

Auto CAD 2000

Auto CAD 2000

二次开发技术 (ObjectARX)

余承飞 方 勇 编



人民邮电出版社

AutoCAD 2000 二次开发技术 (ObjectARX)

余承飞 方勇 编

人民邮电出版社

内 容 提 要

这是一本介绍 AutoCAD 2000 二次开发技术的书,重点讲述如何使用 ObjectARX 对 AutoCAD 2000 进行二次开发。本书从初次使用 ObjectARX 进行二次开发的编程人员的角度出发,介绍了 ObjectARX 的开发环境、ObjectARX 的基本概念以及其相关的中高级技术。对于曾经使用 ObjectARX 开发 AutoCAD R14 的编程人员,也可以有较大的帮助作用。

全书由浅入深,内容丰富新颖,易学易用,适用于所有 AutoCAD 2000 的 ObjectARX 开发人员阅读使用。

Auto CAD 2000 二次开发技术(ObjectARX)

- ◆ 编 余承飞 方 勇
责任编辑 陈 涛
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
北京朝阳展望印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本:787×1092 1/16
印张:20.25
字数:498 千字 1999 年 8 月第 1 版
印数:1-6 000 册 1999 年 8 月北京第 1 次印刷

ISBN 7-115-08036-4/TP·1259

定价:30.00 元

前言

AutoCAD 是 PC 平台上的主流 CAD 系统,不但具有强大的功能,而且提供非常开放的二次开发环境。通过对 AutoCAD 进行二次开发,可以利用 AutoCAD 内部的各种功能,为各种行业的用户开发专用的 CAD 设计环境,极大地提高设计生产的效率。

AutoCAD 2000 是 AutoCAD 的最新版本,相对于与以前的版本来说,功能有了很大的提高,增加了许多新的特性。在二次开发环境方面,也更为方便灵活。在 AutoCAD 2000 中,可以用 AutoLISP/Visual LISP、ActiveX 和 VBA、ObjectARX 等手段进行二次开发,并且可以通过 AutoCAD SQL Interface(ASI)与外部数据库连接。本书的重点是用 ObjectARX 进行二次开发。由于篇幅的限制,本书不可能讨论与 ObjectARX 有关的所有编程技术;如果读者读完这本书后,能够熟练使用 ObjectARX 进行中等水平的二次开发,本书的目的就达到了。

如果读者是第一次接触 ObjectARX 编程的话,建议按章节顺序阅读本书。最好准备好一台电脑,在必要的时候运行本书中的示例程序。很难想象在不进行实际操作的情况下能掌握 ObjectARX。

本书共有十五章,循序渐进地介绍了 AutoCAD 的二次开发(主要是 ObjectARX)。第一部分介绍了一些有关 Visual FoxPro 开发的基础知识,它是为那些初次接触 VFP 的人编写的。第一部分包括第一章至第四章。

第一章介绍了 AutoCAD 2000 的新特性,以及可用于开发 AutoCAD 2000 的几种开发环境。

第二章讲述如何配置用于 ObjectARX 开发的编程环境,并编写一个最简单的应用程序,使读者对 ObjectARX 应用程序及其开发环境有一个初步的感性认识。

第三章简要介绍了 ObjectARX SDK,使读者对 ObjectARX SDK 提供的工具、在线帮助文件、范例程序和文档有一个大概的了解。在这一章中还着重介绍了 ObjectARX Wizard 和 ObjectARX Add In 工具,它们会为读者的编程工作提供许多方便。

第四章讲述在 ObjectARX 中利用 MFC。读者可以从中体会到 ObjectARX 的灵活性和能够方便地利用外部资源的特性。

第二部分包括第五章到第十章,讲述 ObjectARX 类库、AutoCAD 数据库、数据库对象、实体等基本的 ObjectARX 编程概念,读者应该仔细阅读这些章节。如果暂时不能理解这些章节中某些概念或术语的话,可以先把它们搁置在一边,继续阅读下面的内容,以后再回来看原先不理解的内容。

第三部分包括第十一章到第十五章,属于中高级技术。使用这些技术实现的功能在 ARX 出现之前是实现不了的。

本书由方可燕策划组织,余成飞、方勇编写,韦思亮审稿。参与编写工作的还有叶青、

郭霞、方可军、周兆确、毛诗华、方海英、唐家才、段小华、陈刚、侯志东等。

由于作者水平有限，且编写时间仓促，书中难免有疏漏和错误，恳请广大读者提出宝贵意见。

目 录

第一章	AutoCAD 2000 简介	1
1.1	AutoCAD 2000 的新特性	1
1.2	AutoCAD 2000 支持的编程环境	2
1.2.1	硬件环境	2
1.2.2	开发工具	2
第二章	ObjectARX 入门	4
2.1	学习 ObjectARX 所需的知识	4
2.2	安装必要的软件	4
2.2.1	安装 ObjectARX SDK	5
2.2.2	配置 Microsoft Visual C++ 6.0	5
2.3	编写第一个 ObjectARX 应用程序	6
2.3.1	创建一个 Project	6
2.3.2	设置编译器选项	8
2.3.3	设置连接选项	8
2.3.4	添加 C++ 源文件	9
2.3.5	添加代码	10
2.3.6	创建一个 DEF 文件	13
2.3.7	加载和运行	14
2.4	ObjectARX 应用程序的结构	14
2.4.1	ObjectARX 应用程序的 AutoCAD 消息	16
2.4.2	ARX 应用程序中的事件的顺序	17
2.5	注册新的命令	20
2.5.1	命令栈 (command stack)	20
2.5.2	查找顺序	21
2.5.3	全球 (global) 与地区 (local) 命令名	22
2.5.4	透明命令	23
2.5.5	查询命令	23
2.6	示例应用程序	23
2.7	加载一个 ObjectARX 应用程序	24
2.8	卸载一个 ObjectARX 应用程序	24
第三章	了解 ObjectARX SDK	26
3.1	ObjectARX Wizard	26
3.1.1	安装 ObjectARX 2000 AppWizard	26
3.1.2	使用 ObjectARX 2000 AppWizard	27

3.2	ObjectARX AddIn.....	29
3.2.1	配置 ObjectARX AddIn	29
3.2.2	使用 ObjectARX AddIn	30
3.2.3	使用 ObjectARX Online Help	32
3.3	ObjectARX 文档和帮助文件	32
3.4	范例	33
第四章	在 ObjectARX 中使用 MFC.....	34
4.1	使用 MFC 对话框的 ObjectARX 应用程序.....	34
4.1.1	创建 ARX-MFC Project	34
4.1.2	添加对话框资源.....	34
4.1.3	编写代码.....	36
4.2	在 MFC 对话框中添加 ActiveX 控件	37
4.2.1	添加 ActiveMovie 控件.....	37
4.2.2	添加 ActiveMovie 控件的代码.....	38
4.2.3	运行.....	39
第五章	ObjectARX 类库	41
5.1	AcRx 库.....	41
5.2	AcEd 库	42
5.3	AcDb 库.....	43
5.4	AcGi 库.....	44
5.5	AcGe 库.....	44
第六章	数据库初步	46
6.1	AutoCAD 数据库.....	46
6.1.1	多个数据库.....	47
6.1.2	获取对象 ID.....	47
6.2	重要的数据库对象	47
6.2.1	创建对象.....	48
6.2.2	ObjectARX 代码示例.....	49
6.3	运行示例代码	53
第七章	数据库操作	55
7.1	初始化数据库	55
7.2	创建并向数据库中存入数据	55
7.3	存储一个数据库	56
7.4	wblock 操作	56
7.4.1	从已有的数据库创建一个新的数据库.....	56
7.4.2	用实体创建一个新的数据库.....	57
7.5	插入一个数据库	57
7.6	设定当前数据库的值	58

7.6.1	数据库颜色值.....	58
7.6.2	数据库线型值.....	58
7.6.3	数据库线型比例值.....	58
7.6.4	数据库图层值.....	59
7.7	外部引用.....	59
7.8	数据库操作示例.....	60
7.9	运行示例程序.....	62
第八章	数据库对象.....	63
8.1	打开和关闭数据库对象.....	63
8.2	删除对象.....	65
8.3	数据库的对象所有权.....	66
8.4	添加对象指定的数据.....	66
8.4.1	扩展数据.....	66
8.4.2	扩展字典.....	70
8.5	删除对象.....	77
8.6	对象归档.....	78
第九章	实体.....	79
9.1	已定义的实体.....	79
9.2	实体所有权.....	79
9.3	AutoCAD Release 12 实体.....	81
9.4	共同的(common)实体属性.....	81
9.4.1	实体颜色.....	82
9.4.2	实体线型.....	83
9.4.3	实体线型比例.....	83
9.4.4	实体可见性.....	84
9.4.5	实体图层.....	84
9.5	共同的实体函数.....	85
9.5.1	对象捕捉点.....	86
9.5.2	转换函数.....	86
9.5.3	求交点.....	87
9.5.4	GS 标记和子实体.....	88
9.5.5	炸开实体.....	106
9.6	创建 AutoCAD 实体的实例.....	108
9.6.1	创建一个简单实体.....	108
9.6.2	创建一个简单的图块表记录.....	108
9.6.3	创建有属性定义的图块表记录.....	110
9.6.4	创建一个有属性的图块引用.....	112
9.6.5	搜索一个图块表记录.....	117

9.7 复合实体	120
9.7.1 创建一个复合实体	120
9.7.2 在 polyline 中搜索顶点	122
9.8 坐标系访问	124
9.8.1 实体坐标系	124
9.8.2 AcDb2dPolylineVertex	125
9.9 曲线函数	125
9.10 将实体与超链接相关联	128
9.10.1 AcDbHyperlink 类	128
9.10.2 AcDbHyperlinkCollection 类	128
9.10.3 AcDbEntityHyperlinkPE 类	128
9.10.4 超链接示例	128
第十章 容器对象	132
10.1 符号表和字典的比较	132
10.2 符号表	133
10.2.1 图块表	135
10.2.2 图层表	135
10.2.3 搜索器	138
10.3 字典	140
10.3.1 组和组字典	141
10.3.2 Mline 形式字典	144
10.3.3 创建一个字典	144
10.3.4 搜索字典实体	145
10.4 Xrecord	147
10.4.1 Xrecord 的 DXF 组码	147
10.4.2 例子	148
10.5 版面 (Layout)	152
10.5.1 ObjectARX 版面类	153
第十一章 派生一个定制的 ObjectARX 类	155
11.1 定制类的派生	155
11.2 运行时类识别	156
11.3 类声明宏	157
11.4 类实现宏	158
11.5 类初始化函数	160
第十二章 从 AcDbObject 派生	161
12.1 重载 AcDbObject 虚拟函数	161
12.1.1 AcDbObject: 要重载的基本函数	161
12.1.2 AcDbObject: 经常被重载的函数	161

12.1.3	AcDbObject: 有时被重载的函数	162
12.1.4	AcDbObject: 很少被重载的函数	162
12.1.5	AcRxObject: 很少被重载的函数	164
12.1.6	AcDbEntity: 要重载的函数	164
12.1.7	AcDbEntity: 经常被重载的函数	165
12.1.8	AcDbEntity: 很少被重载的函数	167
12.1.9	AcDbCurve: 要重载的函数	168
12.2	实现成员函数	171
12.3	归档对象到 DWG 和 DXF 文件	171
12.3.1	dwgOut()函数	172
12.3.2	dwgIn()函数	173
12.3.3	dxOut()函数	173
12.3.4	dxIn()函数	173
12.3.5	错误检查	173
12.3.6	实现 DWG 归档函数	173
12.3.7	实现 DXF 归档函数	175
12.4	对象引用	181
12.5	所有权引用	182
12.5.1	所有权的使用	183
12.5.2	所有权的类型	183
12.5.3	建立一个所有权层次	184
12.6	指针引用	195
12.6.1	硬指针	195
12.6.2	软指针	196
12.7	清除 (Purge)	196
12.8	撤消 (Undo) 和重复 (Redo)	196
12.8.1	自动 undo	197
12.8.2	部分 undo	197
12.8.3	redo	200
12.9	subErase、subOpen、subClose 和 subCancel	200
12.10	一个定制的对象类的例子	213
12.10.1	头文件	213
12.10.2	源文件	214
12.11	对象版本支持	220
第十三章	从 AcDbEntity 派生	223
13.1	显示实体	223
13.1.1	重载 AcDbEntity::saveAs()	225
13.1.2	AcGi 对象的范围	225

13.1.3	重生成类型.....	226
13.1.4	设置实体特性.....	226
13.1.5	体素.....	232
13.1.6	线性模拟.....	244
13.1.7	等值线.....	245
13.1.8	视口.....	245
13.1.9	转换.....	245
13.2	内部实体函数.....	257
13.2.1	实现物体捕捉点函数.....	257
13.2.2	实现夹点函数.....	260
13.2.3	实现伸展点函数.....	263
13.2.4	转换函数.....	265
13.2.5	与其它实体相交.....	266
13.2.6	一个定制的实体与另一个实体相交.....	272
13.2.7	炸开一个实体.....	273
13.3	扩展实体功能.....	273
13.4	使用 AcEdJig.....	273
13.4.1	从 AcEdJig 派生一个新的类.....	274
13.4.2	使用 AcEdJig 的一般步骤.....	274
13.4.3	为拖动序列设置参数.....	274
13.4.4	拖动循环.....	275
13.4.5	实现 sampler()、update()和 entity()函数.....	276
13.4.6	添加实体到数据库.....	279
13.4.7	示例代码.....	279
第十四章	代理对象.....	286
14.1	已定义的代理对象.....	286
14.2	代理对象生命周期.....	286
14.3	用户遇到代理对象.....	287
14.4	代理实体的显示.....	287
14.5	编辑代理实体.....	288
14.6	卸载一个应用程序.....	288
第十五章	通知(Notification).....	289
15.1	通知概览.....	289
15.1.1	响应器类.....	289
15.1.2	对象响应器的类型.....	290
15.2	使用响应器.....	290
15.2.1	AcDbject 和数据库通知事件.....	291
15.2.2	定制的通知.....	292

15.2.3	使用一个编辑器响应器.....	292
15.2.4	使用一个数据库响应器.....	292
15.2.5	使用一个对象响应器.....	296
15.3	通知使用原则.....	308

第一章 AutoCAD 2000 简介

1.1 AutoCAD 2000 的新特性

AutoCAD 2000 提供了许多新的特性，增强改进了 AutoCAD。本章对这些特性进行概要的浏览。如果还不熟悉这些特性，那么在对 AutoCAD 2000 进行开发之前，应该先了解它的新特性，以便更好的利用它们。

可以在 AutoCAD 2000 中选择菜单 Help / What's new 来查看这些新特性的详细列表。这里列举了对 AutoCAD 二次开发有较大影响的新特性，如表 1-1 所示。

表 1-1 AutoCAD 2000 的新特性

特性	简要描述
多文档环境	多文档环境是 AutoCAD 2000 的最重要的新特性，也是对 AutoCAD 二次开发影响最大的特性。它使用户可以在一个 AutoCAD 进程中打开多张图，并在它们之间拷贝、移动和绘制对象，以及将它们之间的数据相关联
AutoCAD DesignCenter	这是非常好用和直观的工具，在编程和调试程序时可以很方便地查看图形的各种属性
快速尺寸标注	使用新的 QDIM 命令产生尺寸标注
新的物体捕捉方式	使用新的“平行”和“扩展”物体捕捉方式
AutoTrack	使用极坐标和物体捕捉跟踪来在指定极坐标角度或在与物体捕捉点相关的角度生成对象
部分打开和部分加载	只打开与编辑图形、Xref 的需要的部分
实时 3D 旋转	使用新的 3DORBIT 命令轻松地处理 3D 物体的视图
多个活动的工作平面	视口和视图可以有不同的用户坐标系 (UCS) 和标高 (elevation) 设置，使在 3D 中工作更容易
引用编辑	在当前图形中编辑外部引用和图块引用
IntelliMouse	使用 Microsoft IntelliMouse，包括 IntelliMouse 平移和放大
快速选择	使用快速选择，基于对象类型或对象属性选择对象
快捷菜单	右键单击指点设备显示新的快捷菜单
长名称对象名	突破 31 个字符的限制。可以长达 255 个字符，可包含空格

续表

特性	简要描述
加载/卸载应用程序	使用了新的加载/卸载对话框
通过 Web 访问文件	使用 Web 访问和存储 AutoCAD 数据
超链接	附加超链接到 AutoCAD 对象或图形定位上
EPlot	用安全、可用于 Internet、中性的格式生成图形的一个电子出图 (或版面)
非矩形视口	创建非常规形状的视口
Lineweight	使用新的 Lineweight 属性向对象添加宽度

1.2 AutoCAD 2000 支持的编程环境

本节介绍 AutoCAD 2000 所需的硬件环境和它所支持的编程环境。

1.2.1 硬件环境

AutoCAD 2000 现在只能在 PC 机上运行,目前支持 Windows95、Windows98 或 Windows NT 4.0 操作系统。它要求的推荐硬件配置是:

- Pentium 133 或更好 (或兼容处理器)
- 64 MB RAM (至少 32MB)
- Windows98、Windows 95 或 Windows NT 4.0 操作系统
- 1024 x 768 VGA 视频显示 (至少 800 x 600)
- Windows 视频显示驱动
- 130MB 硬盘空间和 64MB 交换空间
- 指点设备 (鼠标或有 Wintab 驱动的数字仪)
- 四速 CD-ROM 驱动器
- IBM 兼容并行口
- 串口 (用于数字仪或某些绘图仪)
- 打印机或绘图仪
- Modem (连接到 Internet, 非必须)
- 声卡, 用于多媒体学习
- TCP/IP 或 IPX 支持 (只对于多用户才需要)

1.2.2 开发工具

目前有三种软件开发环境可用于 AutoCAD 2000 的开发: AutoLISP/Visual LISP、VBA 和 ObjectARX。分别介绍如下。

1. AutoLISP / Visual LISP

AutoLISP 作为 AutoCAD 的二次开发工具历史已经很久了。从 R14 开始, AutoCAD 提供了 Visual LISP 开发工具, 它实际上是 AutoLISP 的编程开发环境, 在很大程度上克服了 AutoLISP 原来的效率低和保密性差的缺陷。Visual LISP 编程环境已经嵌入 AutoCAD 2000 中, 在 AutoCAD 2000 的菜单上选择 Tools / AutoLISP / Visual LISP Editor 就可以进入 Visual LISP 开发环境。

2. VBA

VBA 是 Visual Basic for Applications programming environment 的缩写。AutoCAD ActiveX 提供一个机制, 从 AutoCAD 内部或外部程序化地处理 AutoCAD。它通过向“外部世界”揭示 AutoCAD 对象来作到这一点。一旦这些对象被揭示, 它们就可以被许多不同的编程语言和环境, 以及其它应用程序如 Microsoft Word VBA 或 Excel VBA 访问, 如图 1-1 所示。

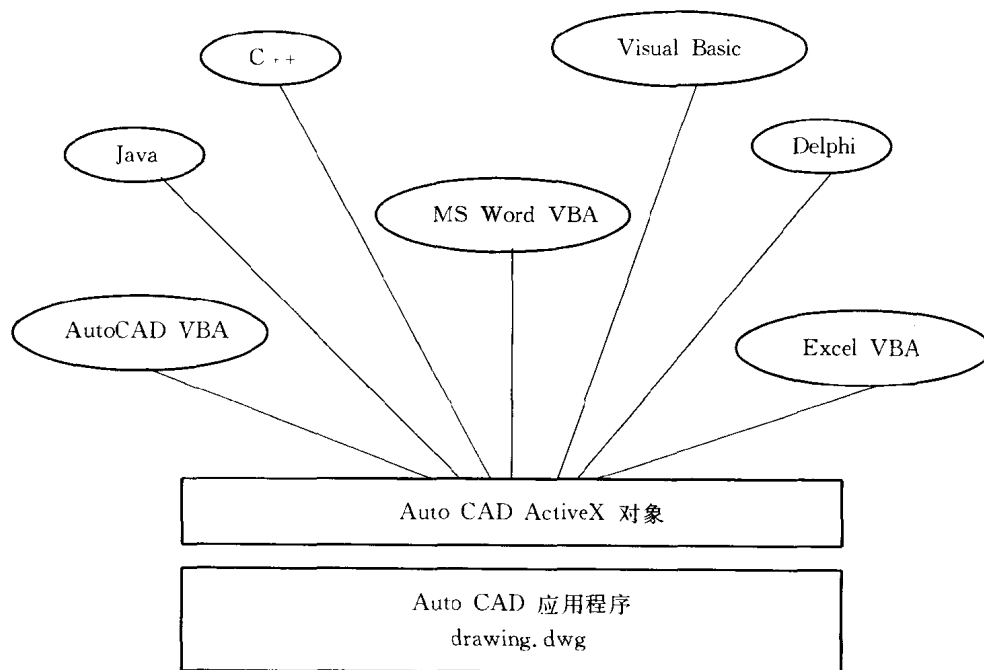


图 1-1 AutoCAD VBA

AutoCAD 2000 本身就具有 VBA 开发环境。键入 VBAIDE 就会进入 VBA 集成开发环境。

3. ObjectARX

这是对 AutoCAD 进行二次开发的最强有力的开发工具。用于开发 AutoCAD 2000 的 ObjectARX 2000 目前只支持 Microsoft Visual C++ 6.0 编译器。但是, 它也比前两种工具更繁琐、更难以掌握。本书将重点介绍使用 ObjectARX 对 AutoCAD 进行二次开发。

第二章 ObjectARX 入门

一个 ObjectARX 应用程序是一个 DLL (动态连接库), 共享 AutoCAD 的地址空间, 对 AutoCAD 进行直接函数调用。为了增强可扩展性, ObjectARX 包括了易于定义新的类的宏和在运行时向库中现有的类添加函数的功能。ObjectARX 库可以与 ADS 及 AutoLISP 联合使用。

2.1 学习 ObjectARX 所需的知识

如果想学习用 ObjectARX 对 AutoCAD 进行二次开发, 当然要具备一些基本知识。包括:

- C/C++ 知识, 清楚面向对象编程的概念, 并且了解 Microsoft Visual C++ 的使用。如果读者以前习惯用其它的 C/C++ 编译器, 那也不成问题; 各种编译器本来就有许多共同点, 而且 Microsoft Visual C++ 是很易于使用的, 现在开始学习使用它也不成问题。
- AutoCAD 的基本知识和使用经验。很难想象从来没有用过 AutoCAD 的人能够从事 ObjectARX 开发, 但是也不是非要对 AutoCAD 非常熟悉不可。只要了解 AutoCAD 基本命令和操作方法, 再有一些使用经验就可以了。
- 如果读者曾经用 AutoLISP 或 ADS 开发过 AutoCAD, 那就太好了。会更迅速地学会 ObjectARX。

2.2 安装必要的软件

要使用本书学习 ObjectARX 编程, 就要准备好以下软件:

- Windows NT 4.0、Windows95 或 Windows98。Autodesk 推荐您使用 Windows NT 4.0, 因为虽然在 Windows95 或 Windows98 中也可以开发和运行 ARX, 但 Windows NT 4.0 是唯一被支持的可以用 Visual C++ 可靠地进行 debug 的开发平台。一些 AutoCAD 的动作会导致在 Windows95 或 Windows98 中的 Visual C++ debug 进程将系统锁死或表现异常。详细信息请参阅 Visual C++ 6.0 的根目录下的 readmevc.htm 中的“Cannot Debug Fibers on Windows 98”主题。但是, 这个限制只是针对软件开发。所生成的应用程序可以在 Windows95/98 中很好地运行。本书并不要求您一定要安装 Windows NT 4.0, 因为大部分操作实例并不需要使用

debug。然而，如果您真正准备用 ObjectARX 开发 AutoCAD 软件的话，建议您还是使用 Windows NT 4.0 平台。

- AutoCAD 2000。
- Microsoft Visual C++ 32 bit Edition Release 6.0。这是 ObjectARX 2000 开发唯一支持的编译器和连接器。
- ObjectARX SDK for AutoCAD 2000。这是 Autodesk 提供的，可以到 www.autodesk.com 下载它。也可以到编者的个人主页 ycf888.yeah.net 下载 ObjectARX SDK for AutoCAD 2000 以及与本书有关的其它资料。请注意，ObjectARX SDK for AutoCAD R14 不能在 AutoCAD 2000 中使用。

2.2.1 安装 ObjectARX SDK

安装好 AutoCAD 和 Microsoft Visual C++ 之后，就可以着手安装 ObjectARX SDK。本书假设把它安装到了 C:\ObjectARX 2000 目录下。注意安装前先删去 Uninstall 老版本的 ObjectARX SDK（如果曾经安装过的话）。安装完毕后，将 REDISTRIB 目录下的 acbr15.dll、acgex15.dll 以及其它应用程序可能需要的 DLL 放置到 Acad.exe 所在的 AutoCAD 主目录下。

2.2.2 配置 Microsoft Visual C++ 6.0

下面要为 ObjectARX SDK for AutoCAD 2000 配置 Microsoft Visual C++ 6.0。步骤如下：

1. 启动 Visual C++，选择在“Tools”菜单中的“Options”项。
2. 选择“Directories”标签，并在“Show directories for”栏中选择“Include files”，如图 2-1 所示。
3. 点击“new”按钮。这时出现一个新的条目栏。

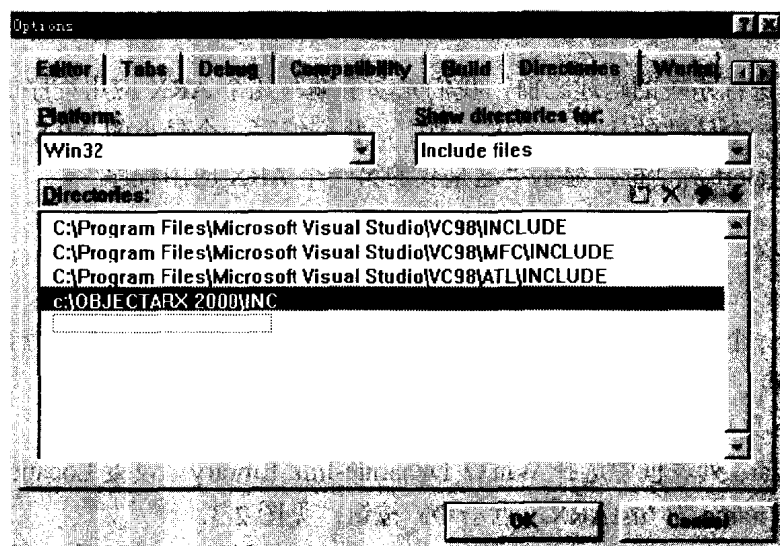


图 2-1 配置 Visual C++ 的 include 目录