

佟士懋 邢芳芳 夏齐霄 编著

AutoCAD ActiveX/VBA 二次开发技术基础及应用实例



随书赠送光盘一张



国防工业出版社
National Defense Industry Press

AutoCAD ActiveX/VBA 二次开发技术基础及应用实例

佟士懋 邢芳芳 夏齐霄 编著

国防工业出版社

·北京·

图书在版编目(CIP)数据

AutoCAD ActiveX/VBA 二次开发技术基础及应用实例/
佟士懋, 邢芳芳, 夏齐霄编著. —北京: 国防工业出版社, 2006. 4

ISBN 7-118-04475-X

I. A... II. ①佟... ②邢... ③夏... III. 计算机辅助设计—应用软件, AutoCAD IV. TP391.72

中国版本图书馆 CIP 数据核字(2006)第 023680 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100044)

北京四季青印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 12½ 字数 280 千字

2006 年 4 月第 1 版第 1 次印刷 印数 1—3500 册 定价 29.00 元(含光盘)

(本书如有印装错误, 我社负责调换)

国防书店: (010)68428422

发行邮购: (010)68414474

发行传真: (010)68411535

发行业务: (010)68472764

前 言

众所周知,AutoCAD是目前全球使用最广泛的一种工程图形设计软件。它强大的生命力不仅在于其完善的绘图功能、良好的用户界面,而且还在于它具有开放式的结构体系和众多的系列化产品,并为用户提供了多种二次开发的工具和方法。

AutoLISP是AutoCAD用于二次开发的高级宏汇编语言,由于推出较早,具有较广泛的使用基础。但作为解释语言,其运行速度和保密性方面有一定的缺陷。在AutoCAD R14版发布后,Autodesk公司推出了可编译的VisualLISP语言,将LISP的源代码编译成.arx文件,大大增强了其开发应用程序的功能。到AutoCAD 2000版,VisualLISP已经成为AutoLISP的替换版本,作为软件中的一个标准模块,内嵌在AutoCAD软件中。

在AutoCAD R11版发布后,Autodesk公司又推出了基于C语言的开发系统ADS(AutoCAD Development System)。AutoCAD R13版新增了以C++为基础的面向对象的开发环境及应用程序接口ObjectARX(AutoCAD Runtime eXtension),利用它开发的应用程序是Windows DLL程序,可直接与AutoCAD进行通信。这种开发工具功能强大、运行效率高。

在Autodesk公司开发AutoCAD R14版的后期,又推出了基于ActiveX Automation(自动化)技术的新型开发方式。在AutoCAD中使用ActiveX接口主要具有如下两个优点。

(1)可以在多种编程环境中编程访问AutoCAD图形。在ActiveX Automation技术出现之前,开发者只能用AutoLISP或C++等访问AutoCAD图形。

(2)更易于与其他Windows应用程序(例如Microsoft Excel和Microsoft Word等)共享数据。

本书将介绍利用VBA/VB对AutoCAD 2004/2005/2006进行二次开发的技术基础、编程经验和应用实例。本书共分8章,每章都有大量的编程范例和综合示例。考虑到利用VB编程的优点(通用性强、开发出的应用程序可编译为可执行文件及打包等),本书增添了介绍VB程序设计的基础知识等章节。

关于AutoCAD ActiveX/VBA的二次开发技术,大体可包括:VBA编程环境、AutoCAD ActiveX对象、AutoCAD环境的设置、组织图形元素、图形对象的创建与编辑、文字、标注和公差、图块和外部引用、表格、在三维空间工作、自定义工具栏和菜单、定义布局和打印、使用事件、与其他应用程序和Windows API交互以及错误处理等多项内容。由于本书实例部分主要介绍如何开发机械零件的二维和三维实体参数化自动绘图应用程序,因此相应地在技术基础部分将重点介绍:AutoCAD ActiveX对象、创建对AutoCAD部件对象的引用、AutoCAD环境的设置和组织图形元素、图形对象的创建与编辑、文字、标注和公差、图块、三维实体对象的创建与编辑等最基本的技术及编程经验,其他部分适

当从简或略去。在应用实例部分,将简略介绍依据 AutoCAD ActiveX/VBA 二次开发技术开发出的齿轮、V 带轮和轴等几类机械零件的 CAD 应用程序。使用这些应用程序时,只需在用户界面上输入绘图的参数,单击“运行”按钮后,即可立即自动绘出实用的(二维或三维实体)机械零件图。这部分内容对于如何开发复杂机械零件的参数化绘图应用程序将有较高的参考价值。此外,在本书的最后一章,将简略介绍名为《M-CAD 常用机械传动零件计算机辅助设计》的软件包,该软件包具备机械设计求算器和参数化绘图两种功能,参数化绘图部分是利用 AutoCAD ActiveX/VBA 开发技术的应用成果。

在本书的附录中,备有各章编程范例的索引表。在本书附带的光盘中,收录了各章综合应用示例的源程序及齿轮、V 带轮和轴等应用实例的可执行文件。利用这部分材料,不仅能帮助读者学习或演示 AutoCAD ActiveX/VBA 二次开发技术常用的功能和编程技巧,而且可以直接应用这些参数化绘图应用程序绘制实用的机械零件图。

《M-CAD 常用机械传动零件计算机辅助设计》软件包是佳睿和佟士懋等教授合作开发的。若读者希望使用及购买该软件包,可与出版社联系。

本书依照简练实用、突出重点的原则,以及结合参数化机械设计的需要对阐述的内容进行取舍。它既适合初学者学习,也可供从事计算机辅助机械设计及 AutoCAD ActiveX/VBA 二次开发技术的专业工作者阅读和参考。

本书由佟士懋教授和北京联合大学的邢芳芳、夏齐霄副教授编写,鉴于作者水平有限和时间仓促,书中疏漏或不足之处,敬请读者批评指正。

作者

2005.9

目 录

| | |
|---|----|
| 第 1 章 走进 AutoCAD ActiveX/VBA | 1 |
| 1.1 ActiveX 自动化技术在 AutoCAD 中的应用 | 1 |
| 1.2 AutoCAD VBA 基础 | 2 |
| 1.2.1 概述 | 2 |
| 1.2.2 VBA 工程及其管理 | 3 |
| 1.2.3 VBA 集成开发环境 | 4 |
| 1.2.4 宏及其管理 | 6 |
| 1.2.5 VBA 中的窗体 | 8 |
| 1.2.6 第一个 VBA 编程示例 | 9 |
| 第 2 章 Visual Basic 程序设计基础 | 13 |
| 2.1 第一个 VB 应用程序 | 13 |
| 2.1.1 创建 VB 应用程序引例 | 13 |
| 2.1.2 对象 | 16 |
| 2.1.3 事件和事件过程 | 16 |
| 2.2 编写 VB 的程序代码 | 17 |
| 2.2.1 程序书写规则 | 17 |
| 2.2.2 变量和赋值语句 | 17 |
| 2.2.3 常数 | 19 |
| 2.2.4 运算 | 19 |
| 2.2.5 流程控制语句 | 20 |
| 2.2.6 数组 | 22 |
| 2.2.7 使用过程和函数 | 23 |
| 2.3 集成开发环境及工程的管理 | 24 |
| 2.3.1 集成开发环境 | 24 |
| 2.3.2 工程的管理 | 24 |
| 2.4 设计应用程序的界面 | 26 |
| 2.4.1 设计窗体 | 26 |
| 2.4.2 在窗体上添加控件 | 26 |
| 2.4.3 消息框和输入框 | 28 |
| 2.4.4 窗体的生命周期 | 29 |
| 2.5 处理应用程序的错误 | 30 |
| 2.5.1 应用程序中的错误 | 30 |

| | | |
|------------|-----------------------------|-----------|
| 2.5.2 | 捕获运行时错误 | 30 |
| 2.5.3 | 响应捕获的错误 | 30 |
| 第3章 | AutoCAD ActiveX 对象 | 32 |
| 3.1 | 使用对象变量 | 32 |
| 3.1.1 | 对象变量 | 32 |
| 3.1.2 | 对象的属性和方法 | 32 |
| 3.1.3 | 使用事件 | 33 |
| 3.2 | AutoCAD的对象模型 | 34 |
| 3.2.1 | Application 对象和 Document 对象 | 34 |
| 3.2.2 | Preferences 对象 | 34 |
| 3.2.3 | 图形对象和非图形对象 | 34 |
| 3.2.4 | Collection 对象 | 36 |
| 3.2.5 | Plot 对象和 Utility 对象 | 36 |
| 3.3 | 根据对象模型访问对象 | 37 |
| 3.3.1 | 访问 AutoCAD 对象时的切入点 | 37 |
| 3.3.2 | 定义中间变量引用常用对象 | 37 |
| 3.4 | 创建对 AutoCAD 部件对象的引用 | 37 |
| 3.4.1 | 引用 AutoCAD 类型库 | 38 |
| 3.4.2 | 声明对象变量 | 38 |
| 3.4.3 | 将对象引用赋予变量 | 39 |
| 3.5 | 使用变体传递数组 | 40 |
| 3.5.1 | 创建变体类型的数组 | 41 |
| 3.5.2 | 使用变体类型的数组 | 41 |
| 3.6 | 创建对 AutoCAD 部件引用的综合示例 | 42 |
| 3.6.1 | 创建对 AutoCAD 部件引用的综合示例用户界面 | 42 |
| 3.6.2 | 创建对 AutoCAD 部件引用的综合示例源程序 | 43 |
| 第4章 | 控制 AutoCAD 环境和组织图形元素 | 44 |
| 4.1 | 控制应用程序窗口 | 44 |
| 4.1.1 | 改变应用程序窗口的位置和尺寸 | 44 |
| 4.1.2 | 设置和查询应用程序窗口状态 | 44 |
| 4.2 | 图形文件的建立、打开、保存和关闭 | 45 |
| 4.2.1 | 创建新图形 | 45 |
| 4.2.2 | 打开现有图形 | 46 |
| 4.2.3 | 保存图形 | 46 |
| 4.2.4 | 关闭图形 | 47 |
| 4.3 | 管理图形窗口 | 48 |
| 4.3.1 | 使用缩放 | 48 |
| 4.3.2 | 使用命名视图 | 50 |
| 4.3.3 | 使用平铺视口 | 50 |

| | | |
|--------------|----------------------------|-----------|
| 4.3.4 | 图形对象的重新生成和更新 | 52 |
| 4.3.5 | 重置当前对象 | 53 |
| 4.4 | 系统变量的存取 | 53 |
| 4.5 | 调用 AutoCAD 命令 | 55 |
| 4.6 | 组织图形元素 | 55 |
| 4.6.1 | 颜色 | 55 |
| 4.6.2 | 线型 | 56 |
| 4.6.3 | 线宽 | 58 |
| 4.6.4 | 图层 | 58 |
| 4.7 | 设置 AutoCAD 系统配置 | 60 |
| 4.8 | 设置 Document 对象的当前属性和确定容器对象 | 61 |
| 4.8.1 | 设置 Document 对象的当前属性 | 61 |
| 4.8.2 | 确定容器对象 | 62 |
| 4.9 | 输入输出其他图形文件格式 | 62 |
| 4.10 | 控制 AutoCAD 环境和组织图形元素的综合示例 | 64 |
| 第 5 章 | 图形对象的创建与编辑 | 67 |
| 5.1 | 创建图形对象 | 67 |
| 5.1.1 | 创建点 | 67 |
| 5.1.2 | 创建直线 | 68 |
| 5.1.3 | 创建圆、圆弧和椭圆 | 68 |
| 5.1.4 | 创建多线 | 69 |
| 5.1.5 | 创建和编辑多段线 | 70 |
| 5.1.6 | 创建样条曲线 | 72 |
| 5.1.7 | 创建实体填充 | 73 |
| 5.1.8 | 面域 | 74 |
| 5.1.9 | 图案填充 | 75 |
| 5.1.10 | 表格 | 79 |
| 5.2 | 图形对象的公共属性和通用的编辑方法 | 80 |
| 5.2.1 | 图形对象的公共属性 | 80 |
| 5.2.2 | 复制 | 81 |
| 5.2.3 | 移动 | 81 |
| 5.2.4 | 旋转 | 82 |
| 5.2.5 | 偏移 | 83 |
| 5.2.6 | 阵列 | 83 |
| 5.2.7 | 删除 | 85 |
| 5.2.8 | 延伸和修剪 | 85 |
| 5.2.9 | 分解 | 86 |
| 5.2.10 | 比例缩放 | 87 |
| 5.2.11 | 镜像 | 88 |

| | | |
|--------------|---|------------|
| 5.2.12 | 使用命名对象 | 89 |
| 5.3 | 文字 | 89 |
| 5.3.1 | 文字样式 | 89 |
| 5.3.2 | 设置字体 | 90 |
| 5.3.3 | Unicode 字符和控制码 | 92 |
| 5.3.4 | 使用单行文字 | 93 |
| 5.3.5 | 使用多行文字 | 95 |
| 5.4 | 块和外部引用 | 97 |
| 5.4.1 | 块操作 | 97 |
| 5.4.2 | 块属性 | 98 |
| 5.4.3 | 外部引用 | 99 |
| 5.5 | 标注和公差 | 101 |
| 5.5.1 | 标注基础知识 | 101 |
| 5.5.2 | 创建标注 | 101 |
| 5.5.3 | 编辑标注 | 105 |
| 5.5.4 | 标注样式 | 107 |
| 5.5.5 | 创建及编辑引线标注 | 110 |
| 5.5.6 | 创建及编辑形位公差 | 112 |
| 5.6 | 图形对象创建与编辑的综合示例 | 114 |
| 5.6.1 | 演示图形对象创建与编辑的综合示例 | 114 |
| 5.6.2 | 生成带键槽的轴剖面综合示例 | 116 |
| 第 6 章 | 三维实体对象的创建与编辑 | 122 |
| 6.1 | 概述 | 122 |
| 6.1.1 | 三维模型 | 122 |
| 6.1.2 | 三维坐标系 | 122 |
| 6.1.3 | 定义用户坐标系 | 124 |
| 6.2 | 创建三维实体对象 | 125 |
| 6.2.1 | 根据基本实体形创建三维实体 | 125 |
| 6.2.2 | 使用 AddExtrudeSolid 方法创建三维实体 | 126 |
| 6.2.3 | 使用 AddExtrudedSolidAlongPath 方法创建三维实体 | 127 |
| 6.2.4 | 利用 AddRevolvedSolid 方法创建三维实体 | 129 |
| 6.3 | 在三维空间中编辑 | 130 |
| 6.3.1 | 在三维空间中旋转对象 | 130 |
| 6.3.2 | 在三维空间中创建阵列 | 131 |
| 6.3.3 | 在三维空间中镜像对象 | 132 |
| 6.4 | 编辑三维实体 | 133 |
| 6.4.1 | 布尔运算和检查干涉 | 133 |
| 6.4.2 | 剖切三维实体和生成相交截面 | 136 |
| 6.5 | 三维实体创建与编辑的综合示例 | 138 |

| | | |
|------------|---|------------|
| 6.5.1 | 三维实体综合示例的用户界面 | 138 |
| 6.5.2 | 三维实体综合示例的源程序 | 138 |
| 第7章 | 应用程序实例 | 142 |
| 7.1 | 概述 | 142 |
| 7.1.1 | 选择或设计样板图 | 142 |
| 7.1.2 | 设计应用程序的界面和编写应用程序的代码 | 142 |
| 7.1.3 | VBA 和 VB 程序代码样例 | 143 |
| 7.2 | 齿轮零件图 | 145 |
| 7.2.1 | 窗体的用户界面和绘图结果 | 145 |
| 7.2.2 | 典型的程序段 | 146 |
| 7.3 | 轴零件图 | 148 |
| 7.3.1 | 窗体的用户界面和绘图结果 | 148 |
| 7.3.2 | 典型的程序段 | 150 |
| 7.4 | V带轮零件图和三维实体 | 152 |
| 7.4.1 | 窗体的用户界面和绘图结果 | 152 |
| 7.4.2 | 典型的程序段 | 152 |
| 第8章 | 《M-CAD 常用机械传动零件计算机辅助设计》软件包简介 | 159 |
| 8.1 | M-CAD 的功能和用途 | 159 |
| 8.2 | M-CAD 的工作环境和安装方法 | 159 |
| 8.3 | M-CAD 的使用方法 | 160 |
| 8.3.1 | 机械传动装置总体设计 | 160 |
| 8.3.2 | V带传动设计 | 161 |
| 8.3.3 | 圆柱齿轮传动设计 | 163 |
| 8.3.4 | 直齿圆锥齿轮传动设计 | 164 |
| 8.3.5 | 普通圆柱蜗杆传动设计 | 166 |
| 8.3.6 | 套筒滚子链传动设计 | 168 |
| 8.3.7 | 滚动轴承选用 | 171 |
| 8.3.8 | 齿轮减速器轴的设计 | 172 |
| 附录1 | 各章编程范例索引表 | 178 |
| 附录2 | 综合应用示例清单 | 183 |
| 附录3 | 参数化绘图应用程序清单 | 186 |
| 附录4 | 本书附带光盘的使用说明 | 187 |

第 1 章 走进 AutoCAD ActiveX/VBA

1.1 ActiveX 自动化技术在 AutoCAD 中的应用

ActiveX 是微软近年推出的一个基于 COM(Component Object Model)的技术规范。它是在 OLE(Object Linked and Embedded)基础上发展起来的新技术,其宗旨是在 Windows 系统的统一管理下协调不同的应用程序,构成复合文档。应用程序可以决定将自己哪些信息暴露出来,这个暴露自身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)。所谓 ActiveX 部件(简称“部件”),是一段可重复使用的编程代码和数据,它由利用 ActiveX Automation(自动化)技术创建的一个或多个对象所组成。由于编写客户程序比将其做成服务程序要简单得多,因此要利用已有的 ActiveX 部件作为被调用的服务程序。

Autodesk 公司采纳了微软的这一技术规范,最早在 AutoCAD R14.0 版推出了基于 ActiveX Automation 技术的开发方式。随后就有 AutoCAD R14.01 版,该版支持“前期绑定”,并且内嵌了 AutoCAD VBA。到 AutoCAD 2000/2002 和 AutoCAD 2004/2005/2006 版,这种新型的开发方式得到了进一步的改进及完善。

AutoCAD ActiveX 使用户能够从 AutoCAD 的内部或外部以编程方式来操作 AutoCAD。它是通过将 AutoCAD 对象显示到“外部世界”来做到这一点的。一旦这些对象被“暴露”,许多不同的编程语言或其他应用程序(例如 VB、C++、C#、Java、Word VBA 或 Excel VBA 等)就可以访问它们。VB 功能强大、语法简单、应用广泛,是良好的外部编程环境。AutoCAD VBA 是内嵌在 AutoCAD 内部的编程环境。

AutoCAD VBA 允许 VBA 环境与 AutoCAD 同时运行。VBA 通过 AutoCAD ActiveX Automation 接口将消息发送到 AutoCAD,并通过 ActiveX Automation 接口对 AutoCAD 进行编程控制。将 AutoCAD、ActiveX Automation 和 VBA/VB 相结合编程有三个基本要素。第一个要素是 AutoCAD 本身,它拥有丰富的封装了的 AutoCAD 图元、数据和命令的对象集。AutoCAD 是一个具有多层接口的开放架构应用程序。第二个要素是 AutoCAD ActiveX Automation 接口,它建立与 AutoCAD 对象的消息传递(通信)。第三个要素是 VBA/VB 集成编程环境 (IDE),它具有自己的对象组、关键词和常量等,能提供程序流、控制、调试和执行等功能。

在 AutoCAD 2000/2002 中,几乎对外暴露了所有对象,包括图形对象(如直线、圆弧)、样式设置对象(如线型、文本样式)、组织结构对象(如图层)、图形显示对象(如视口)以及菜单和工具栏等。到 AutoCAD 2004/2005/2006 版,又得到逐步的改进,如真彩色控制和表格等。

1.2 AutoCAD VBA 基础

1.2.1 概述

VBA(Visual Basic for Applications)与 VB(Visual Basic)一样也是一种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。

AutoCAD 2000/2002 VBA 相当于 VB5,而 AutoCAD 2004/2005/2006 VBA 相当于 VB6,VB 所支持的对象属性和方法,VBA 也支持。

AutoCAD VBA 允许其 Visual Basic 环境与 AutoCAD 同时运行,并通过 ActiveX Automation 接口提供对 AutoCAD 的编程控制。这样就把 AutoCAD、ActiveX Automation 和 VBA 紧密结合在一起,提供一个非常强大的接口。它不仅能控制 AutoCAD 对象,也能向其他应用程序发送数据或从中提取数据。

在 AutoCAD 中与 VBA 应用程序开发有关的下拉菜单如图 1-1 所示。

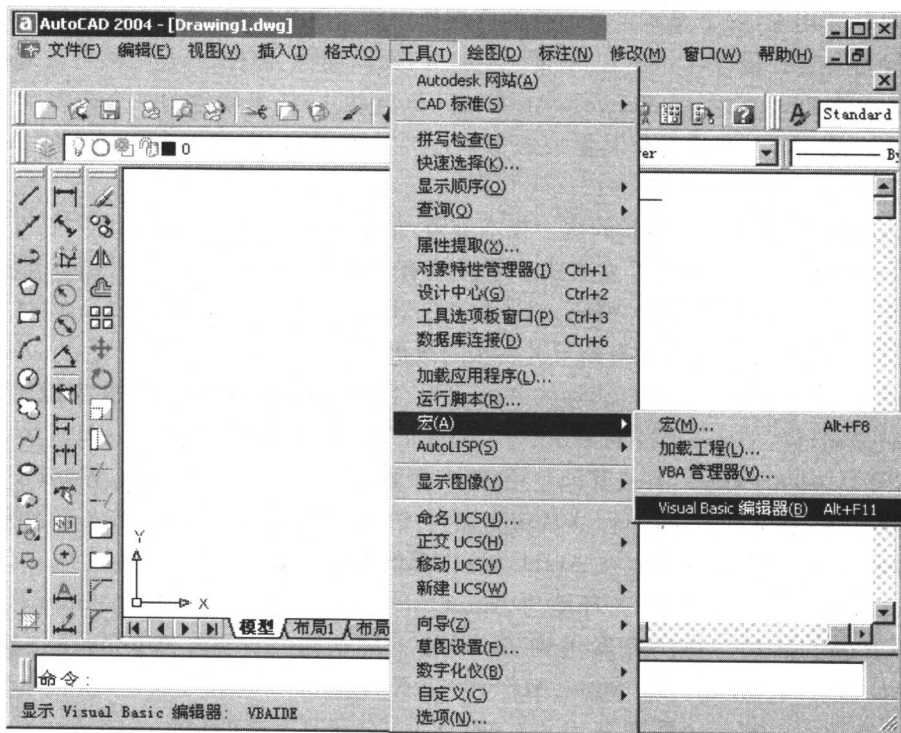


图 1-1 与 VBA 应用程序开发有关的下拉菜单

注:VBA 不支持创建可执行文件,但它提供了口令保护,以工程的方式保护工程窗体、类和模块的可见性。用户可以在 VBA IDE 中选择菜单“工具”|“ACAD Project 属性”,在“ACAD Project—工程属性”对话框的“保护”选项卡进行相应的操作。

VBA 常用的工程术语如下。

(1)内嵌工程(embedded project):内嵌在 AutoCAD 图形中的 VBA 工程,打开图形时自动加载。

(2)全局(通用)工程(global project):指独立存储在“.dvb”文件中的工程,需要先加载。

- (3)一般(规则)文档:未包含 VBA 内嵌工程的 AutoCAD 图形。
- (4)智能文档:包含一个或多个 VBA 内嵌工程的 AutoCAD 图形。
- (5)当前工程:VBA IDE 中当前选定的工程。
- (6)宏:用户可直接执行的 VBA 通用子程序或函数。每一个 VBA 工程至少包含一个宏。

VBA 工程可以包含的组件有对象、窗体、标准模块、类模块和引用。下面对这些元素分别介绍。

(1)ThisDrawing 对象:代表当前(活动)的 AutoCAD 图形(文档)。对于全局工程,它指 AutoCAD 中的当前(活动)文档;对于内嵌工程,指包含此工程的文档。一般情况下,创建一个工程后,该工程只含有一个元素——ThisDrawing 对象,其他部分要用户根据需要添加。

(2)窗体:窗体是放置其他对象的容器,是基本的构造块。用户可以通过它为应用程序创建自定义对话框。它可以包括事件过程、通用过程和变量的窗体级声明。

(3)标准模块:标准模块也称作“代码模块”或简称为“模块”。它包含用户的常用过程和函数。

(4)类模块:类模块创建具有属性和方法的用户自定义对象。

(5)引用:指能被应用程序使用的外部 ActiveX 部件或其他工程文件。

1.2.2 VBA 工程及其管理

当前 AutoCAD 进程中加载的所有 VBA 工程都可以用“VBA 管理器”对话框来管理。可使用它创建、加载、卸载、保存、内嵌或提取 VBA 工程,还可查看有哪些工程内嵌在打开的图形中(如果有的话)。可选择菜单“工具”|“宏”|“VBA 管理器”或键入命令 VBAMAN 打开 VBA 管理器,如图 1-2 所示。

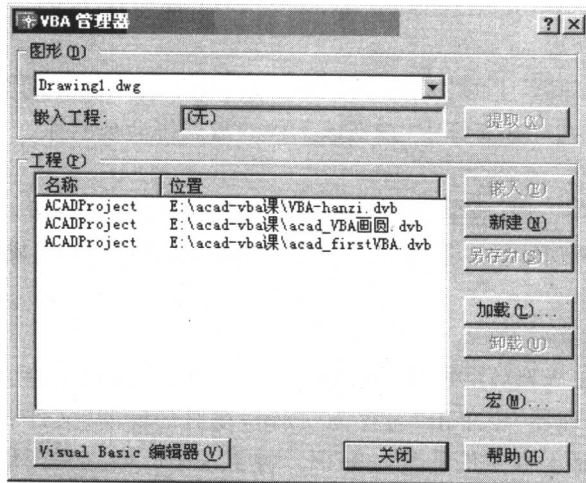


图 1-2 VBA 管理器

1. 加载工程

当需要加载已有的工程时,可应用菜单“工具”|“宏”|“VBA 管理器”打开“VBA 管理

器”对话框,单击“加载”,按钮或选择菜单“工具”|“宏(A)”|“加载工程”打开“打开 VBA 工程”对话框,如图 1-3 所示。

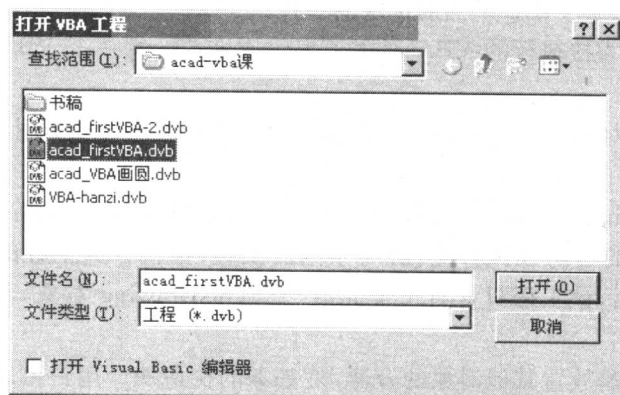


图 1-3 “打开 VBA 工程”对话框

当选取工程将其加载到 AutoCAD 图形中时,此工程引用的其他工程也会被自动加载。也可在命令提示行键入 VBALOAD 执行加载。

2. 卸载工程

在“VBA 管理器”对话框中选择要卸载的工程,单击“卸载”即可。也可在命令提示行键入 VBAUNLOAD 执行卸载。

3. 嵌入工程

将工程嵌入到 AutoCAD 图形中,是指把全局工程的副本放到图形的数据库中,此后该工程就将随着图形的打开而加载,并随着图形的关闭而卸载。注意,一个图形只能嵌入一个内嵌工程。

欲将工程嵌入到 AutoCAD 图形中,应单击“VBA 管理器”上的“嵌入”按钮。

4. 提取工程

当单击“VBA 管理器”上的“提取”按钮时,就使该工程从所在的图形数据库中删除。用户可将该工程另存到外部文件中,否则该工程的数据将会被删除。

5. 创建新工程

要创建新的 VBA 工程,应单击“VBA 管理器”上的“新建”按钮。一个新建 VBA 工程的默认名称为 ACADProject,若要变更工程的名称,应在 VBA 集成开发环境中的工程属性栏中修改。

6. 命名工程和保存工程

工程的名称是在 VBA IDE 的“属性”窗口中设定的,而保存全局工程的文件名(.dvb)应在“VBA 管理器”中单击“另存为”按钮,在对话框的“文件名”框内输入文件名(.dvb),文件名是存储工程文件的唯一标识。对于一个已嵌入到图形中的工程,在保存图形文件时将一起被保存。

1.2.3 VBA 集成开发环境

在 AutoCAD 中加载了工程后,就可以利用 VBA 集成开发环境(IDE)来编辑该工程

的窗体、程序代码和引用等元素,也可以在 VBA IDE 中调试和运行工程。

1. 打开 VBA IDE

可用以下 3 种方法打开 VBA IDE。

- (1)在命令行中键入 VBAIDE。
- (2)应用菜单“工具”|“宏(A)”|“Visual Basic 编辑器”。
- (3)直接按 Alt+F11 键。

打开的 VBA IDE 如图 1-4 所示。

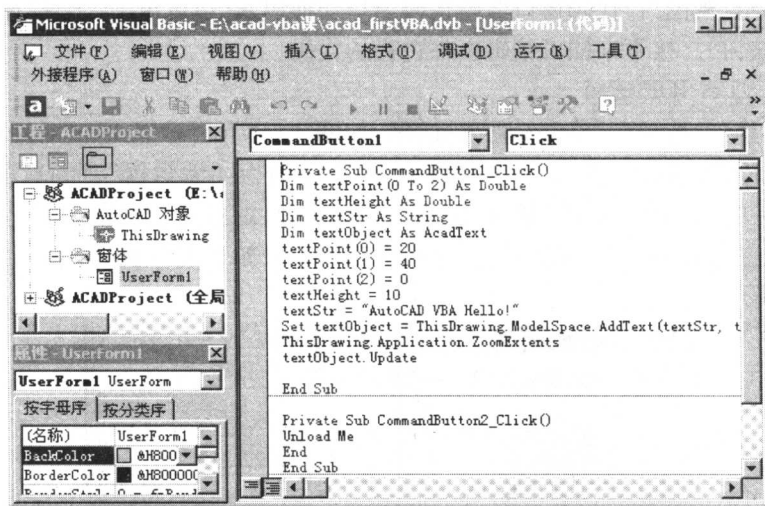


图 1-4 打开的 VBA IDE

注:可以设定在每次打开 AutoCAD 时让系统自动启动 VBA IDE,只需在 acad.arx 文件中增添一行代码 acadvba.arx 即可。

2. 工程信息

可在 VBA IDE 中的“工程”窗口查看所有已经加载的 VBA 工程,并可查看这些工程中所包含的程序代码、类别、窗体及与工程相关的文件等。

3. 工程的组成

每个工程都是由不同元素构成的。它可以包含 AutoCAD 图形、窗体、标准模块、类模块和引用。

4. 添加新元素

用户可以在工程中添加新元素,其方法是从“插入”菜单选择相应的选项进行添加。当新的元素添加到工程中以后,就会以默认的属性显示在“工程”窗口中。

5. 导入已存在的元素

用户可以将已存在的元素导入到当前的工程中。可以导入的文件包括:窗体文件(.frm)、标准模块文件(.bas)、类模块文件(.cls)。当导入已存在的元素时,新添加到当前工程中的是原文件的副本,对导入元素的修改不会影响到原始文件。

导入元素的方法是在菜单“文件”中选“导入文件”项,打开“导入文件”对话框,选择需要的文件即可。导入后,即显示在“工程”窗口中。

6. 编辑元素

(1) 使用窗体编辑器

在窗体编辑器中,用户可以进行应用程序界面的设计。可通过“工程”窗口中选择要编辑的窗体,再选择菜单“视图”|“对象窗口”,将窗口编辑器切换出来(也可单击“工程”窗口左上角的“查看对象”按钮)。接下来即可编辑窗体及其上面控件的属性。

(2) 使用代码编辑器

在代码编辑器中,用户可以进行代码的编辑。可通过“工程”窗口中选择要编辑的模块,再选择菜单“视图”|“代码窗口”,将代码窗口切换出来(也可单击“工程”窗口左上角的“查看代码”按钮)。接下来即可编辑事件过程的代码。

7. 设置 VBA IDE 选项

选择“工具”|“选项”打开“选项”对话框,如图 1-5 所示。应用该对话框可以设置 VBA IDE 的环境特征。它包括 4 个选项卡,内容分别为:“编辑器”、“编辑器格式”、“通用”和“可连接的”。比如,利用“通用”选项卡,用户可以设置在设计状态下是否显示窗体上的网格,设置网格大小以及控件是否与网格对齐。又如利用“编辑器”选项卡设定是否强制要求变量声明。

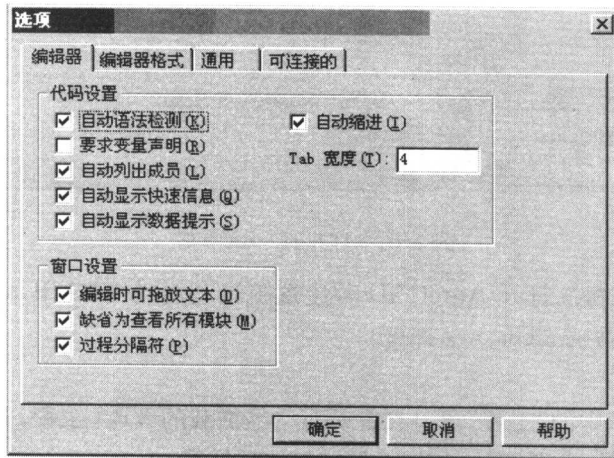


图 1-5 VBA IDE 的“选项”对话框

1.2.4 宏及其管理

1. “宏”对话框

从 AutoCAD 的菜单“工具”|“宏”|“宏”或在 AutoCAD 命令提示行中输入 VBARUN,即可打开“宏”对话框。“宏”对话框显示了图形文件或工程文件中包含的所有宏。通过“宏”对话框,用户可以运行、编辑、删除、创建宏及设置 VBA 工程选项。用户可以通过“宏位置”的下拉列表改变宏的选用范围。打开的“宏”对话框如图 1-6 所示。

如果在 VBARUN 命令前加上_,则可直接在命令提示行输入要运行的宏并运行该宏。

2. 运行宏

运行宏就是在当前(活动)的 AutoCAD 图形中执行宏的程序代码。当前(活动)的图

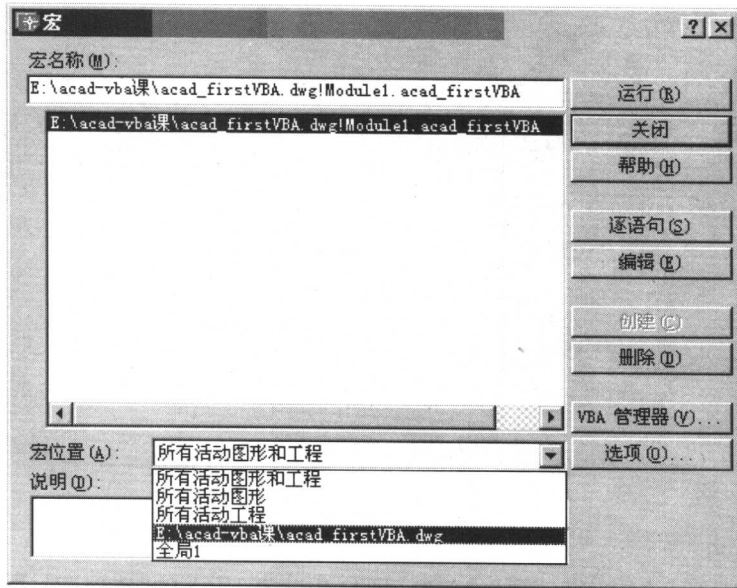


图 1-6 “宏”对话框

形指的是宏开始执行时焦点所在的打开图形。对于全局工程中的宏，ThisDrawing 对象的所有 VBA 引用都将指向当前(活动)的图形。对于内嵌工程中的宏，ThisDrawing 对象总是指内嵌了宏的图形。

运行宏时，首先要打开“宏”对话框，选定要运行的宏，然后单击“运行”按钮即可。

允许在 AutoCAD 命令行键入 VBASTMT 直接运行 VBA 语句。VBA 语句中可以包括关键字、操作符、变量、常量、表达式等内容。语句一般只占一行，若需要多句，可用冒号(:)隔开。

3. 编辑和调试宏

若需修改和编辑已有的宏，需要在 VBA IDE 中进行操作。

宏的调试是按单步运行的方式。其方法是：先在“宏”对话框中选择要调试的宏，然后选择按钮“逐语句”。

4. 创建新宏

利用“宏”对话框可以创建新的宏，步骤如下。

(1)首先在“宏名称(M)”文本框内键入宏的名称，例如键入 acad_firstVBA。

(2)在“宏位置(A)”下拉列表中选择要建立宏的工程，例如选取 acad_firstVBA.dwg，自动生成的宏名称如图 1-6 所示。

若未选取“宏位置”，当运行“创建”时，会自动弹出对话框提示操作者确认。

(3)单击“创建”按钮，打开 VBA IDE。在 Sub acad_firstVBA()和 End Sub 语句之间键入程序代码，如图 1-7 的 VBA IDE 程序代码栏中所示。

5. 删除宏

利用“宏”对话框可以删除宏，步骤如下。

(1)选择要删除的宏的名称。

(2)单击“删除”按钮。