

第七章讲 Unicode,是 Windows NT3.5 中所用的新字符系统。Unicode 克服了当前的基于字符的显示系统的限制,允许大约 60 000 个字符的字符集。Unicode 只在 Chicago 下被部分支持,Chicago 在 OLE 系统中有一些 Unicode 支持。

第八章介绍多线索编程,程序员用它可以生成实现会严重影响用户的外观印象的操作的单独任务。

第九章重点介绍制表对话框(称作属性页对话框),它允许对复杂的对话框进行有效的管理。制表对话框的典型例子就是 Visual C++ 2.0 的 Class Wizard。

第三部分介绍 OLECustom Control 开发工具,及属性、方法和事件的概念。

第十章介绍 OLE 及其概念。

第十一章描述开发 OLE Control 能用的工具,包括 Control Wizard, Test Container, WPS 和登录/注销控件实用程序。

第十二章的内容是生成一个 OLE Control 以一个用 Control Wizard 开发的包容器开始,然后开发一个数字时钟,并把它包含入 Access 格式。

第十三章讨论 OLE Control 向应用程序包含。特别注意了 Access 对 OLE Control 的使用,因为 Access 是第一个支持 OLE Control 的 Windows 应用程序。

第四部分介绍 Open Database Connectivity(开型数据库联通性)(ODBC)。Visual C++ 1.5 在 16 位环境中提供 ODBC。现在,Visual C++ 2.0 支持 32 位环境。

第十四章提供了对 ODBC 的介绍,可使用下列两种方法之一来访问它:MFC 类或调用 C 的数据库函数(这些函数以 SQL 开始)。

第十五章描述 ODBC 服务器,其作用像 ODBC 允许的程序和特定的 Data Source 之间的粘合剂。每个 ODBC 驱动程序都提供标准的函数性和一些可选的增强的函数性。

第十六章介绍 SQL,是 ODBC 使用的标准化的数据库语言。ODBC 程序员在开发时间和运行时间都能生成 SQL 语句。MFC ODBC 类有许多的内建基本的 SQL 能力。

第十七章描述生成 ODBC 包容器应用程序的过程。AppWizard 可以生成管理记录移动的 必要代码。对于最小的应用程序,程序员只需使用标准的对话框控件显示 DataSource 的记录。

第十八章的重点是向现存的应用程序添加 ODBC 所需的过程。

第十九章介绍把应用程序与 DataSource 相连接的过程。

第二十章讨论从 ODBC 驱动程序获得信息的过程。

本书的末尾有五个附录。附录 A 是常用术语的词汇表。附录 B 列出命令行编译器的选项。附录 C 列出了命令行链接器选项。附录 D 列出命令行资源编译器选项。附录 E 列出了可从其他来源得到的可能改进 Visual C++ 2.0 的一些应用程序和实用程序。

目 录

关于作者	I
致谢	II
序言	III
0.1 为什么越来越大	III
0.2 OLE 从何而来	IV
0.3 ODBC	V
0.4 本书如何组织	V

第一部分 Visual C++ 简介

第一章 Visual C++ 环境	2
1.1 VISUAL C++ 各种版本之间的区别	3
1.2 小结	4
第二章 Visual C++ 2.0 的新特征	5
2.1 Visual Workbench	5
2.2 Visual Workbench 用户界面	5
2.2.1 File 菜单	5
2.2.2 Edit 菜单	9
2.2.3 Search 菜单	10
2.2.4 Project 菜单	12
2.2.5 Resource 菜单	17
2.2.6 Debug 菜单	17
2.2.7 Tools 菜单	22
2.2.8 Window 菜单	25
2.2.9 Help 菜单	26
2.2.10 创建新的工程文件	27
2.2.11 AppWizard 用户界面	27
2.2.12 WevCalc 计算器	30
2.2.13 为什么要有基于对话框的程序	34
2.3 VISUAL WORKBENCH 编辑器	34
2.3.1 编辑 C/C++ 源文件	34
2.3.2 编辑资源对象	36
2.4 对 WEVCALC 的总结	57
2.5 小结	57

第三章 新的编译器特点	58
3.1 模板	58
3.1.1 模板是什么	58
3.2 异常处理	66
3.3 结构化异常处理	67
3.3.1 C++异常处理	69
3.4 小结	71
第四章 MFC 中的新特征	72
4.1 对 32 位程序的支持	72
4.2 新的 GDI 函数	72
4.3 新的 USER 函数	72
4.4 对异常处理的支持	72
4.5 关于集合的 C++模板类	72
4.6 制表对话框(特征页)	73
4.7 更先进的图示行支持	73
4.8 Unicode 和双字节字符集(DBCS)支持	73
4.9 新的 3D 控制	74
4.10 细标题条窗口	74
4.11 为 MEC 中的命令范围或控件 IDS 赋单个处理程序	74
4.12 MFC 3.0 对 CString 添加新的函数性	74
4.13 MFC2.0 中移动时的 OLE 类支持	74
4.14 MFC2.0 中移动时对数据库交互的支持	74
4.15 小结	75
第五章 用 Visual C++ 调试	76
5.1 使用 Visual C++ 2.0 的标准调试	76
5.1.1 Disassembly 窗口	77
5.1.2 Memory 窗口	81
5.2 Just-IN-Time 调试	82
5.3 剖析	83
5.3.1 哪些函数使用最频繁	83
5.3.2 哪些函数使用了大部分的时间	83
5.3.3 哪些函数从未被调用	84
5.4 为剖析准备程序	84
5.5 小结	85
第六章 外部 Visual C++ 实用程序	86
6.1 PortTool	86
6.2 Spy++	86
6.2.1 Spy 菜单	87
6.2.2 Tree 菜单	88

6.2.3	View 菜单	88
6.2.4	Messages 菜单	88
6.2.5	Window 菜单	89
6.2.6	Help 菜单	89
6.3	DDE Spy	89
6.4	命令行编译器	89
6.5	链接	91
6.6	Lib	94
6.7	Profiler	95
6.8	BSCMAKE	95
6.9	DumpBin	96
6.10	EditBin	97
6.11	NMake	98
6.12	Resource Compiler	98
6.13	PView	99
6.14	帮助编译器	99
6.15	WinDiff	100
6.16	ZoomIn	100
6.17	Windows 的调试版本	100
6.18	MFC TRACE OPTIONS	101
6.19	Shed:分段的超图 Hotspot 编辑器	103
6.20	帮助文件	103
6.21	小结	106

第二部分 Windows 编程

第七章	使用 Unicode	108
7.1	SBCS 符集	108
7.2	DBCS 字符集	109
7.3	Unicode 字符集	110
7.4	在应用程序中包含 Unicode 支持	114
7.5	小结	115
第八章	多线程执行的编程	116
8.1	MFC 3.0 与多线程执行的程序	116
8.1.1	CWinThread()	116
8.1.2	CWinThread(AFX_ THREADPROC pfnThreadProc,LPVOID pParam)	117
8.1.3	void * m_pThreadParams	117
8.1.4	AFX_ THREADPROC m_pfnThreadProc()	117
8.1.5	void CommonConstruct()	117

8.1.6	virtual ~CWinThread()	118
8.1.7	BOOL CreateThread(DWORD dwCreateFlags=0, UINT nStackSize=0, LPSECURITY_ATTRIBUTES lpSecurityAttrs=NULL);	118
8.1.8	CWnd * m_pMainWnd	118
8.1.9	BOOL m_bAutoDelete	119
8.1.10	HANDLE m_hThread	119
8.1.11	DWORD m_nThreadID	119
8.1.12	int GetThreadPriority()	119
8.1.13	BOOL SetThreadPriority(hThread, nPriority)	119
8.1.14	DWORD SuspendThread(hThread)	119
8.1.15	DWORD ResumeThread(hThread)	120
8.1.16	virtual BOOL InitInstance()	120
8.1.17	virtual int Run()	120
8.1.18	virtual BOOL PreTranslateMessage(MSG * pMsg)	120
8.1.19	virtual BOOL OnIdle(LONG lCount)	120
8.1.20	virtual int ExitInstance() (CException * e, const MSG * pMsg)	120
8.1.21	virtual LRESULT ProcessWndProcException (CException * e, const MSG * pMsg)	120
8.1.22	virtual CWnd * GetMainWnd()	120
8.1.23	void Delete()	120
8.1.24	MSG m_msgCur	120
8.1.25	virtual BOOL PumpMessage()	121
8.1.26	virtual BOOL IsIdleMessage(MSG * pMsg)	121
8.1.27	FAXWIN.H 中的 CWinThread	121
8.1.28	线程的终止	122
8.2	Threadie 的报复机会: 一个多线程执行程序	122
8.3	小结	132
第九章	制表对话框	133
9.1	CPropertyPage	133
9.1.1	CPropertyPage(UINT nIDTemplate, UNIT nIDCaption=0)	133
9.1.2	LPCTSTR lpszTemplateName, UNIT nIDCaption=0)	133
9.1.3	void Cancel(ToClose)	133
9.1.4	void SetModified(BOOL bChanged = TRUE);	134
9.1.5	virtual BOOL OnSetActive()	134
9.1.6	virtual void OnOK();	134
9.1.7	virtual void OnCancel()	134
9.1.8	virtual BOOL PreTranslateMessage(MSG * pMsg)	134

9.1.9	AFXDLGS.H 中的 CPropertyPage	134
9.2	CPropertySheet	136
9.2.1	CPropertySheet(UINT nIDCaption, CWnd * pParentWnd= NULL, UINT iSelectPage=0)	136
9.2.2	BOOL Create(CWnd * pParentWnd=NULL, DWORD dwStyle=WS_SYSMENU WS_POPUP WS_CAPTION WS_MODALFRAME WS_VISIBLE, DWORD dwExStyle = WS_EX_DLGMODALFRAME)	136
9.2.3	int GetPageCount() const	137
9.2.4	CPropertyPage * GetPage(int nPage)const	137
9.2.5	int DoModal()	137
9.2.6	void AddPage (CPropertyPage * pPage)	137
9.2.7	void RemovePage(PropertyPage * pPage) 和 void RemovePage(int nPage)	137
9.2.8	void EndDialog (int nEndID)	137
9.2.9	AFXDLGS.H 中的 CPropertySheet	137
9.3	PropShet 程序	139
9.4	小结	146

第三部分 OLE2

第十章	OLE Custom Control 简介	148
10.1	Control Development Kit(控件开发工具)	148
10.2	几个 OLE 定义	150
10.2.1	OLE	150
10.2.2	对象	150
10.2.3	包容器	151
10.2.4	链接的对象	151
10.2.5	嵌入的对象	151
10.2.6	Windows 对象	151
10.2.7	DDE	151
10.2.8	自动化服务器	151
10.2.9	事件	152
10.2.10	服务器	152
10.3	服务器、文档和包容器之间的关系	152
10.4	什么是 OLE Control	153
10.5	在应用程序中使用 OLE Control	154
10.5.1	来自 Microsoft 的应用程序	154
10.6	小结	154

第十一章 OLE 控件开发工具	155
11.1 ControlWizard	155
11.1.1 Project Options	155
11.1.2 Control Options	156
11.1.3 Controls	157
11.1.4 创建 OLE 控件	157
11.2 Register/Unregister Control	158
11.3 Test Container	158
11.4 WPS	162
11.5 Make TypeLib	165
11.6 小结	167
第十二章 创建一个 OLE 控件	169
12.1 创建一个 OLE 控件容器	169
12.2 向 OLE 控件添加属性	171
12.2.1 库存属性:颜色	171
12.2.2 向 Clock Control 添加函数性	175
12.2.3 另外的库存属性:字体	179
12.2.4 定制属性	181
12.3 向 OLE 控件添加事件	187
12.3.1 库存事件	188
12.3.2 定制事件	190
12.4 向 OLE 控件添加方法	193
12.5 添加库存方法	194
12.6 添加定制方法	195
12.7 再多一点零碎东西	197
12.8 认可合法化	198
12.8.1 向新控件添加认可合法化	198
12.8.2 向时钟添加认可合法化	198
12.9 使用我们的 OLE 控件	200
12.10 传送 OLE 控件	200
12.11 小结	202
第十三章 向应用程序添加 OLE 控件	203
13.1 应用程序用户	203
13.2 应用程序开发人员	203
13.3 OLE 控件开发人员	203
13.4 Access 2.0 和 OLE 控件	204
13.5 小结	208

第四部分 ODBC

第十四章 介绍 ODBC	210
14.1 什么是 ODBC	210
14.2 ODBC 函数	211
14.2.1 初始化	212
14.2.2 访问数据	212
14.2.3 结束链接	214
14.3 MFC 和 ODBC	214
14.3.1 CDatabase 类	215
14.3.2 CRecordSet 类	218
14.3.3 CRecordView 类	227
14.4 ODBC 过程	229
14.5 小结	231
第十五章 ODBC 服务器	232
15.1 驱动程序考查	232
15.2 Access Data(*.MDB)	233
15.2.1 驱动程序参数	233
15.2.2 所要知道的一些事件	234
15.2.3 向 Access Datasource 的链接	234
15.2.4 SQL 函数支持	235
15.3 dBASE Files(*.DBF)	236
15.3.1 驱动程序参数	236
15.3.2 需要知道的一些事情	237
15.3.3 向 dBASE DataSource 的链接	238
15.3.4 SQL 函数支持	238
15.4 FoxPro Files(*.DBF)	239
15.4.1 驱动程序参数	239
15.4.2 要知道的一些事情	240
15.4.3 向 FoxPro DataSource 链接	241
15.4.4 SQL 函数支持	241
15.5 Paradox Files(*.DB)	242
15.5.1 驱动程序参数	242
15.5.2 要知道的一些事情	243
15.5.3 向 Paradox DataSource 链接	244
15.5.4 SQL 函数支持	244
15.6 Btrieve Data(FILE.DDF)	246
15.6.1 驱动程序参数	246
15.6.2 应知道的一些事情	246

15.6.3	向 Btrieve DataSource	247
15.6.4	SQL 函数支持	247
15.7	Excel Files(*.XLS)	248
15.7.1	驱动程序参数	248
15.7.2	需要知道的一些事情	248
15.7.3	向 Excel DataSource 链接	249
15.7.4	SQL 函数支持	249
15.8	Text Files(*.TXT, *.CSV)	250
15.8.1	驱动程序参数	251
15.8.2	要知道的一些事情	251
15.8.3	向文本文件 DataSource 链接	252
15.8.4	SQL 函数支持	252
15.9	SQL Server	253
15.9.1	驱动程序参数	253
15.9.2	需要知道的事情	253
15.9.3	向 SQL Server DataSource 的链接	254
15.9.4	SQL 函数支持	254
15.10	Oracle	255
15.10.1	驱动程序参数	255
15.10.2	需要知道的事情	255
15.10.3	向 Oracle Server DataSource 链接	255
15.10.4	SQL 函数支持	256
15.11	小结	256
第十六章	SQL 介绍	257
16.1	SQL 语句组成	257
16.1.1	ALTER DATABASE 命令	258
16.1.2	ALTER TABLE 命令	258
16.1.3	BEGIN...END 关键字	258
16.1.4	BEGIN TRANSACTION 语句	258
16.1.5	BREAK 命令	258
16.1.6	CHECKPOINT 命令	258
16.1.7	COMMIT TRANSACTION 命令	258
16.1.8	COMPUTE 关键字	259
16.1.9	CONTINUE 命令	259
16.1.10	CONVERT 命令	259
16.1.11	CREATE DATABASE 命令	259
16.1.12	CREATE DEFAULT 命令	259
16.1.13	CREATE INDEX 命令	259
16.1.14	CREATE PROCEDURE 命令	259