

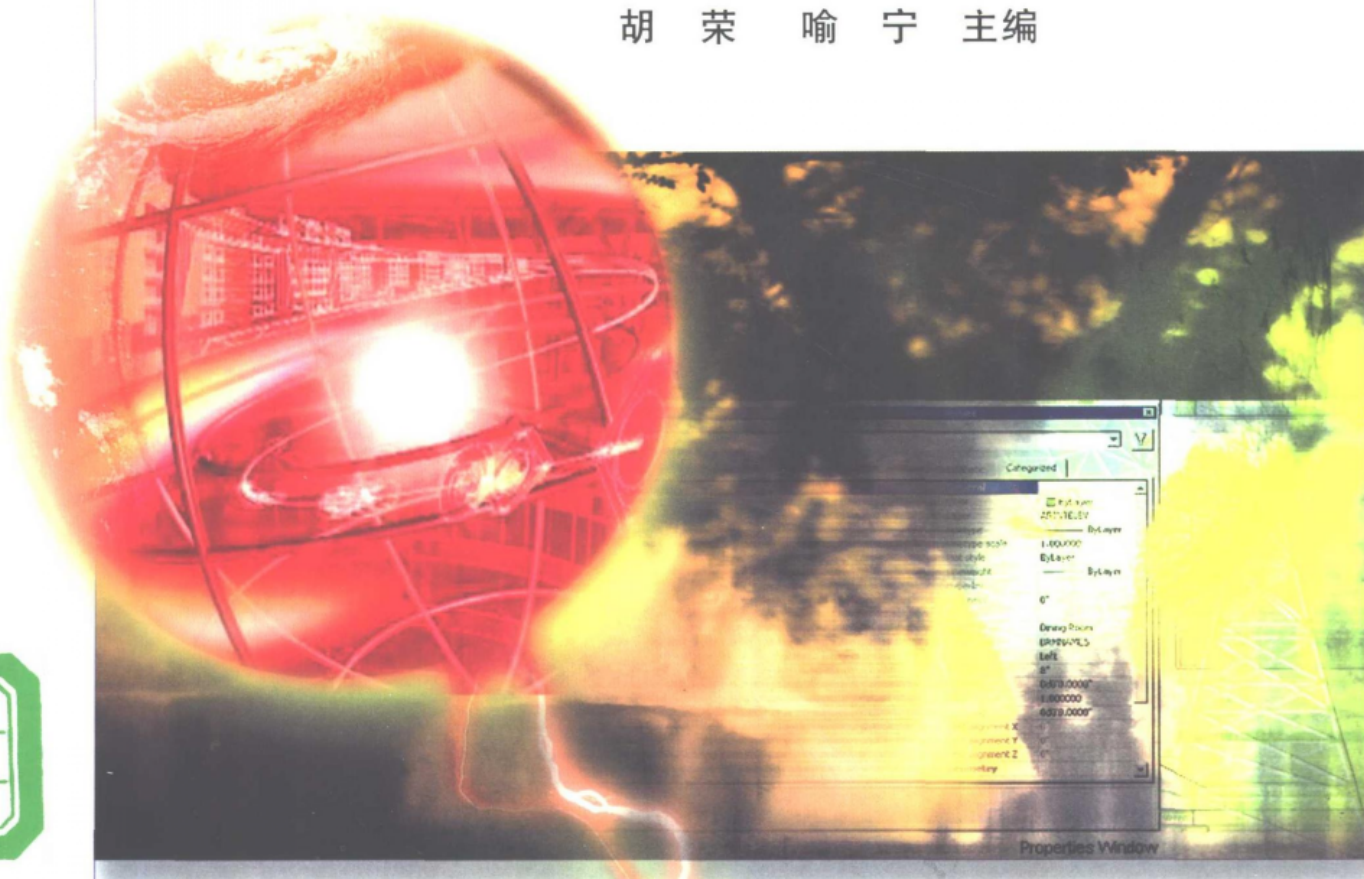
Auto CAD 2000

Auto CAD 2000

开发工具

— VBA 及 ActiveX 开发指南

胡荣 喻宁 主编



人民邮电出版社

AutoCAD 2000 开发工具—— VBA 及 ActiveX 开发指南

胡 荣 喻 宁 主 编

人 民 邮 电 出 版 社

内 容 简 介

本书以 AutoCAD 2000 为基本的开发环境,通过较多的编程实例,详细地介绍了 Autodesk 公司推出的功能强大的开发工具 VBA。

全书分为 14 章,分别介绍了 VBA 的基础应用,控制 AutoCAD 的开发环境,使用 AutoCAD 绘图,对 AutoCAD 的各种定制、高级绘图与组织等,最后还介绍了一个实例——设计花园小径。

本书实例丰富、层次清晰,适用于机械、电子、建筑、广告设计等领域的工程技术人员和设计人员,对想提高自己的设计水平的技术人员、AutoCAD 的设计人员具有很好的参考价值。本书还可供高等院校和培训学校的师生参考使用。

Auto CAD 2000 开发工具——VBA 及 ActiveX 开发指南

◆ 主 编 胡 荣 喻 宁

责任编辑 黄汉兵

◆ 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:17.75

字数:437 千字

1999 年 10 月第 1 版

印数:5 001—8 000 册

2000 年 2 月北京第 2 次印刷

ISBN 7-115-08186-7/TP·1370

定价:27.00 元

前 言

CAD (Computer Aided Design) 即计算机辅助设计, 是运用计算机系统辅助一项设计的建立、修改、分析和优化的过程。随着工程设计 CAD 技术的不断发展, 其覆盖的工作领域也在不断扩大, 包括工程设计的项目管理、初步设计、分析计算、工程图绘制、系统优化等内容。当今计算机技术的迅猛发展给各行各业都带来了一系列深刻的技术变革。在机械、电子、建筑、广告等领域中, 应用 CAD 技术非常广泛。AutoCAD 是 Autodesk 公司开发的绘图软件包, 最早用于 DOS 系统下, 较新的版本可以运行于多种平台。由于 AutoCAD 多年来成功的生产和营销, 使其成为计算机环境下最为流行和普遍采用的交互图形软件, 它具有完备的系统开放性和丰富的个性化能力。到目前为止, Autodesk 公司提供了 4 种二次开发手段: AutoLISP、ADS、VBA 和 ObjectARX。Microsoft VBA 是一个面向对象的编程环境, 它和 Visual Basic (VB) 一样有很强的开发能力。使用 VBA 开发有以下一些特点:

- 定制简单, 数据共享方便;
- 使用 VBA 可进行可视定制;
- 使用 VBA 编码简单;
- 方便 AutoCAD 用户进行操作;
- 有利于 CAD 管理者管理。

本书所介绍的主要内容是怎样使用 VBA 对 AutoCAD 进行二次开发。全书共分为 14 章, 从各方面说明了使用 VBA 进行开发的过程与方法。本书包含了丰富的实例, 实例主要以机械零件的参数化设计为主, 简单易懂。通过本书的学习, 读者将能熟练地使用 VBA 进行 AutoCAD 的二次开发。

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

本书由胡荣和喻宁主编, 张伟、马志斌和杜小民等同志也参与了部分章节的编写。江思敏同志在百忙之余阅读了本书, 并提出了许多宝贵的意见, 在此深表感谢。由于作者水平有限, 时间仓促, 书中缺点和不足在所难免, 敬请广大读者批评指正。

编者

1999/8/25

目 录

前 言

第0章 绪 论	1
0.1 AutoCAD ActiveX 技术概述	1
0.2 VBA 接口概述	2
0.3 AutoCAD ActiveX 和 VBA	3
0.4 本书组成结构	4
0.5 本书的约定	4
0.6 本书的样本代码	4
第1章 VBA 初步	7
1.1 理解嵌入和全局的 VBA 工程	7
1.2 用 VBA 管理器组织工程	8
1.2.1 加载已经存在的工程	8
1.2.2 卸载工程	9
1.2.3 把一个工程嵌入图形文件	9
1.2.4 从图形文件中提取工程	9
1.2.5 创建一个新工程文件	9
1.2.6 保存嵌入的工程	10
1.3 处理宏	10
1.3.1 运行宏	11
1.3.2 编辑宏	11
1.3.3 创建宏	11
1.3.4 删除宏	12
1.3.5 设置工程选项	12
1.3.6 转入并运行宏	12
1.4 用 VBA IDE 编辑工程	13
1.4.1 打开 VBA IDE	13
1.4.2 观察工程信息	13
1.4.3 定义工程中的元件	14
1.4.4 引入已有元件	14
1.4.5 编辑元件	15
1.4.6 运行宏	16
1.4.7 给工程命名	16
1.4.8 保存工程	17

1.4.9 其它 VBA 工程的参考	17
1.4.10 设置 VBA IDE 选项	18
1.5 练习	19
1.6 其它	20
1.7 AutoCAD VBA 工程术语	20
1.8 AutoCAD VBA 命令	21
第 2 章 ActiveX 自动化基础	22
2.1 理解 AutoCAD 对象模型	22
2.1.1 简单介绍程序对象	22
2.1.2 简单介绍文档对象	23
2.1.3 简单介绍集合对象	24
2.1.4 简单介绍图形对象和非图形对象	24
2.1.5 简单介绍参数选择、绘图和 utility 对象	24
2.2 访问对象层	25
2.3 使用集合对象	26
2.3.1 访问集合	27
2.3.2 给集合对象加新成员	27
2.3.3 通过集合对象重新声明	27
2.3.4 删除集合对象中的成员	28
2.4 属性和方法	28
2.5 父对象	29
2.6 定位类型库	29
2.7 得到数据库中的第一个实体	29
2.8 在方法和属性中使用变量	30
2.8.1 什么是变量	30
2.8.2 使用数组变量	30
2.8.3 数组转换为变量	30
2.8.4 解释变量数组	31
2.9 使用其它的编程语言	32
第 3 章 设置 AutoCAD 环境	36
3.1 打开、保存、关闭图形文件	36
3.2 设置 AutoCAD 参数选择	38
3.3 设置程序窗口	39
3.3.1 改变程序窗口的大小和位置	39
3.3.2 最大化和最小化 AutoCAD 窗口	39
3.3.3 查询 AutoCAD 窗口的当前状态	39
3.3.4 设置程序窗口为不可见	40
3.4 设置绘图窗口	40

3.4.1	修改文档窗口的位置和大小	40
3.4.2	最小化、最大化文档窗口	40
3.4.3	查询文档窗口的当前状态	40
3.4.4	使用缩放	41
3.4.5	使用命名视图	44
3.4.6	使用平铺视点	45
3.4.7	修正文档窗口中的几何学	48
3.5	重置活动对象	49
3.6	设置并返回系统变量	49
3.7	精确绘图	50
3.8	提示用户输入	57
3.8.1	GetString 方法	57
3.8.2	GetPoint 方法	58
3.8.3	GetKeyword 方法	58
3.8.4	控制用户输入	59
3.9	访问 AutoCAD 命令行	60
3.10	无文档工作	60
3.11	引入其它文件格式	61
3.12	输出文件格式	61
第 4 章	创建和编辑 AutoCAD 实体	63
4.1	创建对象	63
4.1.1	创建容器对象	63
4.1.2	创建直线	64
4.1.4	创建点对象	65
4.1.7	创建填充图	70
4.2	编辑对象	73
4.2.1	使用有名对象	73
4.2.2	选择对象	74
4.2.3	复制对象	76
4.2.4	移动对象	83
4.2.5	删除对象	85
4.2.6	缩放对象	85
4.2.7	变换对象	86
4.2.8	延伸和剪切对象	89
4.2.9	分解对象	89
4.2.10	编辑多义线	91
4.2.11	编辑样条曲线	92
4.2.12	编辑填充图	95

4.3 使用图层、颜色和线型	98
4.3.1 使用图层	98
4.3.2 使用颜色	103
4.3.3 使用线型	103
4.3.4 给实体设置图层、颜色、线型	106
4.4 给图形添加文本	108
4.4.1 文本类型	108
4.4.2 单行文本 (Text)	114
4.4.3 多行文本 (Mtext)	118
4.4.4 Unicode 字符、控制代码和特殊字符	122
4.4.5 替换字体	123
4.4.6 检查拼写	124
第 5 章 尺寸和公差	125
5.1 尺寸标注概述	125
5.1.1 查看尺寸标注的各组成部分	126
5.1.2 定义尺寸标注系统变量	127
5.1.3 设置尺寸标注文本类型	127
5.1.4 理解引出线	127
5.1.5 理解关联尺寸	127
5.2 建立尺寸标注	128
5.2.1 建立长度标注	128
5.2.2 建立半径标注	129
5.2.3 建立角度标注	130
5.2.4 建立坐标标注	131
5.3 编辑尺寸标注	132
5.4 以尺寸类型方式工作	133
5.5 在模型空间和图纸空间进行尺寸标注	137
5.6 建立旁注线和注释	137
5.6.1 建立旁注线	137
5.6.2 给旁注线添加注释	138
5.6.3 旁注线的关联性	139
5.6.4 编辑旁注线关联性	140
5.6.5 编辑旁注线	140
5.7 建立几何公差	140
第 6 章 定制工具条和菜单	142
6.1 菜单条和菜单条集	142
6.2 装载菜单组	144
6.3 修改菜单条	145

6.4	创建和编辑下拉菜单和快捷菜单	146
6.4.1	创建新菜单	147
6.4.2	向菜单加入菜单项	147
6.4.3	添加分隔符	149
6.4.4	给菜单项设一个加速键	149
6.4.5	创建层叠子菜单	150
6.4.6	从菜单中删除菜单项	151
6.4.7	菜单项属性	152
6.5	创建和编辑工具条	154
6.5.1	创建新工具条	154
6.5.2	添加新的工具条按钮	154
6.5.3	添加分隔符	156
6.5.4	定义工具按钮图像	156
6.5.5	创建 Flyout 工具条	158
6.5.6	浮动定位工具条	159
6.5.7	删除工具条按钮	160
6.5.8	工具条属性	160
6.6	创建宏	161
6.6.1	宏字符映射的 ASCII 等同字符	161
6.6.2	终止宏	162
6.6.3	暂停用户输入	163
6.6.4	取消一个命令	164
6.6.5	宏循环	164
6.6.6	使用单对象选择模式	164
6.7	创建菜单项和工具条项的状态行帮助	164
6.8	添加右击键项目	165
第 7 章	事件	167
7.1	理解 AutoCAD 中的事件	167
7.2	事件处理程序编写	167
7.3	处理应用程序级事件	168
7.4	处理文档级事件	171
7.5	处理对象级事件	173
第 8 章	三维空间	176
8.1	三维坐标	176
8.2	定义用户坐标系统	178
8.3	坐标转换	180
8.4	创建三维对象	182
8.4.1	线框	183

8.4.2	网格	184
8.4.3	多面网格	186
8.4.4	实体对象	187
8.5	编辑 3D 对象	188
8.6	编辑 3D 实体	191
第 9 章	定义布局和打印	195
9.1	模型空间和图纸空间	195
9.2	布局	195
9.2.1	布局和块之间的关系	195
9.2.2	打印配置	196
9.2.3	布局设置	196
9.3	视点	197
9.3.1	切换到图纸空间布局	198
9.3.2	切换到模型空间布局	199
9.3.3	创建图纸空间视点	199
9.3.4	修改视点, 视图和内容	202
9.3.5	在图纸空间缩放样本线型	203
9.3.6	在打印的视点中隐藏直线	204
9.4	打印图形	204
9.4.1	Performing Basic Plotting	204
9.4.2	从模型空间打印	205
9.4.3	从图纸空间打印	205
第 10 章	高级绘图和组织技术	207
10.1	光栅图像	207
10.1.1	图形文件中的光栅图像	207
10.1.2	附加、比例确定光栅图像	209
10.1.3	管理光栅图像	210
10.1.4	修改图像和图像边界	210
10.1.5	剪贴图像	212
10.2	使用块和属性	214
10.2.1	使用块	214
10.2.2	使用属性	220
10.3	使用外部参照	226
10.3.1	更新 Xrefs	227
10.3.2	附贴 Xrefs	227
10.3.3	分离 Xrefs	228
10.3.4	重载 Xrefs	229
10.3.5	卸载 Xrefs	230

10.3.6 捆绑 Xrefs	231
10.3.7 剪贴 Blocks and Xrefs	233
10.4 赋值和接收外部数据	233
第 11 章 使用 VBA 开发应用程序	236
11.1 更多的 VBA 技术	236
11.2 在 VBA 中使用窗体	236
11.2.1 在设计模式下设计, 在运行模式下运行	237
11.2.2 在你的工程中创立一个新的窗体	237
11.2.3 向窗体中添加控件	237
11.2.4 显示和隐藏窗体	239
11.2.5 加载和卸载窗体	239
11.2.6 设计在有模式窗体情况下的应用程序	240
11.3 处理错误	240
11.3.1 捕获运行错误	241
11.3.2 对被捕获的错误作出响应	242
11.3.3 对 AutoCAD 用户输入错误作出响应	242
11.4 加密 VBA 代码模块	243
11.5 从工具条或菜单中运行 VBA 宏	243
11.6 自动载入一个 VBA 工程	243
11.7 自动运行 VBA 宏	243
11.8 载入工程后自动打开 VBA 的 IDE 环境	244
11.9 工作于空文档状态	244
11.10 发布你的应用程序	244
第 12 章 与其它应用程序、数据库及 Windows API 通信	245
12.1 与 Visual Lisp 程序通信	245
12.2 与其它 Windows 应用程序通信	245
12.2.1 引用其它程序的 ActiveX 对象库	246
12.2.2 建立其它应用程序的一个实例	246
12.2.3 对其它应用程序的对象进行编程	247
12.3 使用数据访问对象(DAO)访问数据库信息	249
12.3.1 引用 DAO 对象库	249
12.3.2 打开数据库	249
12.3.3 用 DAO 对象模式编程	250
12.4 在 VBA 中访问 Windows API	250
第 13 章 一个 ActiveX 和 VBA 的应用实例——设计花园小径	252
13.1 检查运行环境	252
13.2 定义目标	253
13.3 编写你的第一个函数	253

13.4	获取输入	254
13.4.1	声明变量	254
13.4.2	输入 gpuser 子程序	255
13.5	绘制小路轮廓	257
13.6	绘制瓷砖	258
13.7	把所有东西结合起来	261
13.8	单步执行代码	261
13.9	执行宏	262
13.10	加入对话框环境	263
13.10.1	建立对话框	263
13.10.2	使用项目窗口为你的工程导航	265
13.10.3	更新已有代码	266
13.10.4	给对话框添加代码	268

第0章 绪论

概述

本章介绍使用 ActivX 接口以及 Visual Basic 编程环境开发 AutoCAD 的概念。另外本章也将介绍 AutoCAD ActiveX 和 VBA 所带的所有文档和样本代码。

本章内容

- AutoCAD ActiveX 技术概述
- VBA (Visual Basic for Applications) 接口概述
- AutoCAD ActiveX 和 VBA
- 本书组成结构
- 本书的约定
- 本书的样本代码

0.1 AutoCAD ActiveX 技术概述

AutoCAD ActiveX 提供在 AutoCAD 内或在 AutoCAD 外控制编程的机制。它是通过使 AutoCAD 对象开放到外部世界来做到这一点的。只要这些对象被开放，就可以通过多种不同的编程语言环境和其它应用程序（比如 Microsoft Word VBA 或者 Excel VBA）来访问它们，如图 0.1 所示。

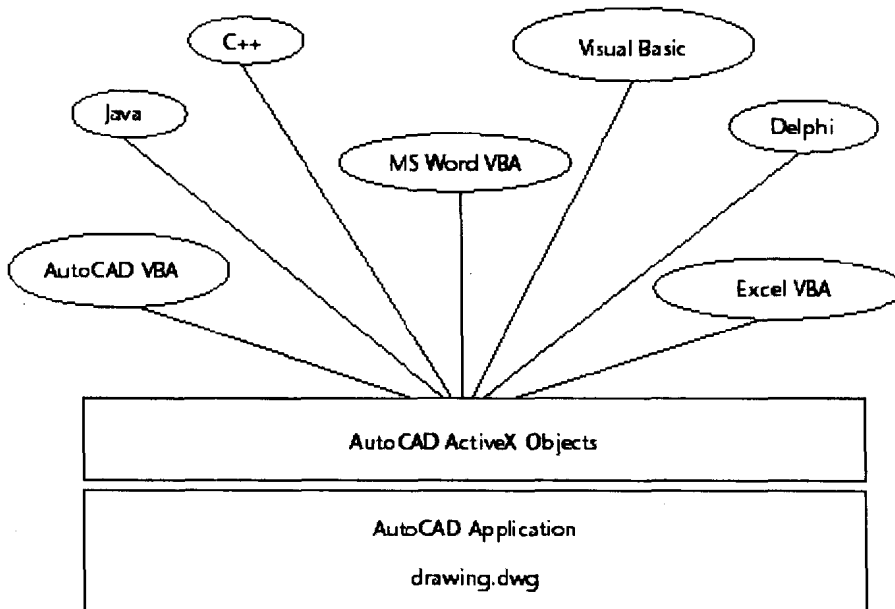


图 0.1 AutoCAD 的开发结构

实现 AutoCAD 的 ActiveX 界面有两个好处：

其一是对 AutoCAD 绘图的编程对更多的编程环境开放,而在 ActiveX 之前,这一类的开发一直都局限在 AutoLISP 和 C++的接口上。

其二是和其它的 Windows 应用程序(比如 Microsoft Excel 和 Word)共享数据变得意想不到的方便。

本节将介绍 AutoCAD ActiveX 对象概览:

对任何 ActiveX 应用程序来说,它的主体都是一个对象。每个公开的对象都代表 AutoCAD 很具体的一部分。AutoCAD ActiveX 接口中就有很多不同类型的对象。例如:

- 图形对象,比如直线、圆弧、文本和尺寸;
- 类型设置,比如线型和尺寸类型;
- 组织结构,比如层、组、块;
- 图形显示,比如视图和视口;
- 图形和 AutoCAD 应用程序也可认为是对象。

0.2 VBA 接口概述

Microsoft VBA 是一个面向对象的编程环境,它和 Visual Basic (VB)一样有很强的开发能力。VBA 和 VB 两者之间的区别是 VBA 和 AutoCAD 运行在同一处理空间,为 AutoCAD 提供智能和快速的编程环境。

VBA 也提供集成了其它可以使用 VBA 的应用程序的应用。也就是说,当 AutoCAD 使用其它应用程序对象库的时候,AutoCAD 可以看作一个其它的应用程序的自动控制器,比如 Microsoft Word 和 Excel 就是这样的。

Visual Basic 独立的开发版本是需要分开购买的,它用附加控件,如外部的数据库引擎和报告书写能力来增强 AutoCAD VBA。

在 AutoCAD 中使用 VBA 有以下四个好处:

- Visual Basic 编程环境易学易用;
- VBA 作为 AutoCAD 的一个过程运行,这使程序执行速度变得非常快;
- 对话框结构快速有效。允许开发者在设计时启动应用程序并且能很快得到反馈;
- 对象可以独立出来,也可以嵌入 AutoCAD 图形。这样为用户发布他们的开发程序提供更多的灵活性。

1. 怎样在 AutoCAD 中实现 VBA

VBA 通过 AutoCAD ActiveX Automation 接口向 AutoCAD 发送消息。AutoCAD VBA 允许 VBA 环境和 AutoCAD 同时运行,并且通过 AutoCAD ActiveX Automation 接口界面提供 AutoCAD 的编程控制。AutoCAD、ActiveX Automation 和 VBA 之间的结合提供很强的接口功能,不但可以控制 AutoCAD 对象,而且可以从其它应用程序发送数据或接收数据。

在 AutoCAD 中有三个用来定义 ActiveX 和 VBA 编程的基本要素。首先是 AutoCAD 本身,AutoCAD 有一套压缩 AutoCAD 实体,数据和命令的对象。因为 AutoCAD 是作为一个开放结构应用程序设计的,它具有有多级接口。为了有效地使用 VBA,要求开发者必须熟悉 AutoCAD 的可编程能力。如果开发者曾经使用过 AutoCAD 来控制 AutoCAD 编

程开发，那对 AutoCAD 设备理解就已经有了一个很好的基础。不过，我们也将发现基于对象的 VBA 和 Visual LISP 将会有很大的不同。

第二个要素是 AutoCAD ActiveX Automation 接口，它用来建立和 AutoCAD 对象之间的联系。在 VBA 下的编程要求对 ActiveX Automation 有基本的了解。在 ActiveX 和 VBA 参考中有对 AutoCAD ActiveX Automation 接口的说明。经验丰富的 VB 程序员将会看到，AutoCAD ActiveX Automation 接口对理解和开发 AutoCAD VBA 应用程序将非常有用的。

第三个要素是 VBA 编程环境。VBA 编程环境有它自己的一套对象、关键词、常量，并且提供程序流程、控制、调试和执行。微软的 VBA 外部在线帮助包含在 AutoCAD VBA 中，并且可以通过以下几种方法从 VBA IDE 对它进行访问。

- (1) 键盘上的 F1 键；
- (2) 从 VBA IDE 菜单上选 Help；
- (3) 单击 VBA IDE 工具条上的 Question 标志按钮。

2. 使用 AutoCAD 时的限制

为了保证 AutoCAD ActiveX 和 VBA 确定的功能性，要求开发者必须遵守以下系统配置要求：

● Windows NT 4.0

特别推荐安装使用 Windows NT 4.0 Service Pack 3 来运行 AutoCAD、ActiveX 和 VBA。

● Windows 95 或者 Windows 98

Microsoft 对此没有特别的要求。

● 安装、重装、卸载 Microsoft Office 或其它 VBA 应用程序

如果用户在安装 AutoCAD 后安装、重装或者卸载 Microsoft Office 或其它 VBA 应用程序，就必须重新安装 AutoCAD。特别要注意的是：每一次安装 AutoCAD 之后，最好重新启动系统。

0.3 AutoCAD ActiveX 和 VBA

相对其它的 AutoCAD API 环境，AutoCAD ActiveX/VBA 接口在以下几个方面表现出了它的优点：

● 速度

因为使用 VBA 在程序中运行，ActiveX 应用程序比 AutoLISP 或者 ADS 应用程序都要快得多。

● 使用方便

编程语言和开发环境都很方便使用，而且和 AutoCAD 一起安装。

● Windows 交互开发性

ActiveX 和 VBA 是为和其它的 Windows 应用程序一起使用而设计的，并且提供一条很好的通过应用程序交换信息的途径。

● 快速成型

VBA 的快速接口开发为应用程序的成型提供近乎完美的环境，即使这些程序最终将会用另外的编程语言来开发。

● 程序员基础

在这个世界上有数以百万的 Visual Basic 程序员，AutoCAD、ActiveX 和 VBA 技术对这些程序员和更多的以后将学习 Visual Basic 的程序员开放了 AutoCAD 定制和程序开发。

0.4 本书组成结构

这本书将介绍关于使用 AutoCAD2000 开发 ActiveX 和 VBA 应用程序的信息。第一章为 VBA 起步，这一章是关于使用 VBA 开发应用程序的信息说明；第十一章为使用 VBA 开发应用程序，从 VBA 以外的其它开发环境中使用过 ActiveX 的程序员可以跳过这两章。不过应当注意的是，所有本书中的实例代码都可以在 VBA 下运行。

第十三章，“设计花园小径 —— 一个 ActiveX 和 VBA 的教程”，这个指南将引导程序员新手在 AutoCAD 中使用 ActiveX 和 VBA 创建一个 garden path。推荐所有 AutoCAD ActiveX 和 VBA 的新用户都从这里开始。

0.5 本书的约定

这本书的内容假定读者对 Visual Basic 编程语言已经有了一定的编程经验，而且不用重复大量 Visual Basic 中有用的文档。如果读者需要了解更多的关于 Visual Basic 语言和开发环境的用法，请看微软公司开发的 Visual Basic 程序帮助文件，这个帮助文件可以从交互式开发环境（IDE）中的帮助按钮得到。

0.6 本书的样本代码

本书和 ActiveX VBA 参考总共包含了超过 800 个 VBA 子程序的例子，这些子程序将演示 ActiveX 的方法、属性和事件的使用方法。

在 AutoCAD2002 的/ACAD2000/sample 目录下也有很多应用程序实例。这些样本应用程序可以示范很多函数用法，包括从 AutoCAD 图形中提取数据到 Microsoft Excel 的电子数据表完成绘图。这些范例将向你展示怎样联合 Visual Basic 应用程序编程环境的多功能性和利用 AutoCAD ActiveX 接口建立定制的应用程序的能力。

1. 本书中范例代码的运行

ActiveX 和 VBA 开发者指南以及 ActiveX 和 VBA 参考中的所有范例代码都可以从 AutoCAD 帮助文件中拷贝出来，直接粘贴到 AutoCAD VBA 编程环境中，然后和以下的要求一起执行：AutoCAD 中当前激活的图形必须是空的，图形也必须开放到模型空间。另外，在这些手册中的样本代码可以从 AutoCAD 的/ACAD200/Sample 目录中的 SampleCode.dvb 和 Events.dvb 两个文件中得到。

运行实例的步骤如下：

- (1) 从帮助文件中把实例拷贝出来粘贴到空的 VBA 代码模型中；
- (2) 检查 AutoCAD 应该有一个开放到模型空间的空白图；
- (3) 敲入 VBARUN 命令打开宏对话框；
- (4) 选定宏并单击 Run 按钮。

另外，在“Running a Macro”中有更多可利用的运行的宏和宏对话框的信息。

2. 总结样本程序

在 AutoCAD 的/ACAD2000/Sample 目录中有 21 个样本程序，下面表 0.1 列出了这 21 个程序的程序名，描述和每一个样本程序主要源代码文件的所在位置。很多样本程序都有支持文件，这些支持文件也可以从和相应主程序源代码文件的同一目录中得到。和相应主程序源代码文件所在的同一目录中也有一个 readme.txt 文件，它是程序的说明，它也将说明怎样运行程序。

表 0.1 ActiveX 和 VBA 的样本程序：

程序名	程序说明	目录
Ibeam 3D	创建一个 3D 实体 IBeam，可以动态调整大小	/ACAD2000/Sample/VBA/ibeam3d.dvb
Map to Globe	在球体上从 2D 多义线生成 3D 多义线	/ACAD2000/Sample/VBA/Map2Globe.dwg
Menu Customization	示范使用菜单组和菜单条对象	/ACAD2000/Sample/VBA/Menu.dvb
Object Tracker	使用 Xrecord 数据跟踪对象修改	/ACAD2000/Sample/VBA/ObjectTracker.dvb
Save As R12	把 AutoCAD2000 图形文件存成 AutoCADR14 文件格式	/ACAD2000/Sample/VBA/
Tower	画一个传真塔并并进行压力分析	/ACAD2000/Sample/VBA/Tower.dwg
Text Height	改变一个图形中的所有文字高度	/ACAD2000/Sample/VBA/txtht.dvb
External Call	VBA 宏,调用一个注册过的动态连接库的六个函数	/ACAD2000/Sample/VBA/ExternalCall/ExternalCall.dvb
Facility	演示把 AutoCAD 连接到数据库	/ACAD2000/Sample/VBA/Facility/setup/setup.exe
VBA IDE Customization	在 VBA IDE 中创建一个新的工具栏,允许用户装载一个工程,并加入 VBA 管理器,VBA 宏,VBA 选项对话框	/ACAD2000/Sample/VBA/VBAIDEMenu/acad.dvb
Extract Attributes	这个 Excel 宏提取 AutoCAD 块的数据并存入电子数据表格	ACAD2000/Sample/ActiveX/ExtAttr/ExtAttr.xls
Attribute Text	该 AutoCAD 宏提取属性数据到 Microsoft Word 文档和 Midrosoft Excel 电子数据表和图中	/ACAD2000/Sample/VBA/atttext.dvb
Excel Link	这个 AutoCAD 宏是怎样把数据从 AutoCAD 传送到 Excel,然后重新返回到 AutoCAD 的	/ACAD2000/Sample/VBA/ExcelLink.dvb