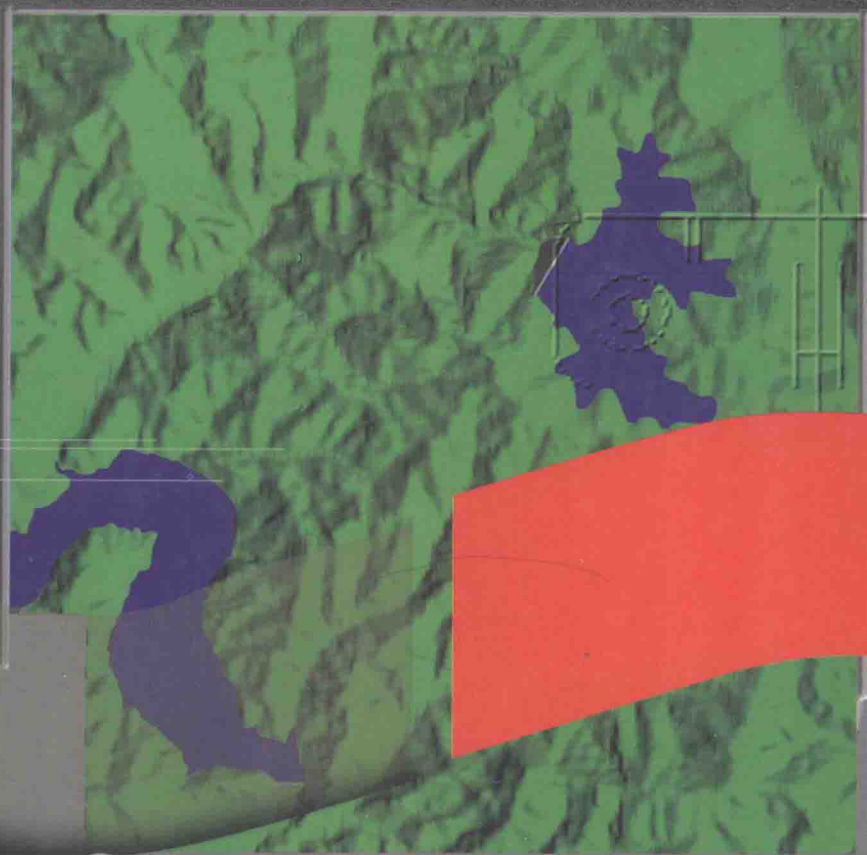


**AutoCAD 2000**  
开发技术丛书



# AutoCAD 2000 VBA 开发技术



清华大学出版社  
<http://www.tup.tsinghua.edu.cn>



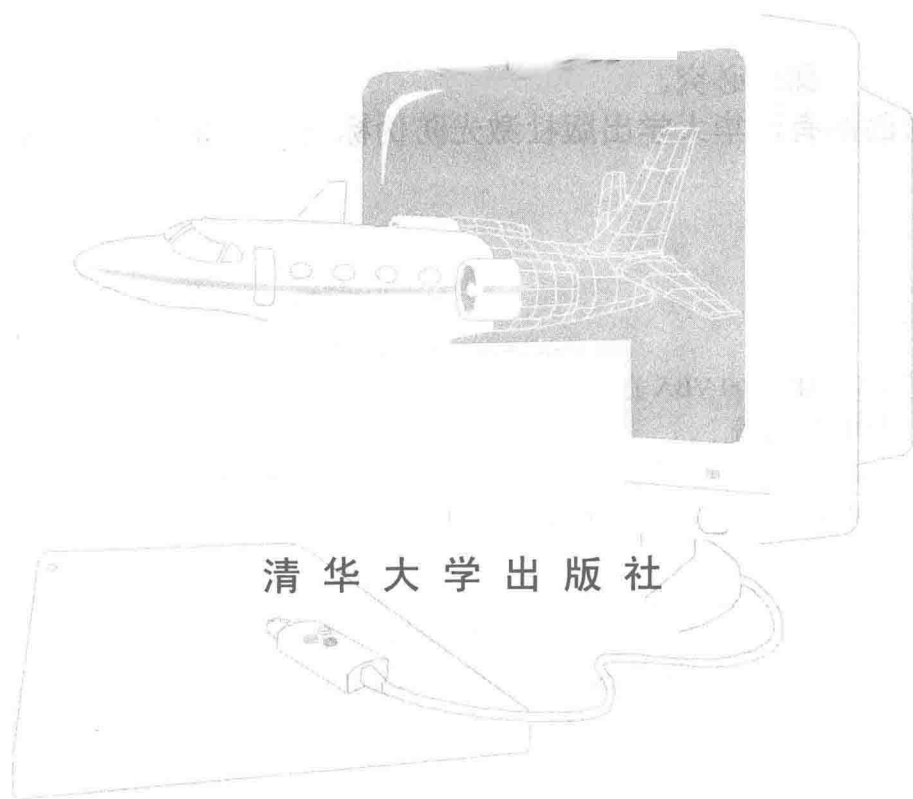
张国宝 编著

# AutoCAD 2000 VBA

---

## 开发技术

张国宇 编著



清华大学出版社

(京)新登字 158 号

### 内 容 简 介

AutoCAD 是目前最为流行的工程图形处理软件,它不但具有完善的绘图功能、良好的用户界面,而且允许用户进行二次开发。以前,开发者一般使用 LISP 语言或基于 C 语言的 ADS, ARX 等开发工具;在 AutoCAD 2000 中,用户可以使用内嵌的 VBA 来开发 AutoCAD。VBA 语法简单,功能强大,易学易用,是各种水平的用户都能接受的开发方式。

本书详细介绍了 VBA 的使用方法,在内容安排上充分考虑到初、中级用户的接受能力,着重讲解最为实用的开发技术,并辅以大量的应用实例。主要内容包括 AutoCAD VBA 综合开发环境、VBA 工程、VBA 对象模型以及各对象可以完成的主要操作、实体对象的创建与编辑、视图区和视图操作、布局和出图、用户坐标系、选择集操作、工具栏和菜单、实用编程技术等。不论对于 AutoCAD 的初学者,还是有经验的 AutoCAD 开发人员,都会在本书中有所收获。

本书可供工程技术人员及大专院校师生使用,也可作为培训班教材。

**版权所有,翻印必究。**

**本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。**

书 名: AutoCAD 2000 VBA 开发技术

作 者: 张国宝 编著

出版者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京密云胶印厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 20.25 字数: 463 千字

版 次: 2000 年 10 月第 1 版 2000 年 10 月第 1 次印刷

书 号: ISBN 7-302-00960-0/TP·353

印 数: 0001~5000

定 价: 30.00 元

# P 前 言

## reface

AutoCAD 2000 面市以来得到了业界人士的高度赞誉，好评如潮，称之为革命性的辅助设计工具毫不为过。这除了 AutoCAD 2000 提供的多文档操作环境、一体化的打印输出体系等诸多软件包本身的强大功能之外，其方便灵活的定制手段同样是成功的重要因素之一。

在 AutoCAD 2000 中，从菜单和工具栏直到每一个图形对象，用户都可以精确地定制。除了使用传统的 AutoLISP，基于 C 语言的 ObjectARX 等诸多开发工具以外，用户还可以使用 Visual Basic，AutoCAD VBA，Visual LISP 等新的开发方式。AutoCAD 2000 已将 AutoCAD VBA，Visual LISP 集成在系统内部，用户使用更加方便。

### 1. AutoCAD 2000 中的 ActiveX 技术

ActiveX 使用户可以利用 VB，VBA 等外部或内部程序来操作 AutoCAD。ActiveX 是微软公司提出的一个基于 COM (Component Object Model，直译为“部件对象模式”)的技术标准，以前称为 OLE (对象连接与嵌入)。其宗旨是在 Windows 系统的统一管理下，协调不同的应用程序，允许这些应用程序之间相互沟通、相互控制。每个应用程序可以决定将自己的哪些信息暴露出来，这些暴露出来的信息可以被其他程序操纵。这个暴露自己信息的应用程序称为服务程序 (server)，而操纵服务程序的程序称为客户程序 (client)。

Autodesk 公司对微软公司的这个技术标准积极响应，在 AutoCAD R14 的开发后期，推出了基于 ActiveX 技术的开发方式，并随正式版免费附送了 AutoCAD R14 VBA 预览版本。在 AutoCAD 2000 中，Autodesk 公司明显地加强了 VBA 的功能，除了弥补一些以前版本出现的各种错误以外，还增加了许多新的对象、方法和属性，大大提高了二次开发 AutoCAD 的能力。这正是广大 AutoCAD 用户和专业开发者所希望的。

应用 AutoCAD ActiveX 技术，可以从其他外部应用程序 (如 Word，Visual Basic 等) 或 AutoCAD 内部 (AutoCAD VBA) 来操作 AutoCAD 暴露的对象。这些暴露的对象可以被任何支持 ActiveX 的技术并可以作为客户程序的应用程序来操纵。常见的可作为客户程序的软件包括：Visual Basic (4.0 以上版本)，Excel 95/97，Word 95/97，Access 95/97，PowerPoint 97 等。VB 也是我们最为常用的支持 ActiveX 技术的开发工具，其语法简单，功能强大，并提供了建立可执行代码的源代码编译器。

应用程序通过 ActiveX 技术暴露对象的差别很大。例如，微软的 Excel 等允许客户程

序处理其内部数据及用户界面，可以改变一个 Excel 工作表上单元格的内容，也可以在 Excel 菜单上增加一项，甚至做更大的改动，使其在外表上看去完全不像 Excel，倒像被人开发好的应用程序。而另外有些应用程序，比如微软的 Visual Basic，只对客户程序暴露其很小的一部分。

在暴露对象多少这个问题上，AutoCAD 2000 较 R14 版本明显地增加了许多。在 AutoCAD 2000 中，几乎对外暴露了所有对象，包括图形实体对象（如：直线、圆弧）、样式设置对象（如：线型、文本样式）、组织结构对象（如：层、块）、图形显示对象（如：视图、视图区）以及菜单、工具栏等，也就是说用户可以操作几乎所有的 AutoCAD 对象。

## 2. AutoCAD VBA

VBA 是 Visual Basic for Application 的缩写，由 Visual Basic 派生而来，现在已经成为 Microsoft 产品的标准语言。与 VB 一样，VBA 是面向对象的程序设计语言，它继承了 VB 语法简单、功能强大的特点，同时，由于 VBA 可与主程序在同一内存空间内运行，大大提高了运行速度，并具有某些“智能”功能。

每一套 AutoCAD 2000 都包含功能强大、易学易用的 VBA 5.0 程序开发环境，这个集成开发环境提供了高质量的用户化编程能力。

(1) VBA 能够使 AutoCAD 与其他应用程序直接共享数据，如 Microsoft Office 97 软件。因为所有的 VBA 应用程序都可以方便地相互沟通，在设计过程中可以很容易地将 AutoCAD 数据与基于 Windows 的应用程序联系起来。例如，可以使用 AutoCAD 2000 生成材料表并直接输出电子数据表为财务部门使用。绘制图纸，然后用 VBA。

(2) 在 VBA 综合开发环境中，可以通过拖动图标的方法，简单而可视化地建立用户界面。用户可以使用 AutoCAD 工具栏，也可以使用标准 VBA 界面工具创建新的工具栏。

(3) VBA 的语言代码完全继承了 VB 的特征，可以说是世界上最简单的编程语言，但其功能不弱。从数据计算、数据库管理到 Internet 软件开发，都可以见到 VB 的身影，并可通过各种在线帮助，随时得到名师真传。

(4) 具有全开放式的对象模型。AutoCAD 2000 软件中的 ActiveX 对象模型十分有用，它显示了每一个对象的方法和属性，用户可以按照自己的愿望编程或定制 AutoCAD 2000 软件。

(5) 良好的运行速度。VBA 是一个内嵌式的控制器，共享 AutoCAD 的内存空间，可以更好地与 AutoCAD 沟通。其运行速度与 ObjectARX 应用程序非常接近，比其他开发方式（如 ADSRX，ADS 和 LISP）要提高许多，见图 1 所示。

无论编程技术水平如何，使用 VBA 编程都不会感到有很多困难。自从带有 VBA 开发接口的 AutoCAD R14 发行以来，已经有超过 80 万的开发者使用了这一软件，开发的应用程序将有更多机会为不断增长的用户群体服务。

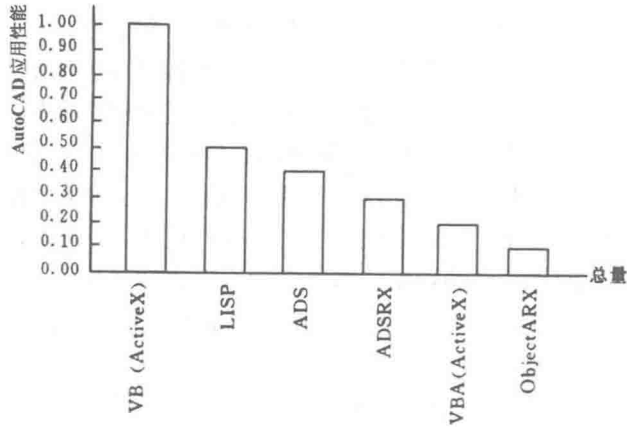


图 1 VBA 与其他开发方式的性能比较

### 3. VBA 的安装环境

AutoCAD 2000 VBA 是嵌入在 AutoCAD 2000 中的开发环境，同 AutoCAD 2000 一起安装。AutoCAD 2000 推荐的最低配置见表 1。

表 1 AutoCAD 2000 推荐的最低配置

CPU	Pentium 133
内存	32MB
剩余硬盘空间 (安装)	150MB
虚拟内存	64MB (由操作系统管理)
操作系统	Windows 9x 或 Windows NT4.0

### 4. 本书的结构安排

本书介绍了用 AutoCAD VBA 定制 AutoCAD 2000 这一全新、便捷的开发方式。在结构安排上，既考虑了 AutoCAD 2000 作为工程制图工具的方面，又考虑了 VBA 作为编程语言和环境的方面，以对象模型为贯穿全书的主线，以完成一个具体的 AutoCAD 操作为目的，对较为熟悉的 AutoCAD 2000 本身的内容没有详细论述，起点较低的读者可以在使用本书的同时参照有关 AutoCAD 方面的书籍；考虑到 AutoCAD 2000 的许多新功能和新概念大部分读者可能是初次接触，因此对于这部分内容本书作了较详细的阐述。

本书各章内容如下。

第 1 章介绍 AutoCAD VBA 综合开发环境、VBA 工程、宏等基本知识，这部分内容是开发 VBA 程序必须掌握的。

第 2 章介绍 AutoCAD VBA 对象模型以及各对象可以完成的主要操作。在面向对象的程序设计中，对象模型是程序设计的基础，完成任何操作都要按照对象模型来定位对象。

第 3 章从打开 AutoCAD 图形文件入手，介绍了有关应用程序和图形文件操作方面的知识，如系统窗口和图形窗口的设置，文件的打开、关闭，系统变量的存取等。

第 4 章是本书的重点，详细论述了实体对象的创建、编辑等方面的内容。

第 5 章和第 6 章介绍了视图、视图区、布局和打印操作等方面的内容。

第 7, 8, 9, 10, 11 章分别介绍了在 VBA 程序中使用用户坐标系、操作选择集、设置 AutoCAD 的运行环境、定制 AutoCAD 菜单和工具栏的方法，以及 AutoCAD VBA 的事件操作等方面的内容。相对而言，这部分是程序设计中较为高级的内容。

第 12 章的内容包括在程序设计中较为实用的操作技术，例如扩展数据的操作、外部文件的读取，以及与其他支持 ActiveX 技术的应用程序进行数据交互等方面的内容。

在第 13 章中，列举了两个实例程序。通过这章的内容，读者可基本掌握 VBA 程序设计的整个过程。

另外，本书包括了 4 个附录。附录 A 介绍了 AutoLISP 函数与 VBA 方法、属性的对应关系，以便熟悉 AutoLISP 的读者可以很快接受 AutoCAD VBA。在附录 B, C, D 中，列出了 AutoCAD 2000 的所有对象、方法、属性。

在本书的编写过程中，得到了张志修、张晓今、赵阵庆等有关领导的大力支持，得到了同事们的帮助和理解，许多同志为本书提出了中肯的建议，在此一并表示诚挚的感谢。

由于编者水平有限，本书也许存在疏漏和错误之处，望各界专家、读者提出宝贵意见。  
E-mail: about cad@263.net

编者于北京勘测设计研究院  
1999 年 7 月



# C 目 录

Content

## 第 1 章 走进 AutoCAD VBA

1.1	认识 VBA IDE.....	1
1.2	VBA 工程文件 .....	4
1.2.1	嵌入式工程和通用式工程 .....	4
1.2.2	工程文件的组成 .....	4
1.2.3	添加工程部件 .....	5
1.2.4	用工程资源管理器查看信息 .....	5
1.2.5	保护工程 .....	5
1.3	工程文件的编辑 .....	6
1.3.1	启动 VBA IDE .....	7
1.3.2	导入存在的文件 .....	7
1.3.3	文件编辑 .....	8
1.3.4	命名和保存工程 .....	10
1.3.5	引用其他 VBA 工程 .....	11
1.3.6	设置 VBA IDE 选项 .....	12
1.4	用 VBA 管理器组织工程 .....	13
1.4.1	加载一个工程 .....	13
1.4.2	卸载一个工程 .....	15
1.4.3	将工程嵌入到图形文件中 .....	15
1.4.4	从图形文件中导出工程 .....	15
1.4.5	创建新工程 .....	15
1.5	宏操作 .....	16
1.5.1	宏的运行 .....	16
1.5.2	编辑宏 .....	16
1.5.3	宏的单步运行 .....	16
1.5.4	创建新宏 .....	17
1.5.5	宏的删除 .....	17
1.5.6	设置宏选项 .....	17
1.6	一个简单的 VBA 实例 .....	17



1.7 与 VBA 有关的 AutoCAD 2000 的命令 .....	19
--------------------------------------	----

## 第 2 章 AutoCAD ActiveX 对象

2.1 使用对象变量.....	20
2.1.1 声明对象变量.....	20
2.1.2 对象引用的赋值.....	21
2.1.3 对象引用的释放.....	21
2.2 对象的方法、属性和事件.....	21
2.2.1 属性的设置和查询.....	22
2.2.2 在代码中使用方法.....	22
2.2.3 为对象的事件编写代码.....	23
2.3 AutoCAD 2000 的对象模型.....	23
2.4 根据对象模型访问对象.....	26
2.4.1 访问 AutoCAD 对象时的切入点.....	26
2.4.2 定义中间变量来引用常用对象.....	27
2.5 集合对象.....	27
2.5.1 引用集合对象.....	29
2.5.2 用 Add 方法添加集合成员.....	30
2.5.3 用 Load 方法添加集合成员.....	32
2.5.4 用 AddEntityName 型方法添加集合成员.....	32
2.5.5 检索集合的成员.....	33
2.5.6 集合枚举操作.....	33
2.6 Variant 型数据在 VBA 程序中的应用.....	34
2.6.1 用 Variant 型数据表示点坐标.....	35
2.6.2 用 Variant 型数据表示其他数组数据.....	36
2.6.3 Variant 型数据在表示输入数据和输出数据时的区别.....	36
2.6.4 将数组转换成 Variant 型数据.....	37

## 第 3 章 AutoCAD 应用程序和图形文件操作

3.1 控制 AutoCAD 应用程序窗口.....	38
3.1.1 改变窗口的大小和位置.....	38
3.1.2 设置和查询窗口状态.....	38
3.2 图形文件的建立、保存、打开和关闭.....	39
3.2.1 建立图形文件.....	39
3.2.2 保存图形文件.....	40

3.2.3	打开图形文件 .....	40
3.2.4	关闭图形文件 .....	41
3.3	控制图形文件窗口 .....	41
3.3.1	窗口的大小、位置和当前状态 .....	41
3.3.2	窗口或实体的显示 .....	43
3.3.3	其他与窗口有关的操作 .....	44
3.4	设置文件的当前属性 .....	44
3.5	系统变量的存取 .....	45
3.5.1	设置系统变量 .....	45
3.5.2	查询系统变量 .....	46
3.6	调用 AutoCAD 命令 .....	46
3.7	在无文件状态下工作 .....	47

## 第 4 章 实体对象的创建与编辑

4.1	实体对象在对象模型中的定位 .....	49
4.2	创建基本实体 .....	50
4.2.1	创建点和直线 .....	51
4.2.2	创建曲线 .....	53
4.2.3	创建实心填充区域 .....	55
4.2.4	创建阴影线 .....	56
4.2.5	创建面域 .....	57
4.3	一般实体的编辑方法 .....	58
4.3.1	引用实体对象 .....	58
4.3.2	实体对象的编辑 .....	59
4.3.3	编辑多义线 .....	63
4.3.4	编辑样条曲线 .....	65
4.3.5	编辑阴影线 .....	67
4.4	实体属性 .....	68
4.5	文本、公差和形状 .....	69
4.5.1	文本样式对象 .....	69
4.5.2	用 AddText 方法创建单行文本 .....	71
4.5.3	用 AddMtext 方法创建段落文本 .....	72
4.5.4	文本编辑 .....	74
4.5.5	公差对象 .....	75
4.5.6	形对象 .....	76
4.6	块和属性 .....	76

4.6.1	块的操作.....	77
4.6.2	属性操作.....	78
4.6.3	块引用和属性引用对象.....	79
4.7	外部引用对象.....	80
4.7.1	创建外部引用对象.....	80
4.7.2	外部引用对象的编辑.....	81
4.8	光栅图像.....	82
4.8.1	创建光栅图像.....	83
4.8.2	编辑光栅图像.....	83
4.9	尺寸标注对象.....	85
4.9.1	创建尺寸标注的方法.....	85
4.9.2	创建标注对象的几点说明.....	87
4.9.3	尺寸标注对象的编辑.....	88
4.9.4	尺寸标注样式对象.....	89
4.9.5	忽视尺寸标注样式中的设置.....	91
4.10	3D 面.....	93
4.10.1	用 AddFace 方法创建 3D 面.....	93
4.10.2	用 Add3DMesh 方法创建多边形网格.....	93
4.10.3	用 AddPolyFaceMesh 方法创建多面网格.....	95
4.11	3D 实心体.....	95
4.12	自定义对象.....	96

## 第 5 章 视图区和视图操作

5.1	AutoCAD 的视图区.....	98
5.1.1	平铺视图区.....	99
5.1.2	浮动视图区.....	102
5.2	视图对象.....	103

## 第 6 章 布局 and 出图

6.1	布局、块、模型空间和图纸空间.....	106
6.2	创建布局对象.....	107
6.3	模型空间和图纸空间布局的切换.....	108
6.3.1	切换到图纸空间布局.....	108
6.3.2	切换到模型空间布局.....	109
6.4	布局的页面设置.....	109
6.4.1	设置纸张尺寸和出图单位.....	110

6.4.2	调整出图原点.....	110
6.4.3	设置出图区域.....	110
6.4.4	设置出图比例.....	110
6.4.5	设置线重比例.....	111
6.4.6	设置出图设备.....	111
6.5	绘图输出.....	111
6.5.1	初始化绘图设置.....	111
6.5.2	在模型空间出图.....	112
6.5.3	在图纸空间出图.....	112
<b>第 7 章 在用户坐标系中工作</b>		
7.1	创建用户坐标系.....	114
7.2	坐标变换.....	114
7.3	在用户坐标系中创建实体.....	115
<b>第 8 章 选择集操作</b>		
8.1	创建选择集.....	118
8.2	向选择集中添加实体.....	119
8.2.1	用 Select 方法添加实体.....	119
8.2.2	用 SelectAtPoint 方法添加实体.....	121
8.2.3	用 SelectOnScreen 方法添加实体.....	121
8.2.4	用 SelectByPolygon 方法添加实体.....	122
8.2.5	用 GetEntity 方法和 AddItems 方法添加实体.....	123
8.2.6	使用过滤机制选择实体.....	124
8.3	选择集的删除操作.....	129
<b>第 9 章 设置 AutoCAD 2000 的运行环境</b>		
9.1	引用优先设置对象.....	131
9.2	设置环境选项.....	131
9.3	设置图形文件选项.....	131
<b>第 10 章 工具栏和菜单</b>		
10.1	基本概念.....	133

10.2	装入菜单文件.....	134
10.3	创建新的菜单组.....	134
10.4	改变菜单栏.....	135
10.4.1	在菜单栏上插入菜单.....	135
10.4.2	从菜单栏上移走下拉菜单.....	136
10.4.3	重新排列下拉菜单.....	136
10.5	创建和编辑菜单.....	137
10.5.1	创建新菜单.....	137
10.5.2	添加一个新的菜单项.....	138
10.5.3	添加菜单项分隔符.....	139
10.5.4	指定菜单项的快捷键.....	140
10.5.5	创建级联式下级菜单.....	140
10.5.6	从菜单中删除一个菜单项.....	141
10.5.7	菜单项的属性.....	142
10.6	工具栏的创建与编辑.....	144
10.6.1	创建工具栏.....	144
10.6.2	在工具栏中添加工具按钮.....	144
10.6.3	添加工具按钮分隔符.....	146
10.6.4	定义工具按钮图标.....	146
10.6.5	工具栏的浮动和停靠.....	148
10.6.6	删除工具按钮.....	149
10.6.7	使用工具按钮的属性.....	149
10.7	宏的创建.....	149
10.7.1	宏字符的 ASCII 码值.....	150
10.7.2	中断宏的运行.....	151
10.7.3	在宏中提示用户输入数据.....	152
10.7.4	取消命令.....	152
10.7.5	宏的连续运行.....	153
10.7.6	使用单步对象选择模式.....	153
10.8	状态行帮助信息的使用.....	153
10.9	向快捷菜单中添加菜单项.....	154

## 第 11 章 使用事件

11.1	过程参数的传递方式.....	157
11.2	AutoCAD 中的事件.....	158
11.2.1	三种类型的事件.....	158

11.2.2	编写事件过程的几个限制 .....	159
11.3	应用程序级事件 .....	160
11.3.1	编写事件代码 .....	160
11.3.2	BeginFileDrop 事件 .....	162
11.3.3	AutoCAD 2000 中的应用程序级事件 .....	163
11.4	文档级事件 .....	163
11.4.1	AutoCAD 2000 中的文档级事件 .....	164
11.4.2	在 VBA 程序中使用文档级事件 .....	165
11.4.3	在 VBA 以外的环境中使用文档级事件 .....	165
11.5	一般对象级事件 .....	167
11.6	在 Visual Basic 程序中使用 AutoCAD 的事件 .....	168

## 第 12 章 实用编程技术

12.1	实用工具对象 .....	171
12.1.1	用户输入方法 .....	172
12.1.2	数据类型转换 .....	177
12.2	扩展数据和扩展记录 .....	179
12.2.1	理解组码 .....	179
12.2.2	扩展数据的组织 .....	180
12.2.3	设置对象的扩展数据 .....	181
12.2.4	读取扩展数据 .....	183
12.2.5	扩展记录 .....	184
12.3	在实体中保存超级链接 .....	186
12.4	查询两个实体的交点 .....	188
12.5	查询包含实体的外框 .....	189
12.6	与其他程序交换信息 .....	190
12.6.1	与 LISP 程序交换信息 .....	190
12.6.2	操作其他 Windows 应用程序 .....	191
12.6.3	操作 Windows API .....	193
12.7	在 VBA 中访问外部文件 .....	194
12.7.1	文件管理的语句和函数 .....	194
12.7.2	文件的读写操作 .....	194
12.8	图形数据交换 .....	199
12.8.1	用 Export 方法导出图形 .....	199
12.8.2	用 Import 方法导入图形 .....	200

## 第 13 章 两个实例程序

13.1	绘制铁轨线的程序.....	201
13.1.1	设计窗体界面.....	202
13.1.2	编写代码.....	203
13.2	绘制花园路径的程序.....	208
13.2.1	设计程序界面.....	208
13.2.2	编写程序代码.....	210

## 附录

附录 A	Visual LISP 和 ActiveX/VBA 的比较.....	217
附录 B	AutoCAD ActiveX/VBA 对象.....	224
附录 C	AutoCAD ActiveX/VBA 方法.....	281
附录 D	AutoCAD ActiveX/VBA 属性.....	291



# 第 1 章 走进 AutoCAD VBA

本章将讲解 VBA 的基础知识，主要包括 VBA 集成开发环境（IDE）、工程文件、宏等方面的内容，这些都是 VBA 程序设计中非常重要的概念。

## 1.1 认识 VBA IDE

VBA IDE 是进行 VBA 程序设计所必需的集成开发环境，它集 VBA 程序的代码编写、界面设计、程序修改和调试为一体，可充分节省程序员的时间和精力。

在 AutoCAD 的命令行中键入“VBAIDE”命令后，进入 VBA IDE（见图 1.1）。在这个集成开发环境中，除了 Windows 应用程序普遍带有的菜单栏和工具栏之外，还有许多新的东西，它们是我们开发 AutoCAD VBA 程序的得力助手。简要介绍如下。

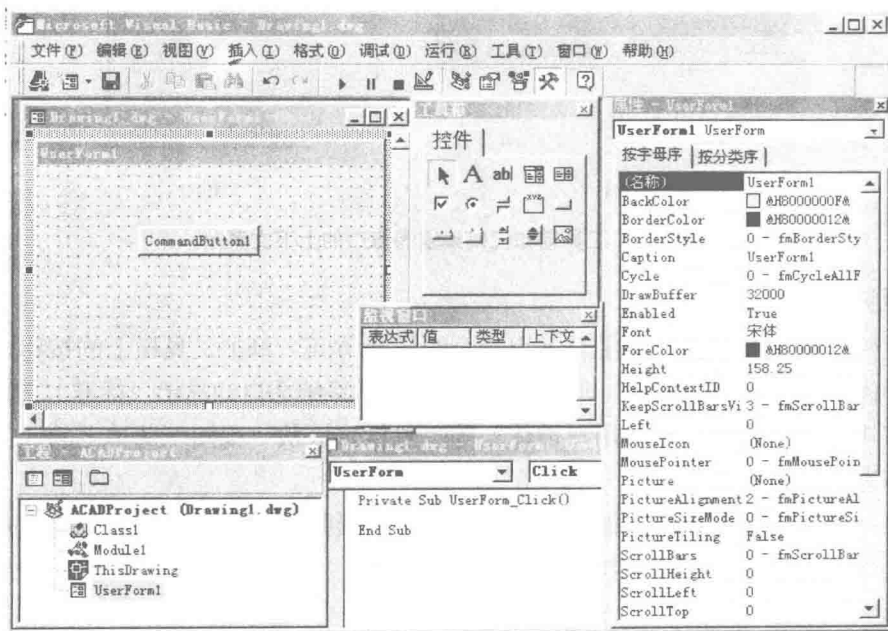


图 1.1 AutoCAD VBA 集成开发环境

### 1. 菜单栏

菜单栏是 VBA IDE 启动命令的主要手段。在菜单栏中显示了所有的 VBA 菜单，使用鼠标单击某个菜单时就选择了相应的命令。菜单栏除了提供标准的“文件”、“编辑”、“视图”、“窗口”和“帮助”等较为常见的菜单之外，还提供了编程专用的功能菜单，如“工具”、“格式”和“调试”等。

## 2. 上下文菜单

上下文菜单并不是自动可见的，在要使用的对象上单击鼠标右键才可打开上下文菜单。菜单中包括与对象有关的操作快捷键。之所以称之为“上下文菜单”，就是因为菜单的内容可随着当前环境（即上下文）而变化，也就是说菜单中有效的菜单项取决于单击鼠标时的环境。例如，在“工具箱”上单击鼠标右键时，弹出的上下文菜单包括“附加控件”、“删除”、“自定义”等菜单项（见图 1.2(a)）；而在代码编辑器窗口中单击鼠标右键，则弹出与代码编辑有关的菜单（见图 1.2(b)）。

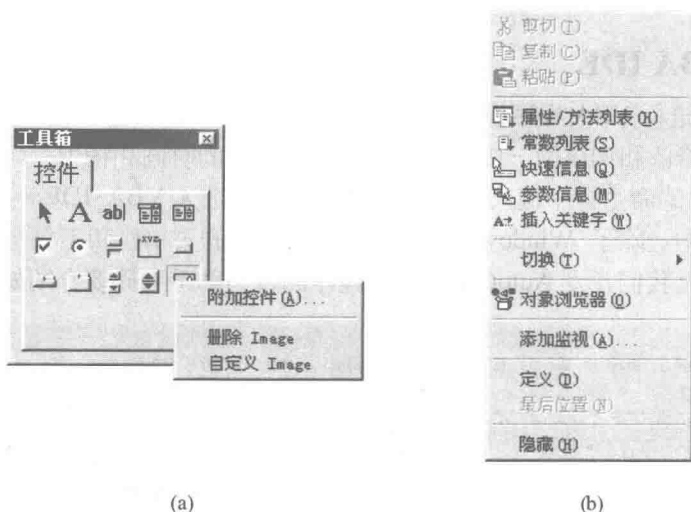


图 1.2 工具箱和代码编辑器窗口的上下文菜单

## 3. 工具栏

工具栏提供了在编程环境中对常用命令的快速访问。单击工具栏上的按钮，则执行该按钮所代表的操作。按照默认规定，启动 VBA 后系统会自动显示“标准”工具栏，其他如“编辑”、“调试”和“用户窗体”等工具栏可以从“视图”菜单上的“工具栏”命令中移进或移出。

工具栏能紧贴在菜单栏下面，也能以垂直条状紧贴在左边框上。如果将它从菜单下面拖开，则它能“悬”在窗口中。

## 4. 工具箱

工具箱提供一组工具，用于设计时在窗体中放置控件。除了默认工具之外，还可以通过上下文菜单中选定“附加控件”来使用其他 ActiveX 控件，见图 1.2。

## 5. 工程资源管理器窗口

工程是指用于创建一个应用程序的文件的集合。在工程资源管理器窗口中，将列出当前工程中的对象、窗体、模块、引用等，见图 1.3。