

Auto CAD 2000

附光盘



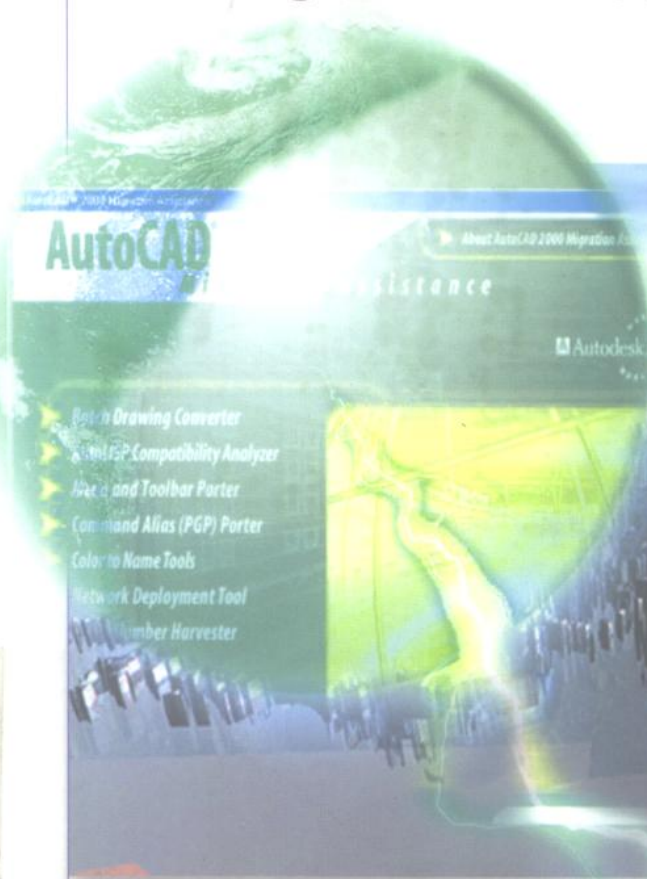
AutoCAD

2000

开发工具

— ObjectARX 开发工具与应用实例

江思敏 曹默 胡春江 主编



人民邮电出版社

AutoCAD2000 开发工具—— ObjectARX 开发工具与应用实例

江思敏 曹默 胡春江 主编

人民邮电出版社

内 容 简 介

本书主要针对 AutoCAD 2000 的二次开发, 系统讲述了如何使用 AutoCAD 2000 的一个优秀的开发工具 ObjectARX 2000 来进行 CAD 软件的开发。全书系统阐述了如何使用 ObjectARX 2000 进行应用程序开发, 包括 ObjectARX 基础知识、ObjectARX 类库以及如何使用 MFC 开发应用程序。本书主要以结合实例的方式, 由浅入深地进行系统阐述, 内容丰富、全面, 是 ObjectARX 2000 开发工具使用者及使用 AutoCAD 2000 软件的开发人员参考学习的好工具。

本书适用于使用 AutoCAD 进行产品设计, 并使用 ObjectARX 进行应用程序二次开发的工程技术人员和软件开发人员。本书还可供高等院校和培训学校的师生参考。

JS/12/12

AutoCAD 2000 开发工具—— ObjectARX 开发工具与应用实例

- ◆ 主 编 江思敏 曹 默 胡春江
责任编辑 须春美
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
北京朝阳隆昌印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787 × 1092 1/16
印张: 27.25
字数: 674 千字
印数: 1-5 000 册

1999 年 11 月第 1 版

1999 年 11 月北京第 1 次印刷

ISBN 7-115-08242-1/TP·1409

定价: 56.00 元

编者的话

计算机辅助设计 (Computer-Aided Design, 即 CAD) 技术的发展非常迅速, 在机械、电子、建筑、土木、广告等领域中, CAD 技术应用非常广泛。伴随着 CAD 技术的发展, 各种商品化的 CAD 软件也应运而生并迅速发展, AutoCAD 软件就是其中的优秀软件之一。Autodesk 公司自从开发了 AutoCAD 系列软件以来, 已经发展到 AutoCAD R13、R14, 取得了巨大的成功, 现在已经发展到 AutoCAD 2000。AutoCAD 系列 CAD 软件经过长期的不断的完善, 现在已经是全世界广大从事 CAD 技术的科技人员非常喜欢使用的软件之一。

AutoCAD 2000 是 Autodesk 于 1999 年推出的最新版本的软件, AutoCAD 软件集图形处理、产品设计、图形数据管理以及网络技术于一体, 成为连接世界的产品设计平台, 在 CAD 发展领域具有深远的意义。AutoCAD 2000 不但可以用来进行产品设计, 而且还可利用二次开发工具, 实现产品的参数化设计, 并可以针对用户的需求实现产品的二次软件开发, 大大提高产品设计的效率。AutoCAD 2000 提供了多种开发工具: AutoLISP 及 Visual LISP、ObjectARX 2000 和 VBA 等, 这些开发工具都可以基于 AutoCAD 2000 平台进行二次开发。

本书主要讲述如何使用 AutoCAD 2000 的非常优秀的二次开发工具 ObjectARX 2000 进行应用程序开发。全书主要分两部分, 第一部分从第 1 章到第 7 章, 主要讲述 ObjectARX 2000 基础知识, 包括 ObjectARX 2000 常用类、图形数据库技术、实体处理以及相关的基础知识。第二部分从第 8 章到第 19 章, 主要讲述如何使用 MFC 开发 ObjectARX 应用程序, 其中包括全局函数、选择集、实体和符号表函数、有关类的派生、协议扩展、事务管理、代理对象、多文档管理以及 MFC 类库等知识。本书以结合实例的方式, 由浅入深地进行系统阐述, 内容丰富、全面, 是 ObjectARX 2000 开发工具使用者及使用 AutoCAD 2000 软件的开发人员参考学习的好工具。

阅读本书的读者, 应该具有使用 C++ 语言开发程序的能力或了解这些知识, 并会使用 Visual C++ 软件开发工具, 因为本书讲述的程序内容是使用 Visual C++ 进行编写的。

本书由江思敏、曹默和胡春江主编, 李枫、何红兵和朱海涛等同志也参与编写了部分章节。由于水平有限, 时间仓促, 书中的缺点和不足在所难免, 敬请广大读者批评指正。

编者

1999 年 8 月

目 录

第 1 章 概要	1
1.1 ObjectARX 编程环境	1
1.1.1 访问 AutoCAD 数据库	1
1.1.2 与 AutoCAD 编辑器交互作用	1
1.1.3 使用 MFC 创建用户界面	2
1.1.4 支持多文档界面 (MDI)	2
1.1.5 创建自定义类	2
1.1.6 建立复杂的应用程序	2
1.1.7 与其它编程环境交互作用	2
1.2 ObjectARX 类库	2
1.2.1 AcRx 库	3
1.2.2 AcEd 库	4
1.2.3 AcDb 库	4
1.2.4 AcGi 库	5
1.2.5 AcGe 库	5
1.3 初步知识	7
1.3.1 系统要求	7
1.3.2 安装 ObjectARX	7
第 2 章 数据库入门	8
2.1 AutoCAD 数据库概述	8
2.1.1 多个数据库	9
2.1.2 Obtaining Object Ids	9
2.2 基本的数据库对象	9
2.3 在 AutoCAD 中创建对象	10
2.4 创建 ObjectARX 对象	12
2.4.1 创建实体	12
2.4.2 创建一个新层	13
2.4.3 打开与关闭 ObjectARX 对象	13
2.4.4 向组字典添加一个组	14
第 3 章 ObjectARX 应用程序基础	16
3.1 创建一个 ObjectARX 应用程序	16
3.1.1 创建自定义类	16
3.1.2 响应 AutoCAD 消息	16

3.1.3	ObjectARX 应用程序的事件顺序	20
3.1.4	为 AutoCAD 执行一个入口点	20
3.1.5	初始化一个 ObjectARX 应用程序	21
3.1.6	准备卸载	22
3.2	实例应用程序	22
3.3	注册新命令	23
3.3.1	命令堆栈	23
3.3.2	查找顺序	25
3.3.3	全局与本地化命令名比较	25
3.3.4	透明命令和模态命令	25
3.4	装载一个 ObjectARX 应用程序	25
3.4.1	库搜索路径	26
3.4.2	列出已装载的 ObjectARX 应用程序	26
3.5	卸载一个 ObjectARX 应用程序	26
3.6	请求装载	27
3.6.1	AutoCAD、Windows 系统注册和 ObjectARX 应用程序	27
3.6.2	ARX 应用程序安装时的注册修正	28
3.6.3	DEMANDLOAD 系统变量	30
3.6.4	关于客户对象检测的请求装载	30
3.6.5	关于命令的请求装载	31
3.6.6	关于 AutoCAD 启动的请求装载	32
3.6.7	使用系统注册管理应用程序	32
3.7	ARX 命令	33
3.8	从 AutoLISP 运行 ObjectARX 应用程序	34
3.9	错误处理	34
第 4 章	数据库操作	39
4.1	初始化数据库	39
4.2	创建和删除一个数据库	39
4.3	保存一个数据库	40
4.3.1	设定默认文件格式	40
4.3.2	全局保存函数	41
4.4	wblock 操作	41
4.4.1	从一个存在的数据库生成一个新的数据库	42
4.4.2	使用实体生成一个新数据库	42
4.5	插入一个数据库	42
4.6	设定当前数据库的值	43
4.6.1	数据库颜色值	43
4.6.2	数据库线型值	43

4.6.3 数据库线型比例值	43
4.6.4 数据库层值	44
4.7 数据库操作实例	44
4.8 长事务处理	46
4.8.1 类和函数概要	46
4.8.2 长事务处理实例	48
4.9 外部引用	51
4.9.1 外部引用的前后处理	52
4.9.2 锁住文件及一致性检查	53
4.10 索引和过滤器	53
4.11 图形摘要信息	54
4.12 最近被 Autodesk 软件所保存	55
第 5 章 数据库对象	56
5.1 打开和关闭数据库对象	56
5.2 删除对象	58
5.3 对象的数据库所有权	59
5.4 添加特定对象的数据	59
5.4.1 扩展数据	59
5.4.2 扩展字典	62
5.5 擦除对象	68
5.6 对象归档	69
第 6 章 实体对象	70
6.1 实体的基本知识	70
6.1.1 实体说明	70
6.1.2 实体所有关系	71
6.1.3 AutoCAD R12 实体	71
6.2 实体的公共特性	72
6.2.1 实体颜色	72
6.2.2 线型	73
6.2.3 实体线型比例	73
6.2.4 实体的可见性	74
6.2.5 实体图层	74
6.3 实体公共函数	75
6.3.1 对象捕捉点	75
6.3.2 几何变换函数	76
6.3.3 交点	77
6.3.4 GS 标记和子实体	78
6.3.5 实体炸开	94

6.4	创建 AutoCAD 实体实例	95
6.4.1	创建一个简单实体	95
6.4.2	创建一个简单的块表记录	96
6.4.3	创建一个具有属性定义的块表记录	97
6.4.4	创建一个具有属性的块引用	99
6.4.5	浏览一个块表记录	103
6.5	创建 AutoCAD 实体实例	105
6.5.1	创建一个复杂实体	105
6.5.2	浏览一条多义线的顶点	106
6.6	坐标系统访问	107
6.6.1	实体坐标系统	107
6.6.2	AcDb2dPolylineVertex 类	108
6.7	曲线函数	108
6.8	与实体实现超链接	109
6.8.1	AcDbHyperlink 类	109
6.8.2	AcDbHyperlinkCollection 类	110
6.8.3	AcDbEntityHyperlinkPE 类	110
6.8.4	超链接实例	110
第 7 章	容器对象	113
7.1	符号表和字典的比较	113
7.2	符号表	115
7.2.1	块表	116
7.2.2	层表	116
7.2.3	迭代器	119
7.3	字典	120
7.3.1	组和组字典	120
7.3.2	复合线样式字典	123
7.3.3	布局字典	123
7.3.4	创建字典	124
7.3.5	列举字典条目	125
7.4	布局	126
7.5	扩展记录	127
7.5.1	扩展记录的 DXF 组码	128
7.5.2	示例	128
第 8 章	全局函数	131
8.1	ObjectARX 中定义的变量、类型和值	131
8.1.1	一般类型及定义	131
8.1.2	有用的值	132

8.1.3	ObjectARX 函数返回码	136
8.1.4	用户输入控制码	137
8.2	普通特征的 ObjectARX 库函数	137
8.2.1	ObjectARX 全局函数调用与 AutoLISP 调用比较	137
8.2.2	函数返回值	138
8.2.3	外部函数	139
8.2.4	出错处理	140
8.2.5	应用程序间通讯	140
8.2.6	外部应用程序处理	144
8.3	表与其它动态分配的数据	145
8.4	结果缓冲区内存管理	147
8.4.1	链表的建立和删除	148
8.4.2	AutoLISP 表	149
8.4.3	带 DXF 码的实体链表	151
8.4.4	命令和函数调用表	151
8.5	命令和查询函数	152
8.5.1	acedCommand()和 acedCmd()函数	152
8.5.2	系统变量函数	154
8.5.3	AutoLISP 符号函数	155
8.5.4	文件搜索函数	156
8.5.5	对象捕捉函数	157
8.5.6	视口描述符函数	157
8.5.7	几何功能函数	158
8.5.8	文本框功能函数	159
8.6	获取用户输入函数	161
8.6.1	用户输入函数	161
8.6.2	控制用户输入函数条件	162
8.6.3	选择集图形拖动函数	164
8.6.4	用户中断	164
8.6.5	返回值给 AutoLISP 函数	165
8.7	变换函数	166
8.7.1	字符串转换函数	166
8.7.2	单位转换函数	167
8.7.3	字符类型处理函数	168
8.8	坐标系变换	168
8.9	显示控制函数	169
8.9.1	交互输出 Interactive Output	169
8.9.2	图形和文本屏幕控制	170

8.9.3 低层图形和用户输入控制	170
8.10 字符匹配	171
第9章 选择集、实体和符号表函数	172
9.1 选择集名和实体名	172
9.2 选择集的操作	172
9.2.1 选择集的过滤器列表	175
9.2.2 选择集操作	180
9.2.3 选择集变换	182
9.3 实体名和实体数据函数	185
9.3.1 实体名函数	185
9.3.2 实体数据函数	193
9.3.3 实体数据函数和图形屏幕	202
9.3.4 扩展数据的标记	203
9.4 访问符号表	209
第10章 派生自定义类	212
10.1 派生自定义类	212
10.2 运行时类识别	213
10.3 类声明宏	214
10.4 类执行宏	215
10.5 类初始化函数	216
第11章 派生 AcDbObject 类	217
11.1 对象的引用	217
11.2 所有关系引用	218
11.2.1 所有关系的使用	219
11.2.2 所有关系类型	219
11.2.3 建立所有关系体系	219
11.3 指针引用	229
11.3.1 紧密关联指针	229
11.3.2 松散关联指针	229
11.4 自定义类的长期处理问题	230
11.5 删除对象	232
11.6 撤消和重复操作	232
11.6.1 自动撤销操作	232
11.6.2 部分撤销操作	233
11.6.3 重复操作	235
11.7 subErase, subOpen, subClose 和 subCancel 函数	235
11.8 编程实例	248
11.8.1 头文件	248

11.8.2 源文件	249
11.9 对象版本支持	255
11.9.1 类版本	255
11.9.2 类的重命名	258
11.9.3 类数据或扩展数据的版本号	258
第 12 章 派生 AcDbEntity 类	260
12.1 派生自定义实体	260
12.1.1 必须重载的 AcDbEntity 函数	260
12.1.2 通常需要重载的 AcDbEntity 函数	260
12.1.3 很少重载的 AcDbEntity 函数	262
12.2 重载实体的公共函数	262
12.2.1 重载 worldDraw()和 viewportDraw()函数	262
12.2.2 重载 saveAs()函数	263
12.2.3 执行对象捕捉点函数	264
12.2.4 执行夹点函数	266
12.2.5 拉伸点函数	269
12.2.6 变换函数	270
12.2.7 与其它实体相交	271
12.2.8 与其它自定义实体相交	277
12.2.9 分解实体	277
12.3 扩展实体功能	278
12.4 使用 AcEdJig 类	278
12.4.1 由 AcEdJig 派生新类	278
12.4.2 使用 AcEdJig 类的一般步骤	278
12.4.3 设置拖动关联提示	279
12.4.4 拖动循环	279
12.4.5 sampler()、update()和 entity()函数的实现	280
12.4.6 添加实体到数据库	282
12.4.7 实例	282
第 13 章 代理对象	289
13.1 定义代理对象	289
13.2 代理对象生命周期	289
13.3 处理代理对象	290
13.4 显示代理实体	290
13.5 编辑代理实体	290
13.6 卸载应用程序	291
第 14 章 通知处理	292
14.1 通知概述	292

14.1.1	响应类	292
14.1.2	响应对象类型	293
14.2	使用响应	293
14.2.1	AcDbObject 和数据库通知事件	294
14.2.2	自定义通知	294
14.2.3	使用编辑响应	295
14.2.4	使用数据库响应	295
14.2.5	使用对象响应	299
14.3	通知使用原则	310
第 15 章	事务管理	312
15.1	事务管理概述	312
15.2	事务管理器	312
15.3	事务嵌套	313
15.4	事务边界	313
15.5	在事务中获取对象指针	314
15.6	事务与新建对象	314
15.7	提交时间原则	315
15.8	事务与取消	315
15.9	事务模型与打开和关闭机制的混合使用	315
15.10	事务与图形生成	315
15.11	事务响应	316
15.12	事务管理实例	316
第 16 章	协议扩展	333
16.1	定义协议扩展	333
16.2	实现协议扩展	333
16.2.1	声明并定义协议扩展类	333
16.2.2	注册协议扩展类	334
16.2.3	协议扩展缺省类	335
16.2.4	卸载应用程序	335
16.2.5	使用协议扩展函数	335
16.3	MATCH 命令的协议扩展	336
16.4	协议扩展实例	336
第 17 章	输入点处理	342
17.1	自定义对象捕捉模式	342
17.1.1	创建并注册一个自定义对象捕捉模式	342
17.1.2	创建协议扩展类	343
17.1.3	创建自定义符号	344
17.1.4	实例	344

17.2 输入点管理	353
17.2.1 输入点管理器	353
17.2.2 输入上下文事件	354
17.2.3 输入点过滤器和监控器	360
第 18 章 使用 MFC 类库	374
18.1 在 ObjectARX 应用程序中使用 MFC	374
18.2 在 ObjectARX 应用程序中使用动态链接 MFC 库	374
18.2.1 VC++ 工程中动态链接 MFC 库的设置	374
18.2.2 调试应用程序	375
18.2.3 资源管理	375
18.3 建立 MFC 用户界面支持	376
18.3.1 类层次结构	377
18.3.2 AdUi 消息	377
18.3.3 AdUi 提示窗口	377
18.3.4 AdUi 对话框类	378
18.3.5 AcUi Dialog 类	379
18.3.6 可扩充标签类	379
18.3.7 AdUi 和 AcUi 编辑控件	379
18.3.8 AdUi 和 AcUi 组合框控件	380
18.3.9 AcUi MRU 组合框	381
18.3.10 AdUi 按钮类	381
18.3.11 AcUi 按钮类	381
18.3.12 对话框数据驻留	382
18.3.13 使用并扩充 AdUi 标签对话框系统	382
18.3.14 创建可扩充的自定义标签对话框	382
18.3.15 扩充 AutoCAD 内部标签对话框	383
18.4 在 VC++ AppWizard 中使用 AdUi 和 AcUi	385
18.4.1 创建 ObjectARX MFC 应用程序框架	385
18.4.2 使用 App Studio 创建 MFC 对话框	387
18.4.3 创建类和控件	387
18.4.4 创建对话框处理	388
18.4.5 为处理函数增加代码	389
第 19 章 多文档界面	398
19.1 简介	398
19.1.1 文档执行上下文	398
19.1.2 数据实例	398
19.1.3 文档锁定	398
19.1.4 文档处理类	399

19.2	术语	399
19.3	SDI 系统变量	401
19.4	兼容级别	402
19.4.1	SDI-Only 级别	402
19.4.2	MDI-Aware 级别	402
19.4.3	MDI-Capable 级别	404
19.4.4	MDI-Enhanced 级别	405
19.5	与多文档交互作用	405
19.5.1	访问当前文档及其相关对象	405
19.5.2	访问与非重入文档关联的数据库	405
19.5.3	设置未激活的当前文档	406
19.6	文档事件通知	406
19.7	应用程序特殊文档对象	406
19.8	非重入命令	407
19.8.1	生成非重入命令	407
19.8.2	非重入的 AutoCAD 命令	407
19.9	多文档命令	408
19.10	关闭文档切换	409
19.11	应用程序执行上下文	409
19.11.1	应用程序执行上下文中的代码调用	409
19.11.2	代码区别	410
19.11.3	其它应用程序执行上下文	410
19.12	数据库的 Undo 和事务管理功能	411
19.13	独立的文档数据库	411
19.14	一个 MDI-Aware 型例程	412

第 1 章 概要

一个 ObjectARX 应用程序是一个分享 AutoCAD 的地址空间并为 AutoCAD 直接调用的动态链接库(DLL)。用户可以向 ObjectARX 编程环境添加新类，并可以导出它们为其它程序所用。用户创建的 ObjectARX 实体实际上与 AutoCAD 内置的实体是没有区别的。用户可以在运行时间向已存在的 AutoCAD 类添加函数以扩展 ObjectARX 协议。

本章将概述组成 ObjectARX 的类库，并讲述如何开始使用 ObjectARX。学习本章的读者，应该熟悉 C++ 编程语言、面向对象的编程方法以及 AutoCAD。

1.1 ObjectARX 编程环境

ObjectARX 编程环境提供了一个面向对象的 C++ 应用程序开发界面，使开发者能够使用、定做和扩展 AutoCAD。ObjectARX 库包括各种各样的工具，可以使应用程序开发者方便地利用 AutoCAD 的开放结构，这些工具可以方便应用程序对 AutoCAD 数据库结构、图形系统和本地命令直接访问。另外，这些库被设计得可以与 Visual LISP 及其它应用程序开发界面协同工作，以便开发者可以根据自己的需要和经验来选择最适合的开发工具。

作为一个开发者，用户可以使用 ObjectARX 完成以下任务：

- 访问 AutoCAD 数据库；
- 与 AutoCAD 编辑器交互作用；
- 使用 MFC 创建用户界面；
- 支持多文档界面 (MDI)；
- 创建自定义类；
- 建立复杂的应用程序；
- 与其它编程环境交互作用。

下面将简短地叙述这些任务主题，在本书的其它章节，将会对它们进行更加详细的描述。

1.1.1 访问 AutoCAD 数据库

一个 AutoCAD 图形是保存在数据库中对象的集合。这些对象不但表示图形实体，而且也表示了如符号表和字典等的内在结构。ObjectARX 提供给用户应用程序访问这些数据库结构的方法，另外，用户可以为指定的应用程序创建数据库。

1.1.2 与 AutoCAD 编辑器交互作用

ObjectARX 提供了可以与 AutoCAD 编辑器交互作用的类和成员函数。用户可以注册 AutoCAD 命令，这些命令将与内置命令一样。用户的应用程序可以接收和响应发生在 AutoCAD 中的各种事件的通知。

1.1.3 使用 MFC 创建用户界面

ObjectARX应用程序可以使用与AutoCAD共享的动态链接MFC库来创建。用户可以使用这些库来创建标准Microsoft窗口图形用户界面（GUIs）。

1.1.4 支持多文档界面（MDI）

使用ObjectARX，用户可以创建支持AutoCAD多文档界面的应用程序，并且用户可以确保用户的应用程序将会正确地与Microsoft窗口环境的其它应用程序交互作用。

1.1.5 创建自定义类

用户可以在ObjectARX层次下以ObjectARX 类为基础创建自己的自定义类。另外，当创建自定义类时，用户也可以利用ObjectARX 扩展的图形库。

1.1.6 建立复杂的应用程序

ObjectARX支持复杂应用程序的开发，它提供了如下特征：

- 通知；
- 事务处理；
- 深度克隆；
- 引用编辑；
- 协议扩展；
- 协议对象支持。

1.1.7 与其它编程环境交互作用

ObjectARX应用程序可与其它程序（如Visual LISP、ActiveX和COM）开发界面通信。另外，ObjectARX应用程序可以通过实体与URLs关联、与Internet链接，并可以从网络（WWW）上装载和保存图形文件。

1.2 ObjectARX 类库

ObjectARX 环境包含如下几组的库和函数：

- AcRx：用于约束一个应用程序以及实时类注册和识别的类；
- AcEd：用于注册本地命令和系统事件通知的类；
- AcDb：用于组成 AutoCAD 数据库的类；
- AcGi：用于渲染 AutoCAD 实体的图形界面；
- AcGe：用于一般线形代数和几何实体的通用库。

表 1.1 列出了链接 ObjectARX 应用程序所需要的库。所有的 ObjectARX 应用程序必须与 acad.lib 和 rxapi.lib 库链接。对于其它的库也是要求的，但这要依赖于用户正在使用的 ObjectARX 库和函数的前缀。

表 1.1 链接 ObjectARX 应用程序所需要的库

前 缀	需要的库
AcRx	acad.lib、rxapi.lib、acr15.lib
AcEd	acad.lib、rxapi.lib、acedapi.lib、acr15.lib
AcDb	acad.lib、rxapi.lib、acdb15.lib、acr15.lib
AcGi	acad.lib、rxapi.lib、acgiapi.lib、acr15.lib
AcGe	acad.lib、rxapi.lib、acge15.lib、acr15.lib

1.2.1 AcRx 库

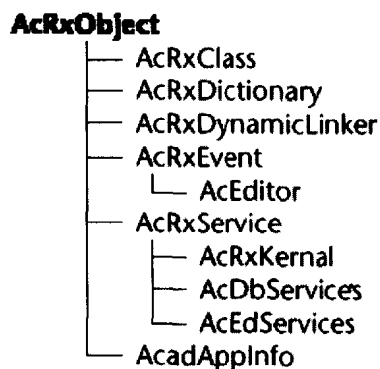
AcRx 库为动态链接库(DLL)初始化及链接提供了系统级的类，这些类也可以用于实时类注册和识别。这个库的基本类是 AcRxObject，它可以为下述操作提供工具：

- 对象实时类识别和继承分析；
- 向一个已存在的类实时添加新的协议（参考第 16 章）；
- 实体相等与比较测试；
- 实体拷贝。

AcRx 库也提供了一套 C++ 宏，以帮助用户生成由 AcRxObject 派生的新的 ObjectARX 类。

AcRxDictionary 是该库中另一个重要的类。一个字典是一个从文本字符向另一个实体的映射。AcRx 库将它们的实体、类和服务字典放在一个全局实体字典中，这是 AcRxDictionary 类的一个应用实例。应用程序可以增加实体到这个字典中，以便它们可以为其它应用程序所应用。

AcRx 库中的类层次结构如下：



每个 AcRxObject 的子类都有一个相关的类描述对象（AcRxClass 类型），用于运行时间类型识别。ObjectARX 提供了测试一个对象是否是一个特定的类或派生类的函数，也提供了确定两个对象是否是相同的类的函数，还提供了为给定的类返回类描述的函数。