

精通

C#

数据库编程

王华杰 黄 山 编 著



科学出版社

www.sciencep.com

TP312C

605

精通

C#

数据库编程

王华杰 黄山 编 著



科学出版社

www.sciencep.com

内 容 简 介

本书通过一系列规模较大并具有现实意义的实例,深入系统地介绍使用 C#开发数据库应用程序的各项关键技术。

全书共分 14 章,第 1 至第 3 章是对 C#数据库开发基础性问题的深入讨论,并且介绍 C#数据库开发的新技术,第 4 至第 8 章从理论和实例两个方面深入介绍 C#中两种最重要的数据库访问方法 DAO 和 ADO.NET,随后的 5 个章节通过一个大型实例深入讲述 C#数据库编程技术及技巧。第 14 章介绍如何制作帮助文件和如何打包发行程序。本书是为那些对 C#编程有所了解并想学习 C#数据库编程技巧的读者所编写,并可作为软件开发人员的参考资料。

读者在本书使用过程中遇到的技术问题,请与电子邮件地址: jcxxt@sina.com 联系。本书所需示例程序的源代码请从网址: www.b-xr.com 下载。

需要本书或需要得到技术支持的读者,请与北京中关村 083 信箱(邮编 100080)发行部联系。电话: 010-62528991, 62524940, 62521921, 62521724, 82610344, 82675588(总机) 传真: 010-62520573, E-mail: yanmc@bhp.com.cn

图书在版编目(CIP)数据

精通 C#数据库编程 / 王华杰, 黄山编著. —北京: 科学出版社, 2003. 10

ISBN 7-03-012110-4

I. 精… II. ①王… ②黄… III. C 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 075277 号

责任编辑: 安源 杨如林 / 责任校对: 但明天

责任印刷: 媛明 / 封面设计: 王 翼

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京市媛明印刷厂印刷

科学出版社发行 各地新华书店经销

*

2003 年 10 月第 一 版 开本: 787×1092 1/16

2003 年 10 月第一次印刷 印张: 22 7/8

印数: 1—4 000 字数: 529 000

定价: 35.00 元

前 言

对于许多软件开发人员来说，也许已经对软件开发所需要的开发知识和工具及其相关书籍进行过详细的阅读和研究，但真到了联系实际的时候又不知如何下手，特别是开发大型的软件工程，总感觉到理论和实际之间还缺少点什么，这就是经验。本书就是对这个“经验”进行详细描述“指导说明书”。

本书是根据作者多年的软件开发实践经验编写而成的。全书以开发实例为主线，在基于.NET平台和C#开发工具的基础上，详细描述了数据库工程开发的具体过程。第1章概括介绍了对Visual Studio.NET的认识；第2章简述了数据库的基础知识；第3章概要介绍了结构化查询语言SQL；第4章对访问数据库的方案进行了比较；第5章介绍了C#中非常重要的数据库访问对象DAO；第6章通过一个实例详细介绍了如何应用DAO数据库访问对象访问ACCESS数据库；第7章介绍了C#中全新的数据库访问对象ADO.NET，第8章通过一个工程实例介绍如何应用ADO.NET访问数据库，第9章至第13章详细描述了一个完整的实例：图书管理借阅系统的几个主要功能模块的具体设计和实现过程，其中还包括了C#数据库程序的制作技巧和开发注意事项以及使用C#开发界面的技巧和经验；第14章对帮助文件的制作和打包发行进行了介绍。

本书由王华杰、黄山执笔编写，此外，张维、吴轶秦、韩璐、王瑾、吴君华、付鑫育、李龙、钱少伟、刘荣强、李伟光、朱峰、许大中、魏勇、萧玉、丁桦、李林、邵华刚、朱莉、肖育新、戴礼华、钟行兆、马军、李志盛、柳欢兵、关政、徐方方、钟华和王昊平等同志在整理材料方面给予了作者很大的帮助。在此，作者向他们表示深深的谢意。

本书限于作者水平，在内容选材和叙述上的不当之处，竭诚欢迎广大读者对本书提出批评和建议。

编 者

目 录

第1章 认识 Visual Studio.NET	1	2.1.7 数据库引擎	26
1.1 Visual Studio.NET 的安装	1	2.2 数据库设计理论	27
1.1.1 安装 IIS	2	2.2.1 数据库设计步骤	27
1.1.2 手动配置 FrontPage 服务器扩展 ..	3	2.2.2 需求分析	27
1.1.3 修复 .NET 框架	3	2.2.3 概念结构设计	28
1.1.4 安装消息队列服务	4	2.2.4 逻辑结构设计	29
1.1.5 安装 Visual Studio 远程调试器 ..	4	2.2.5 物理设计与实施	30
1.1.6 安装 Visual Studio 6.0 存储 过程版本控制	4	2.3 数据库设计实例	31
1.1.7 安装 SQL Server 桌面 引擎 (MSDE)	5	2.3.1 设计客户管理系统	31
1.1.8 正式开始安装 Visual Studio.NET ..	5	2.3.2 在 ACCESS 中为客户 管理系统建立数据库	33
1.2 总览 Visual Studio.NET	7	2.4 小结	36
1.2.1 总览 Visual Studio.NET	7	第3章 结构化查询语言	37
1.2.2 Visual Studio.NET 中 增加的新功能	8	3.1 SQL 语言基础	37
1.2.3 Visual Studio.NET 开发环境 的几大功能模块	9	3.1.1 SQL 的特点	37
1.2.4 Windows 应用程序的 制作流程	13	3.1.2 SQL 的优点	38
1.3 使用窗体设计器和代码设计器	14	3.1.3 SQL 语言中的常用数据类型 ..	38
1.3.1 使用窗体设计器	14	3.1.4 SQL 的标识符	40
1.3.2 使用代码设计器	14	3.1.5 SQL 中的运算符	41
1.4 使用控件工具箱	15	3.2 SQL 中常用的函数	41
1.4.1 管理“工具箱”中的 选项卡和项	15	3.2.1 日期函数	41
1.4.2 使用“工具箱”	17	3.2.2 字符串函数	41
1.5 使用属性窗口	18	3.2.3 数学函数	42
1.6 小结	19	3.2.4 集合函数	43
第2章 数据库基础	20	3.2.5 文本和图像函数	43
2.1 数据库基础知识	20	3.2.6 转换函数	44
2.1.1 数据库系统	20	3.3 运用 SQL 语言操作表、视图与索引 ..	44
2.1.2 数据模型	21	3.3.1 表的操作	44
2.1.3 关系型数据库模型	22	3.3.2 数据操作	46
2.1.4 函数依赖	22	3.3.3 创建索引	48
2.1.5 关键字	24	3.3.4 建立视图	49
2.1.6 关系规范化	24	3.4 运用 SQL 建立查询	49
		3.4.1 简单查询	49
		3.4.2 查询统计	52
		3.4.3 使用 UNION 运算符 实现多查询联合	54
		3.4.4 在查询中连接多张表	55

3.4.5	子查询	57
3.4.6	利用查询结果创建新表	58
3.5	在 Visual C# 中建立查询	58
3.5.1	Visual C# 中的查询环境	58
3.5.2	在 Visual C# 中建立关系查询	67
3.6	小结	74
第 4 章	访问数据库方案的比较	75
4.1	访问数据库技术的历史	75
4.2	访问数据库的接口方案	75
4.3	访问数据库的前景	76
4.4	小结	77
第 5 章	DAO 数据访问对象	78
5.1	DAO 对象模型	78
5.1.1	面向对象编程	78
5.1.2	DAO 对象模型	78
5.2	使用 DAO 操作数据	81
5.2.1	DBEngine 对象	81
5.2.2	Workspace 对象	83
5.2.3	Database 对象	84
5.2.4	Record 对象	85
5.2.5	创建 Record 对象	87
5.3	C# 中远程数据库的访问	90
5.3.1	数据连接	91
5.3.2	数据处理	92
5.4	小结	93
第 6 章	学生名册管理系统	94
6.1	使用 Microsoft Access 建立数据库	94
6.2	使用 DAO 读取数据库	94
6.2.1	定制开发环境	95
6.2.2	对窗体进行可视化设计	99
6.2.3	设置数据库环境	104
6.2.4	进行底层代码设计	105
6.2.5	增强功能设计	127
6.3	小结	140
第 7 章	ADO.NET 数据库访问对象	141
7.1	ADO.NET 概述	141
7.1.1	ADO.NET 简介	141
7.1.2	Connection 对象	143
7.1.3	Command 对象	144

7.1.4	DataReader 对象	145
7.2	DataSet 和 DataSetCommand 对象	149
7.2.1	ADO.NET 的 DataSet 和 ADO 的 Recordset 的比较	149
7.2.2	理解 DataSet	150
7.2.3	将 DataSet 与现有数据 一起使用	154
7.2.4	合并 DataSet 内容	160
7.2.5	复制 DataSet 内容	162
7.2.6	使用 DataSet 事件	163
7.2.7	简单的可视化设计	164
7.2.8	类型化的 DataSet	170
7.3	ADO.NET 的应用	171
7.3.1	读取数据	171
7.3.2	转换现有代码	172
7.3.3	更新数据	173
7.3.4	XML 扩展支持	174
7.4	小结	174
第 8 章	ADO.NET 开发实例	175
8.1	数据库浏览器的功能和设计思路	175
8.1.1	数据库浏览器的主要功能	175
8.1.2	数据库浏览器的设计思路	176
8.2	创建 CAccessDatabase 类	177
8.2.1	增加一个 CAccessDatabase 类 ..	177
8.2.2	设计 CAccessDatabase 类	177
8.3	创建 CDatabase 类	199
8.3.1	增加一个 CDatabase 类	199
8.3.2	设计 CDatabase 类	199
8.4	创建 AccessExplorer 窗体类	215
8.4.1	增加一个 AccessExplorer 窗体 ...	215
8.4.2	设计 AccessExplorer 类	216
8.5	创建 LoginAccess 类	230
8.5.1	增加一个 LoginAccess 窗体	230
8.5.2	设计 LoginAccess 类	231
8.6	创建 QueryAnalyzer 类	234
8.6.1	增加一个 QueryAnalyzer 窗体 ...	234
8.6.2	设计 QueryAnalyzer 类	236
8.7	创建 MainForm 类	244
8.7.1	增加一个 MainForm 窗体	244

8.7.2 设计 MainForm 类.....	245	11.2.4 学生查询功能的设计	283
8.8 完善工程.....	248	11.3 小结	288
8.9 小结	248	第 12 章 图书管理系统	289
第 9 章 图书管理借阅系统功能分析	250	12.1 界面设计	289
9.1 图书管理系统的功能分析	250	12.1.1 图书浏览功能的界面设计	289
9.2 学生管理系统的功能分析	251	12.1.2 图书编辑功能的界面设计	290
9.3 图书在线借阅系统的功能分析	252	12.1.3 图书查询功能的界面设计	291
9.3.1 图书在线借阅系统的功能	252	12.1.4 界面的美化	293
9.3.2 图书在线借阅系统的 功能抽象	253	12.2 底层代码设计	294
9.3.3 图书在线借阅系统的 辅助功能	253	12.2.1 连接数据库	294
9.4 三个子系统的在整个工程中的关系	254	12.2.2 图书浏览功能的设计	297
9.5 小结	254	12.2.3 图书编辑功能的设计	300
第 10 章 图书管理借阅系统数据库的设计	255	12.2.4 图书查询功能的设计	304
10.1 学生信息表	255	12.3 小结	309
10.2 图书信息表	256	第 13 章 图书在线借阅系统	310
10.3 图书借阅信息表	257	13.1 安装和配置 IIS 服务器	310
10.4 建立三张表之间的关联	258	13.1.1 配置 IP	310
10.4.1 在 C# 打开数据库 lib.mdb	258	13.1.2 安装 IIS 服务器	312
10.4.2 建立三张表之间的关联	260	13.1.3 配置 IIS 服务器	313
10.5 建立查询	262	13.2 建立网站	314
10.5.1 建立学生信息表中 所需要的查询	262	13.2.1 制作网站的首页	314
10.5.2 建立图书信息表中 所需要的查询	264	13.2.2 设置网站的 IP	315
10.5.3 建立借阅信息表中 所需要的查询	266	13.3 体会 WebForm	316
10.6 小结	267	13.4 设计图书在线借阅系统	318
第 11 章 学生管理系统	268	13.5 实现登录功能和在线借阅功能	318
11.1 界面设计	268	13.5.1 界面设计	318
11.1.1 学生浏览功能的界面设计	268	13.5.2 底层代码设计	322
11.1.2 学生编辑功能的界面设计	270	13.6 实现更改密码功能	336
11.1.3 学生查询功能的界面设计	270	13.6.1 设计更改密码窗体	336
11.1.4 界面的美化	273	13.6.2 设计底层代码	336
11.2 底层代码设计	274	13.7 小结	340
11.2.1 连接数据库	274	第 14 章 建立帮助文件和打包发行	341
11.2.2 学生浏览功能的设计	277	14.1 建立帮助文件	341
11.2.3 学生编辑功能的设计	279	14.1.1 创建项目	341
		14.1.2 工具按钮介绍	342
		14.2 调用帮助文件	349
		14.3 打包发行	350
		14.4 小结	357

第 1 章 认识 Visual Studio.NET

Visual Studio.NET 是支持 C#语言编程的平台，所以在学习使用 C#编写数据库之前，首先了解一下 Visual Studio.NET 是必要的。Visual Studio.NET 的开发环境不像 Turbo C 和 Microsoft C 那样简单，里面的编辑器、编译器、工程属性等许多工具都有很多设置，错误的设置可能导致正常程序不能够运行，除此之外，Visual Studio.NET 还拥有很强的代码向导功能，恰当地使用开发向导可以使程序开发达到事半功倍的效果。

下面就先来整体看看 Visual Studio.NET 的运作。

1.1 Visual Studio.NET 的安装

Visual Studio.NET 不仅是一个庞然大物（需要 3Gbyte 的物理空间），而且对计算机硬件或是系统平台都有着严格的要求，具体情况如表 1.1 所示。

表 1.1 安装 Visual Studio.NET 对计算机系统的要求

要求	企业级结构设计版	企业级开发版	专业版	学院版
处理器	具有 PentiumII 级处理器的计算机，450MHz 以上 建议：PentiumIII 级，600MHz Windows2000 Professional——96Mbyte； Windows2000 Server——192Mbyte； WindowsNT4.0 Workstation——64Mbyte； WindowsNT4.0 Server——160Mbyte；	相同	相同	相同
RAM	Windows XP Professional——160Mbyte 建议：Windows 2000Professional——128Mbyte； Windows 2000Server——256Mbyte； WindowsWorkstation——96Mbyte； Windows NT4.0Server——192Mbyte； Windows XPProfessional——192Mbyte	相同	相同	相同
可用硬盘空间	系统驱动器上有 500Mbyte，安装驱动器上有 3Gbyte	相同	相同	相同
CD-ROM 驱动 或 DVD-ROM	必选	必选	必选	必选
视频	800×600，256 色 建议：增强色（16 位）	相同	相同	相同
鼠标	Microsoft 鼠标或兼容的指点设备	相同	相同	相同

分析表格 1.1，发现 RAM 一栏中没有涉及到 Windows 98 和 Windows Me，这意味着 Windows 98 和 Windows Me 不能够运行 Visual Studio.NET。从这点就体现出它对操作系统

的苛刻。

Visual Studio.NET 拥有不同的组件，这些组件在不同的 Windows 操作系统中也有不同的要求，具体情况如表 1.2 所示。

表 1.2 Visual Studio.NET 在不同的操作系统上所需的组件

运行组件	Windows NT4	Windows 2000	Windows XP	Windows 98 Windows Me
开发 ASPWeb 应用程序和 XMLWebservices	Internet Information Server 4.0, 随 Windows NT4OptionPack 提供	Internet 信息 服务 (IIS)	IIS	不支持
编译与 MicrosoftWindows 消息队列(MSMQ) 相关的代码	Microsoft Message Queue Server 1.0, 随 Windows NT 4O ption Pack 提供	消息队列服务	消息队列服务	不支持
在远程计算机上 调试代码	Visual Studio 远程调试器	Visual Studio 远程调 试器	Visual Studio 远程 调试器	Visual Studio 远程 调试器
使用源代码管理 来控制存储过程 的版本	Visual Studio 6.0 存储过程版本控制 Visual Source Safe Microsoft SQL Server	Visual Studio 6.0 存储过程版本控制 Visual Source Safe Microsoft SQL Server	Visual Studio 6.0 存 储过程版本控制 Visual Source Safe Microsoft SQL Server	不支持

表 1.2 中显示出 Windows 98 和 Windows Me 对“在远程计算机上调试代码”这一功能的支持。即 Visual Studio.NET 开发的应用程序可以使 Windows 98 和 Windows Me 进行远程调试。

如果已经安装 Windows 2000 或以上版本，就请如表 1.2 所示，检查需要的 Windows 组件是否都已安装完毕。

下面简单介绍如何添加以上 Windows 组件。

1.1.1 安装 IIS

1. 在 Windows 2000 或 Windows XP 上安装 IIS

具体的操作步骤如下所示：

(1) 打开计算机中的“控制面板”，双击“添加/删除程序”，然后选择“添加/删除 Windows 组件”。

(2) 在“Windows 组件向导”中，选中“Internet 信息服务(IIS)”复选框。单击“下一步”开始安装。

(3) 安装完成后，在“添加/删除程序”中选择已安装的 Visual Studio.NET 产品（如 Visual Studio.NET 企业版或 Visual Basic.NET），然后单击“更改”按钮。

(4) 重新安装 Visual Studio.NET 产品。

2. 在 Windows NT 4.0 上安装 IIS

在 Windows NT 4.0 上安装 IIS, 需要安装 Windows NT 4.0 OptionPack, 可以从 <http://www.microsoft.com/NTServer/nts/downloads/recommended/NT4OptPk/default.asp> 位置下载。务必安装 Internet Information Server 4.0。

1.1.2 手动配置 FrontPage 服务器扩展

如果是在 Windows FAT16 或 FAT32 文件系统上安装, 在安装 IIS 之后必须手动配置 FrontPage 2000 服务器扩展。

1. 在 Windows 2000 和 Windows XP 上手动配置 FrontPage 服务器扩展

(1) 打开计算机中的“控制面板”, 双击“管理工具”, 然后进入“计算机管理”。

(2) 在“计算机管理”界面中, 展开“服务和应用程序”节点和“Internet 信息服务”节点。

(3) 鼠标右击“默认 Web 站点”处, 选择“所有任务”, 然后选择“配置服务器扩展”。

注意: 如果缺少“配置服务器扩展”菜单命令, 则说明已安装 FrontPage 2000 服务器扩展。

(4) 在“服务器扩展配置向导”的第一页上选择“下一步”。

(5) 在“警告”对话框中选择“是”。

(6) 对于配置邮件服务器设置选择“否”, 然后选择“下一步”, 直到选择“完成”。

2. 在 Windows NT 4.0 上手动配置 FrontPage 服务器扩展

(1) 打开计算机中的“Windows NT 4.0 OptionPack”。双击“Microsoft Internet Information Server”, 然后进入“Internet 服务管理器”。

(2) 展开“控制台根”节点和“Internet Information Server”节点。

(3) 鼠标右击“默认 Web 站点”处, 选择“任务”, 然后选择“配置服务器扩展”。

注意: 如果缺少“配置服务器扩展”菜单命令, 则说明本机已安装 FrontPage 2000 服务器扩展。可以跳过这一步骤。

(4) 在“服务器扩展配置向导”的第一页上, 选择“下一步”。

(5) 在“创建 Windows 组”页上, 选择“下一步”。

(6) 在“访问控制”页上, 选择“下一步”。

(7) 对配置邮件服务器设置选择“否”, 然后选择“下一步”。直到选择“完成”。

1.1.3 修复 .NET 框架

在 Visual Studio.NET 组件更新或安装了 IIS 之后, 必须修复 .NET 框架才能成功的创建 Web 应用程序和 XML Web Services。

修复 .NET 框架用到了 dotnetfx.exe 程序, 该程序有如下命令行选项:

- /Q ——无提示安装方式;
- /T:<完整路径> ——指定临时工作文件夹;
- /C ——(与/T 并用) 仅将文件解到指定文件夹;

- /C:<Cmd> ——忽略作者定义的安装路径。

下面分别给出了 DVD 版本和 CD 版本中修复过程：

1. 对于 Visual Studio.NET DVD

请插入该 DVD，然后在“开始”菜单上选择“运行”命令，输入 cmd 打开控制台，在控制台下进行如下操作：首先切换到光驱目录，其次输入如下所示命令。

```
cd wcu
cd dotNetFramework
dotnetfx.exe/t:c:\temp/c:"msiexec.exe/fvecmsc:\temp\netfx.msi"
```

2. 对于 Visual Studio.NET CD

请插入该 CD，然后在“开始”菜单上选择“运行”，首先切换到光驱目录，其次输入如下所示命令。

```
cd dotNetFramework
dotnetfx.exe/t:c:\temp/c:"msiexec.exe/fvecmsc:\temp\netfx.msi"
```

1.1.4 安装消息队列服务

如果您没有在计算机上安装 Microsoft 消息队列，当您试图编译使用消息队列的代码时，可能会收到指示缺少 mqa.dll 文件的错误提示。

1. 在 Windows 2000 和 Windows XP 上安装消息队列服务

(1) 打开计算机中的“控制面板”，双击“添加/删除程序”，然后选择“添加/删除 Windows 组件”。

(2) 在“Windows 组件向导”中，选中“消息队列服务”复选框。单击“下一步”开始安装。

(3) 其余步骤选择默认值。

2. 在 Windows NT 4.0 上安装消息队列服务

在 Windows NT 4.0 上安装消息队列服务，需要 Windows NT 4.0 OptionPack，可以从 <http://www.microsoft.com/NTServer/nts/downloads/recommended/NT4OptPk/default.asp> 位置下载。务必安装 Microsoft Message Queue Server1.0。

1.1.5 安装 Visual Studio 远程调试器

Visual Studio 远程调试器允许将一台或多台计算机设置为调试平台，而设置一台计算机为测试平台。它包括 SQL、TSQL、本机、托管和脚本调试所必需的文件。

1.1.6 安装 Visual Studio 6.0 存储过程版本控制

Visual Studio 6.0 存储过程版本控制允许在 Visual Source Safe 中包含存储过程，并实现源代码管理功能，但必须安装 SQL Server。

安装 Visual Studio 6.0 存储过程版本控制：

(1) 请插入 Visual Studio.NET CD1 或 DVD, 然后在“开始”菜单上选择“运行”命令, 输入 `cmd` 打开控制台。

(2) 在控制台下通过 `cd` 切换到 Visual Studio.NET CD1 或 DVD。

(3) 输入如下所示命令。

```
msiexec /qb+ /i vs_setup.msi NOVSUI=1 SERVER_SETUP=1 ADDLOCAL =
Visual_Database_Tools_Sproc_Version
```

1.1.7 安装 SQL Server 桌面引擎 (MSDE)

安装 MSDE 包括三个步骤:

- (1) 设置一个自定义的 MSDE 实例名, 这一步是可选的。
- (2) 运行 MSDE 安装程序。
- (3) 启动 MSDE 服务。

如果要使用默认实例名: Visual Studio.Net, 则可以跳过以下步骤。

1. 创建一个自定义的 MSDE 实例名

(1) 从“开始”菜单中, 选择“运行”命令。在“打开”框中, 键入 `c:\ProgramFiles\MicrosoftVisual Studio.NET\Setup\MSDE\`, 然后单击“确定”按钮。

(2) 双击 `setup.ini` 文件, 用所选的实例名替换 `INSTANCENAME=Visual StudiodotNET` 中的 `Visual StudiodotNET`。

(3) 在“文件”菜单上, 选择“保存”命令, 然后选择“退出”命令。

2. 安装 MSDE

(1) 从“开始”菜单中, 选择“运行”命令。

(2) 在“打开”对话框中, 键入 `c:\ProgramFiles\MicrosoftVisualStudio.NET\Setup\MSDE\`, 然后单击“确定”按钮。

(3) 双击 `setup.exe` 文件。

安装 MSDE 后, 必须启动 MSDE 服务。

3. 启动 MSDE 服务

(1) 从“开始”菜单中, 选择“运行”命令。

(2) 在“打开”对话框中, 键入 `NETSTARTMSSQL$<实例名>`, 然后单击“确定”按钮。

如果选择使用默认实例名, 则请输入 `NETSTARTMSSQL$Visual StudiodotNET`。

经过复杂的安装准备, 现在可以正是开始安装 Visual Studio.NET。

1.1.8 正式开始安装 Visual Studio.NET

安装时先放入第一张 CD, 弹出如图 1.1 所示的对话框。可以看到安装 Visual Studio.NET 的 3 大步骤。

第一步, Windows 组件更新。替换的组件有: IE 浏览器, 为了弥补 IE 中原有的漏洞, 使 Visual Studio.NET 构造的网络程序更加安全可靠; Jet 数据库引擎, 实现对数据库的更好连接, 这也就是 Visual Studio.NET 构造数据库对象——ADO.NET 的基础; 给 Windows 加了

补丁，弥补 Windows 几乎所有缺陷；安装了 dotNetFramework，所谓的 dotNet 网络平台。因为它是由 DLL 类库所构成，所以具有较高的性能，其编译速度是 Java 虚拟机无法比的。

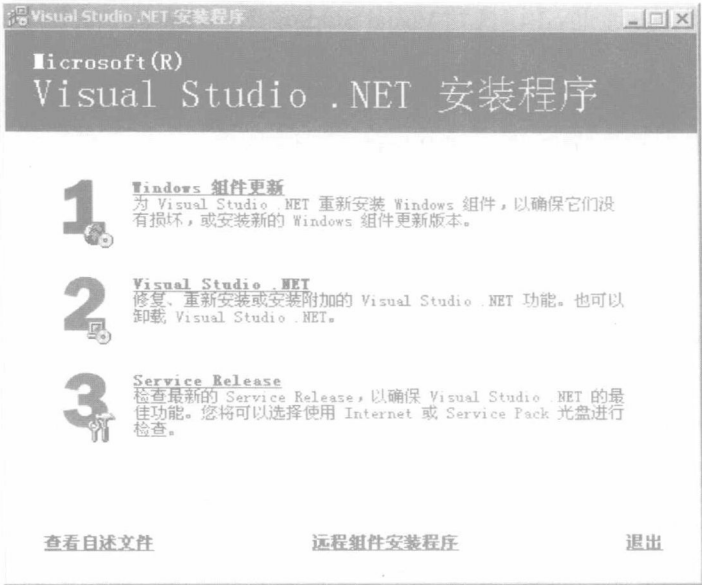


图 1.1 Visual Studio.NET 向导

注意：如果在升级的过程中出现了什么问题，向导就会提示重新升级某些组件。

第二步，如图 1.2 所示 Visual Studio.NET 的安装画面。主要是接受协议，输入产品密钥，选择要安装的程序，选择安装的位置等。

第三步，是用来升级服务的。现在 Visual Studio.NET 刚刚发布不久，所以这一项工作就没有必要做了。

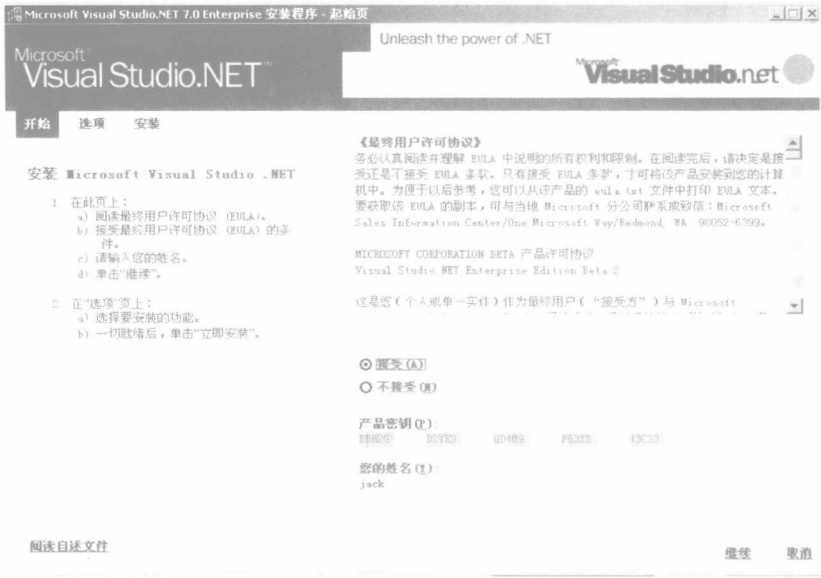


图 1.2 开始安装 Visual Studio.NET

1.2 总览 Visual Studio.NET

找到 Visual Studio.NET 的快捷项，如图 1.3 所示。



图 1.3 Visual Studio.NET 应用程序项

相对于早期的产品，Visual Studio.NET 融合了 Visual Basic 和 Visual C++，并且新创了 C# 语言。可以说把这些不同的工具集成到了同一个开发环境中。

微软为什么要把不同的语言集成到同一个开发环境之中呢？

只是为了看起来简洁吗？当然不是。微软并不是简单的把众多开发语言集成到一起，而是让这些不同的语言共同使用一个底层 DotNet 类库，即在这一层它们是利用同一种语言来编写程序的，这种中间语言的方法就从根本上实现了不同语言之间完美互换。从而大大缓解了程序员处理各种语言间相互兼容的问题。这也正是 Visual Studio.NET 开发环境的一个重要特性。

1.2.1 总览 Visual Studio.NET

如图 1.4 所示，Visual Studio.NET 与 Visual Basic 6.0 开发环境相似，只是整体风格显得更加淡雅一些。这正是微软一贯的特点：尽量保持原有产品的使用习惯，让用户容易上手，全面改进原有产品的各种缺陷。下面还是来看看隐藏在 Visual Studio.NET 的新功能吧。

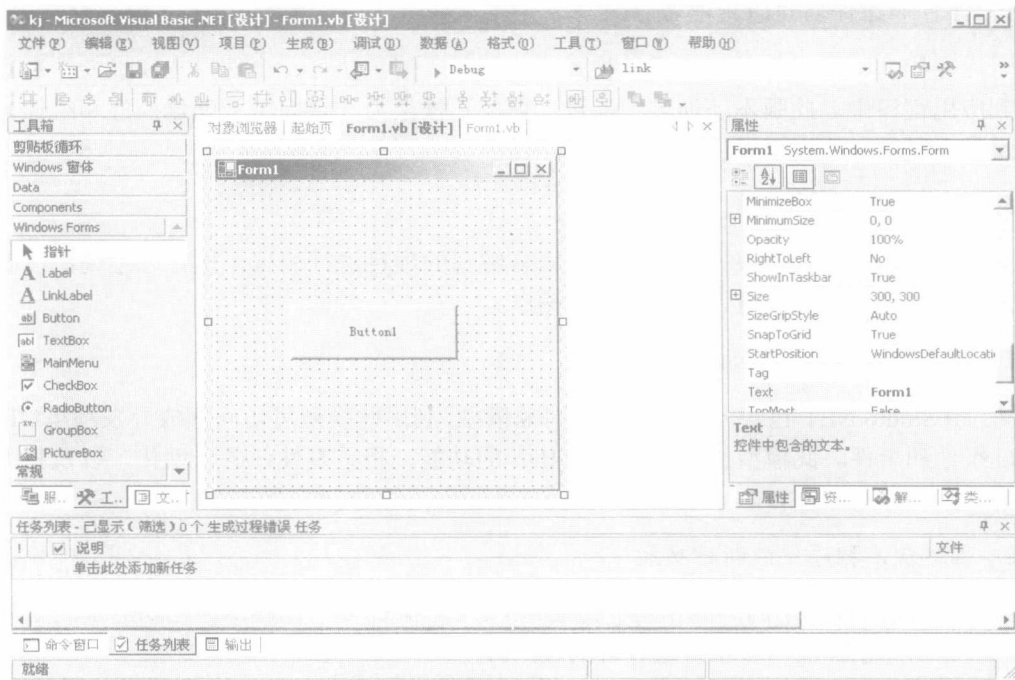


图 1.4 Visual Studio.NET 开发环境

1.2.2 Visual Studio.NET 中增加的新功能

在 Visual Studio.NET 中, Visual Basic 和 C#都具有 RAD (应用程序快速开发) 特性以及项目模板、设计器和其他的开发组件。因为这两种语言都使用 .NET 框架基类。

1. Web 开发中的新增内容

Visual Basic 和 C#都对 Web 窗体和 XML Web Services 支持。Web 窗体使你能够方便快捷地为 ASP.NET Web 应用程序创建基于浏览器的界面。XML Web Services 能够将 Visual Basic 或 C#方法打包, 使其可以通过 Web 被访问。

2. 数据中的新增功能

ADO.NET 为所有基于 .NET 框架的 Visual Studio 应用程序提供可伸缩、高性能的数据访问。可以在数据源中直接访问数据, 也可以创建内存缓存 (即数据集) 来处理已断开连接的数据。您还可以将 XML 作为关系数据直接使用, 或者直接创建并编辑 XML 文件。

3. Windows 窗体和控件中的新增功能

可以使用 Windows 窗体和控件来创建分布式应用程序的表示层。Windows 窗体提供一个有条理的、面向对象的、可扩展的类集, 它使您能够开发丰富的 Windows 应用程序。

4. 项目中的新增内容

使用此版本中的项目模板, 可以方便地创建各种 Windows 和 ASP.NET Web 应用程序和控件。项目模板将设置必要的引用, 并将命名空间和默认项导入项目。

5. 组件和组件创作的新增功能

可以使用非可视组件和相关功能将资源 (例如, 消息队列、事件日志和性能计数器) 合并到应用程序中。此版本还通过组件设计器类和框架类 (它们用于创建控件和组件) 为创建组件提供了 RAD 支持。

6. 新增的调试功能

现在, 可以对同一解决方案中以不同语言编写的项目进行调试, 执行远程调试, 以及在全部代码中实现跟踪, 检索经过微调的输出。

7. 扩展性和自动化中的新功能

Visual Studio.NET 包含一个可编程的对象模型, 利用此模型可访问集成开发环境 (IDE) 的基础组件和事件。此模型使您能够扩展 IDE 的功能, 将重复性任务自动化, 以及将 IDE 与其他应用程序进行整合。

8. 国际应用程序中的新增功能

Visual Basic.NET 中的国际化功能建立在 .NET 框架的全球化和本地化功能基础上, 使您可以很容易地创建面向多个区域设置的应用程序。

9. Visual Basic 语言的新功能

Visual Basic.NET 是新一代的 Visual Basic 开发工具,它可以简便快捷地创建.NET 应用程序(包括 XML Webservice 和 ASP.NET Web 应用程序)。

Visual Basic.NET 具有许多新功能和改进功能(如继承、接口和重载),这使 Visual Basic.NET 成为功能强大的面向对象的编程工具。其他新的语言功能包括自由线程处理和结构化异常处理。Visual Basic.NET 还完全集成了 .NET 框架和公共语言运行库, .NET 框架和公共语言运行库共同提供语言互操作性、垃圾回收、增强安全性和改进版本等支持。

10. Visual C#语言

Visual C#(读作 Csharp)是 Visual Studio.NET 中引入的一种新的编程语言。从 C++演变而来,是一种简单、现代、类型安全和面向对象的语言,使用 C#创建运行于 .NET 平台上的企业级应用程序。用 Visual C#编写的代码被编译成托管代码,意味着它将受益于公共语言运行库的支持,服务包括:语言互操作性、垃圾回收、增强的安全性以及改进版本等。

在 Visual Studio.NET 中, C#利用项目模板、设计器、属性页、代码助理、对象模型等开发组件,方便快捷地实现基于 .NET 的客户端应用程序。

下面我们从 Visual Studio.NET 开发环境着手循序渐进地学习一下 Visual Studio.NET。

1.2.3 Visual Studio.NET 开发环境的几大功能模块

回顾图 1.4,发现整个框架中有很多子窗口,这些窗口是按不同的功能来分布的,下面就对它们进行说明。如图 1.5 所示,整个框架被分成 6 大功能区域。

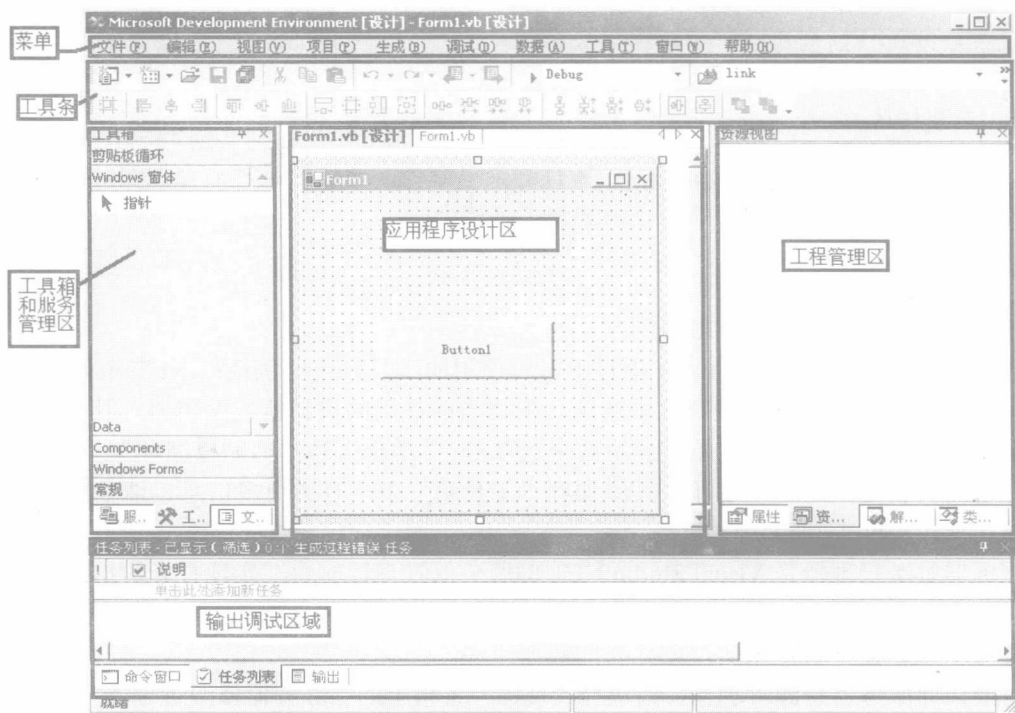


图 1.5 Visual Studio.NET 开发环境的几大功能模块

1. 菜单区

最上边的是菜单区，是标准的 Windows 功能区域。

- “文件”菜单：中包含着新建、打开、保存、退出等常见的操作。添加如：添加新项、打开解决方案等特有操作。
- “编辑”菜单：和标准 Windows 应用程序所包含的操作一样，有撤销、重复、剪贴、复制等操作。
- “视图”菜单：开启能被打开的所有子窗口的命令。
- “项目”菜单：包含很多项目管理的命令，如：添加 Windows 窗体、添加用户控件等。
- “生成”菜单：包含设置应用程序的生成状况，生成以及重新生成应用程序的可执行文件等命令。
- “调试”菜单：主要有调试、逐步执行、设置断点等操作，和显示调试时所需要的窗口。
- “数据”菜单：在编写数据库应用程序时才会用到，主要实现“生成数据集”和“预览数据”两个命令。
- “工具”菜单：包含如宏命令、生成注释 Web 页等辅助性工具。
- “窗口”菜单：包含常见的窗口管理命令，如拆分、合并、隐藏、浮动等。
- “帮助”菜单：提供 Visual Studio.NET 帮助说明，其中最有特色的一项“动态帮助”，可以提供实时帮助。

注意：对于初学者，最好打开“动态帮助”，这样有了问题就可以随时解决，而且还可以从“动态帮助”里学到很多新的知识。

2. 工具栏区

如图 1.5 所示，工具栏几乎包含了所有的菜单命令，通过工具栏命令按钮，引发菜单的功能选项。我们可以把常用的命令添加到工具栏上。

3. 工具箱和服务管理区

工具箱位于工具栏区下方最左侧。

“工具箱”包括 Visual Studio 项目中可以使用的工具，主要包括：.NET 组件、COM 组件、HTML 对象、代码片段和文本选项。根据当前使用的设计器或编辑器，其中的选项卡和选项都会有所变化。不过始终显示两个选项卡：“常规”和“剪贴板循环”。

我们可将自定义的选项卡和工具添加到“工具箱”中。有关更多信息，请参阅第 1 章第 4 节“使用控件工具箱”。

服务器资源管理器是 Visual Studio.NET 的服务器管理控制台。实现打开数据连接，登录服务器，浏览它们的数据库和系统服务等。可以将显示在其上的组件拖放到 Visual Studio.NET 设计器平台，来实现创建新的数据组件。

若要访问服务器资源管理器，请选择“视图”菜单上的“服务器资源管理器”。若要使窗口在不使用时自动关闭，请选择“窗口”菜单上的“自动隐藏”。