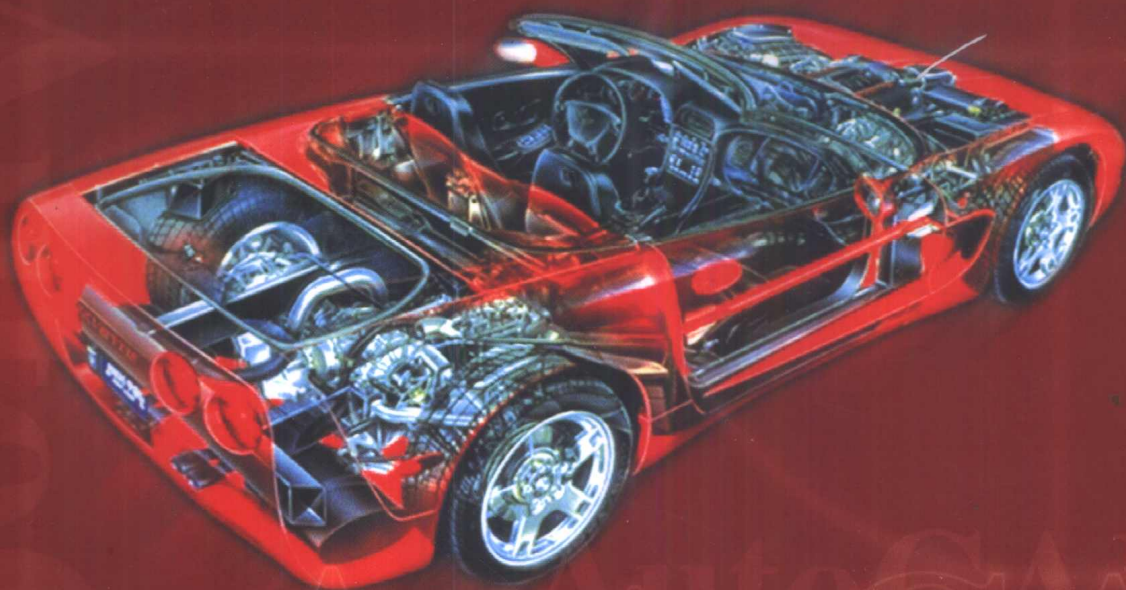




设计大师 AutoCAD 2002 高级使用篇

孙海林 陆帅华 赵海峰 编著



设计大师 AutoCAD 2002

高级使用篇

孙海林 陆帅华 赵海峰 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

AutoCAD 已经广泛应用于建筑、机械、电子、航天、冶金、地质等工程设计领域。AutoCAD 2002 不仅向用户提供了丰富的绘图命令、强有力的编辑功能和友好的用户界面,而且提供了各种各样的开发工具。本书系统地介绍了对 AutoCAD 2002 的定制及 Visual LISP 的开发技术。主要内容有:定义新的 AutoCAD 命令;建立专用的线型、图案和符号库;定制带有专业特色的各种菜单、工具栏和对话框;图形数据库;AutoLISP 编程和在 Visual LISP 环境下的 AutoLISP 编程;VBA 编程。在本书附录中还列出了 AutoCAD 系统变量、AutoLISP 错误代码供读者参考。

本书既适合于从事 CAD 工作的专业人员使用,也适合广大高校师生学习参考。

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:设计大师 AutoCAD 2002 高级使用篇

作 者:孙海林 陆帅华 赵海峰 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>

责任编辑:欧振旭

印 刷 者:北京市清华园胶印厂

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 **印张:**24.75 **字数:**565 千字

版 次:2002 年 3 月第 1 版 2002 年 3 月第 1 次印刷

书 号:ISBN 7-302-05151-8/TP·3018

印 数:0001~5000

定 价:32.00 元

前 言

AutoCAD 是美国 AutoDesk 公司推出的通用计算机绘图和设计软件包。其版本更新的速度令人目不暇接, AutoCAD 2002 的推出更显示出其强大的功能和完善的结构。

AutoCAD 的成功在于其结构的充分开放性和可开发性。AutoCAD 不仅向用户提供了丰富的绘图命令、强有力的编辑器和友好的用户界面, 而且向用户提供了各种各样的开发工具。这些开发工具包括菜单、图案、线型、字体、SCR 命令文件及 Visual LISP、ActiveX、ObjectARX 等。

AutoCAD 具有丰富的绘制和编辑图形的功能, 然而仅将其作为取代传统手工绘图的工具是远远不够的, 应该对其进行二次开发, 扩充现有功能, 增加一些具有一定智能化、参数化的命令, 将琐碎的画线、标注等操作由程序处理, 这样才能使设计人员将主要精力用于产品的结构设计上, 发挥设计人员的创造性, 这也是 CAD 的初衷。

AutoCAD 虽然功能强大, 但它只是一个通用的绘图平台。如果不结合用户二次开发的具有专业特点的软件, 就无法获得高效率。多年来, AutoLISP 一直是定义 AutoCAD 的标准, 从 AutoCAD R14 开始, 加入了新一代开发工具——Visual LISP。Visual LISP 是一种嵌入在 AutoCAD 内部的 LISP 语言, 它是 LISP 语言与 AutoCAD 有机结合的产物, Visual LISP 既能完成通常的科学计算和数据处理, 又具备一般高级语言所没有的强大的图形处理能力, 所以 Visual LISP 语言是开发 AutoCAD 最有力的工具。

本书从 AutoCAD 2002 的基本菜单与命令入手, 逐步上升到其高级应用——Script 命令组、幻灯片与形文件, 线型、剖面填充, 图形数据库, 接着重点讲解 Visual LISP 集成开发环境、函数、ActiveX 对象、反应器的使用以及程序的编辑、调试、编译和维护, 最后简要讲述了 VBA 编程。

软件开发是一项负责而辛苦的工作, 一旦开发成功, 所获得的回报不仅是提高工作效率, 更多的是成功后的喜悦。希望本书能够给读者提供帮助, 让大家不仅能学会 AutoCAD 2002 的使用, 而且能结合自己的专业二次开发出高效率的软件。

鉴于作者水平有限, 错误疏漏之处恳请广大读者批评指正。

作者

2001 年 10 月于清华园

目 录

第 1 章 定制基础.....	1
1.1 库索引路径.....	1
1.2 主要支持文件.....	3
1.3 系统目录结构.....	4
1.4 小结	5
第 2 章 定制命令.....	6
2.1 程序参数文件.....	6
2.2 定义命令.....	6
2.2.1 定义外部命令.....	6
2.2.2 定义命令别名.....	9
2.3 REINIT 命令	9
2.4 小结	10
第 3 章 定制脚本文件和幻灯片文件.....	11
3.1 定制脚本文件.....	11
3.1.1 编写脚本文件.....	12
3.1.2 调用脚本文件.....	14
3.1.3 脚本文件设计高级应用	16
3.2 定制幻灯片.....	20
3.2.1 编写幻灯片文件.....	21
3.2.2 观看幻灯片.....	23
3.2.3 建立幻灯片库.....	25
3.2.4 观看幻灯片库中的幻灯片	30
3.2.5 用脚本放映幻灯片	30
3.3 PostScript 支持	35
3.3.1 输出 PostScript 图像	36
3.3.2 输入 PostScript 图像	39
3.4 小结	42
第 4 章 定制形和字体.....	43
4.1 定制形	43
4.1.1 形的定义格式.....	43
4.1.2 矢量长度和方向代码.....	44

4.1.3	编译形文件.....	45
4.1.4	加载形文件.....	46
4.1.5	插入形	46
4.1.6	特殊代码.....	47
4.2	定制大字体.....	53
4.2.1	普通字体说明.....	53
4.2.2	大字体定义.....	54
4.2.3	扩展大字体定义.....	55
4.2.4	使用大字体.....	59
4.2.5	大字体的其他用法.....	60
4.3	小结	61
第 5 章	定制线型和填充图案.....	62
5.1	定制基本线型.....	62
5.1.1	基本线型文件的定义格式.....	62
5.1.2	创建基本线型.....	64
5.1.3	创建线型文件.....	66
5.1.4	线型加载.....	66
5.2	定制复杂线型.....	67
5.2.1	复杂线型的定义格式.....	67
5.2.2	创建复杂线型.....	69
5.2.3	复杂线型加载.....	69
5.3	填充图案.....	72
5.3.1	填充图案的定义格式.....	72
5.3.2	创建填充图案.....	74
5.3.3	加载填充图案.....	78
5.4	小结	80
第 6 章	定制菜单.....	81
6.1	管理菜单文件.....	81
6.1.1	类型和结构.....	81
6.1.2	调用过程.....	86
6.2	定制下拉菜单.....	89
6.2.1	初步定制下拉菜单.....	90
6.2.2	下拉菜单的基本结构.....	93
6.2.3	标签的基本语法.....	95
6.2.4	菜单宏 (Menu Macro) 设计的基本语法	96
6.2.5	编写简单的下拉菜单.....	98

6.2.6	为菜单项增加快捷键和帮助信息	100
6.2.7	增加状态行帮助文本	104
6.2.8	菜单文件汉化	105
6.2.9	编写自己的下拉菜单	106
6.3	图像菜单	114
6.3.1	图像菜单的格式	115
6.3.2	图像菜单的调用	118
6.3.3	定制图像菜单	118
6.3.4	图像菜单的其他问题	122
6.4	屏幕菜单	122
6.4.1	显示屏幕菜单	122
6.4.2	屏幕菜单的格式	124
6.4.3	定制简单的屏幕菜单	127
6.4.4	定制复杂屏幕菜单	129
6.5	图形输入板菜单	132
6.5.1	认识输入板菜单	132
6.5.2	输入板菜单格式	135
6.5.3	输入板菜单编写技巧	138
6.5.4	定制简单输入板菜单	138
6.5.5	定制复杂输入板菜单	141
6.6	定制工具栏	143
6.6.1	显示隐藏工具栏	144
6.6.2	创建工具栏	146
6.6.3	编辑工具栏	147
6.6.4	创建具有弹出式图标的定制工具栏	149
6.6.5	菜单文件中的工具栏	150
6.7	小结	154
第 7 章	DXF 文件	156
7.1	DXF 文件的输入输出	156
7.1.1	DXF 文件输出	156
7.1.2	DXF 文件输入	157
7.2	DXF 文件格式	158
7.2.1	文件结构	158
7.2.2	组码和组值	159
7.3	表 (Tables) 部分	162
7.3.1	注册应用名表 (APPID)	162
7.3.2	块记录表 (BLOCK_RECORD)	163

7.3.3	尺寸标注字体样式表(DIMSTYLE)	163
7.3.4	层表(LAYER)	166
7.3.5	线型表(LTYPE)	167
7.3.6	字体样式表(STYLE)	168
7.3.7	用户坐标系统表(UCS)	169
7.3.8	视图表(VIEW)	170
7.3.9	视口表 (VPORT)	171
7.4	小结	173
第 8 章	AutoLISP 编程	174
8.1	编程基础	174
8.1.1	编程环境	174
8.1.2	程序特点	174
8.2	基本函数介绍	175
8.2.1	数值计算函数	175
8.2.2	逻辑计算函数	180
8.2.3	循环函数	181
8.2.4	转换函数	183
8.2.5	自定义函数	186
8.2.6	其他函数	190
8.3	编辑和调试	197
8.3.1	调试方法	197
8.3.2	常见错误分析	202
8.4	管理图形对象	202
8.4.1	处理选择集	202
8.4.2	处理图形对象	212
8.5	数据处理	220
8.5.1	类型转换	220
8.5.2	显示控制	221
8.5.3	文件处理	227
8.5.4	设备访问及控制	237
8.6	小结	242
第 9 章	Visual LISP 编程	243
9.1	编程基础	243
9.1.1	关于 Visual LISP	243
9.1.2	启动 Visual LISP	244
9.1.3	退出 Visual LISP	244

9.2 编写程序.....	245
9.2.1 合理组织.....	245
9.2.2 使用控制台窗口.....	245
9.2.3 使用文字编辑器.....	247
9.2.4 使用控制台和编辑器的编码辅助工具	249
9.2.5 用 Visual LISP 设置代码格式.....	252
9.2.6 检查语法错误.....	253
9.3 调试程序.....	256
9.3.1 调试功能简介.....	256
9.3.2 通过例子学习调试程序.....	257
9.3.3 使用调试功能.....	260
9.4 编译应用程序.....	264
9.4.1 编译和链接程序.....	264
9.4.2 创建应用程序模块.....	268
9.4.3 针对多文档环境的设计方法.....	274
9.5 维护应用程序.....	279
9.5.1 管理多个 LISP 文件.....	279
9.5.2 操作工程.....	281
9.5.3 优化应用程序代码.....	283
9.6 ActiveX 对象.....	286
9.6.1 在 AutoLISP 中使用 ActiveX 对象	286
9.6.2 理解 AutoCAD 对象模型	287
9.6.3 访问 AutoCAD 对象	287
9.6.4 在 Visual LISP 函数中使用 ActiveX 方法	293
9.6.5 使用 ActiveX 与其他应用程序交互.....	297
9.7 事件反应器.....	303
9.7.1 反应器概述.....	303
9.7.2 反应器类型和事件.....	304
9.7.3 定义回调函数.....	306
9.7.4 创建反应器.....	308
9.7.5 在多重名称空间中使用反应器	310
9.7.6 查询、修改和删除反应器	311
9.7.7 临时反应器和永久反应器	314
9.7.8 反应器使用规则.....	315
9.8 小结	316
第 10 章 VBA 编程.....	317
10.1 VBA 编程环境	317

10.2 编写 VBA 代码	321
10.2.1 VBA 语法	321
10.2.2 使用编辑器	336
10.3 调试和编译	344
10.3.1 VBA 程序的运行	344
10.3.2 VBA 与主程序之间的切换	348
10.4 小结	349
附录 A AutoCAD 系统变量	350
附录 B AutoLISP 错误代码	380

第 1 章 定制基础

知识要点:

- ◇ CAD 命令搜索过程
- ◇ CAD 系统支持文件
- ◇ CAD 系统目录结构

本章主要介绍自定义 CAD 的基础内容,包括 CAD 命令搜索过程、CAD 系统支持文件、CAD 系统目录结构。通过本章的学习,读者将对 CAD 的系统结构有一个全面的了解,为以后学习定制菜单、字体、工具栏、填充图案等内容打下良好的基础。

1.1 库索引路径

AutoCAD 按照库路径中指定的次序搜索支持文件,如下所示:

- 当前目录。
- 包含当前图形文件的目录。
- OPTIONS 中指定搜索路径中所列出的目录。
- 包含 AutoCAD 程序文件的目录。

上述各目录可能相同,这取决于当前环境。

如果某个文件不在搜索路径中,则必须指定其路径名和文件名,AutoCAD 才能找到该文件。例如,如果要将不在库搜索路径中的图形 part5.dwg 插入到当前图形中,则必须指定其完整的路径名:

命令: insert

输入块名或[?]: /files2/olddwgs/part5

如果该图形确实位于指定的位置,AutoCAD 将和通常一样提示完成 INSERT 命令。

选择 Tools|Options 命令,弹出 Options 对话框,如图 1-1 所示。选择 Files 选单,设置 AutoCAD 查找支持文件的搜索路径。这些支持文件包括字体、图形、线型和填充图案等。Working Support File Search Path (工作支持文件搜索路径)列出的是支持文件搜索路径中的路径,这些路径是有效的并且位于当前系统目录结构中(包括系统网络映射)。适当设置这些选项可显著提高 AutoCAD 加载文件的性能。

在 Files 选单中可以指定临时文件的位置。AutoCAD 运行期间在磁盘上创建临时文件,结束运行后将其删除。AutoCAD 的临时目录是 Microsoft Windows 使用的临时目录。如果要从一个写保护的目录运行 AutoCAD (例如,从一个网络驱动器或光盘驱动器上),应为临时文件指定其他位置。所指定的目录不能是写保护的,并且该目录所在的磁盘必须拥有

足够的磁盘空间供临时文件使用。

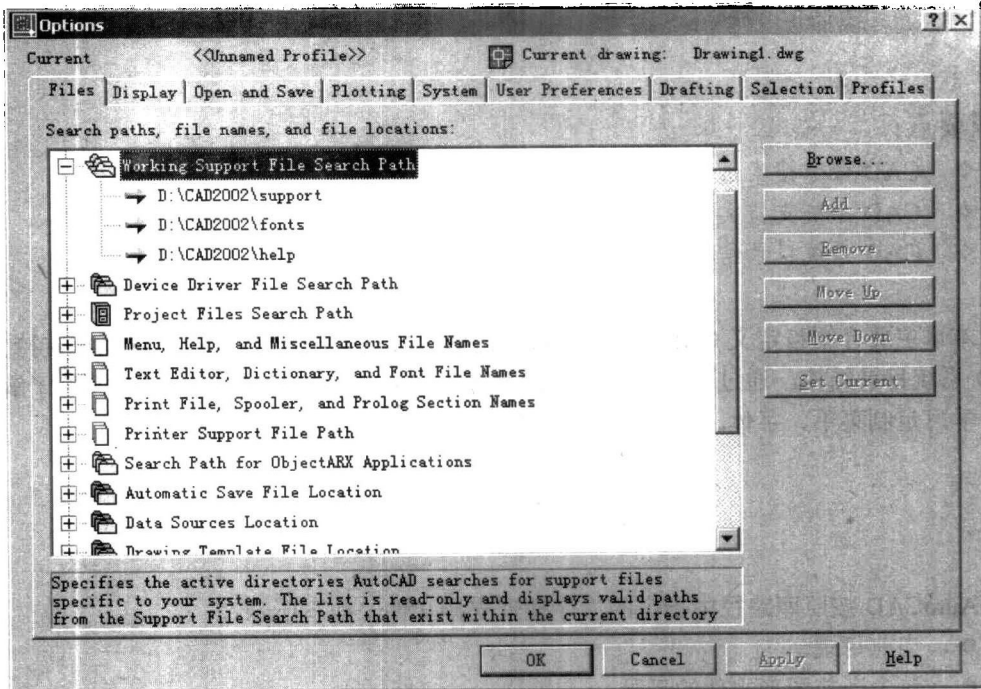


图 1-1 选项对话框

如果要使用自定义菜单，在 **Menu Help and Miscellaneous Files Name**(菜单、帮助和其他文件名称)中指定。默认的菜单是 **ACAD.MNU**。

“File”选单中的其他搜索路径、文件名和文件位置包括：

- 设备驱动程序文件搜索路径。
- 工程文件搜索路径。
- 文字编辑器、词典和字体文件名称。
- 打印文件、后台打印文件和前导部分名称（可以自定义 acad.psf 文件中的前导部分，这样，使用 PSOUT 命令将覆盖 AutoCAD PostScript 代码）。
- 打印机支持文件路径。
- ObjectARX 应用程序搜索路径。
- 自动保存文件。
- 数据源位置。
- 样板图形文件位置。
- 日志文件位置。
- 临时图形文件位置。
- 临时外部参照文件位置。
- 纹理贴图搜索路径。

命令搜索过程

AutoCAD 要对所输入命令进行一系列有效性检查。一个命令可以是内部命令、系统变量、acad.pgp 文件中定义的外部命令或命令别名（在下一节中描述），或是用户定义的 AutoLISP 命令，也可以由 ObjectARX 应用程序或设备驱动程序命令来定义。另外，命令可以在命令行中键入或者从适当的菜单中选择，也可以从脚本文件中或者由 AutoLISP 或 ObjectARX 应用程序来输入。

AutoCAD 验证命令名称的搜索次序如下：

（1）如果输入空响应（按 Space 键或 Enter 键），则 AutoCAD 调用上次执行的命令。HELP 是默认命令。

（2）AutoCAD 先在内置命令表中查找该命令名。如果该命令在内置命令表中，并且没有前缀句号（.），AutoCAD 将接着在未定义命令表中查找该命令。如果该命令未被定义，则继续搜索；否则将执行该命令，除非有其他原因使命令不能执行（例如，不允许透明地运行或在“透视”模式下运行命令）。

（3）AutoCAD 检查该命令名称是否由设备驱动程序定义，然后检查命令名是否由显示驱动程序定义。

（4）AutoCAD 在程序参数文件（acad.pgp）定义的外部命令中查找该命令名称。如果该命令名称对应一个已经定义的外部命令，则运行该命令，并结束搜索。

（5）AutoCAD 在由 AutoLISP 或 ObjectARX 应用程序定义的命令列表中查找该命令名称。自动加载的命令在此时加载。

（6）AutoCAD 在系统变量列表中查找该命令名称。如果找到该名称，AutoCAD 将输入作为变量名，执行 SETVAR 命令。

（7）如果该命令名称对应程序参数文件中定义的命令别名，AutoCAD 用该命令的全名在内置命令列表中重新搜索。

（8）如果前面所有步骤都失败了，则搜索终止，并显示非法命令的警告信息。

1.2 主要支持文件

本节主要讲述 AutoCAD 2002 安装目录下和子目录下的一些主要文件的功能，如表 1-1 所示。

表 1-1 主要支持文件及其功能

目 录 名 称	文 件 名 称	文 件 功 能
CAD2002 根目录	ACAD.EXE	AUTOCAD 可执行程序
	ACAD.DCL	AUTOCAD 对话框文件
CAD2002\SUPPORT 目录	ACAD.MNL	标准菜单使用的 AUTOLISP 文件
	ACAD.MNS	标准 AutoCAD 菜单的源文件

续表

目 录 名 称	文 件 名 称	文 件 功 能
CAD2002\SUPPORT 目录	ACAD.MNU	AUTOCAD 的标准菜单文件
	ACAD.PAT	AUTOCAD 标准的剖面线图案
	ACAD.PGP	AUTOCAD 程序参数文件
	ACAD.PSF	AutoCAD PostScript 支持文件, PSOUT 和 PSFILL 命令的主要支持文件
	ACAD.LIN	AUTOCAD 的标准线型文件
	ACADISO.LIN	标准 AutoCAD ISO 线型库文件
	ACADISO.PAT	标准 AutoCAD ISO 填充图案库文件
	ACAD.UNT	AutoCAD 单位定义文件。包含进行单位换算所需的数据
	ACAD.SLB	SLIDE 的库文件
	SLIDELIB.EXE	幻灯片库的编译程序
	*.LSP	AUTOLISP 文件
	*.DCL	对话框的控制文件
	*.DWG	图形文件
	*.PS	AUTOCAD 的 Postscript 控制文件
CAD2002\DRV 目录	*.HDI	AUTOCAD 的驱动程序
CAD2002\HELP 目录	*.CHM	已经编译好的帮助文件
	*.HLP	编译后的 WinHelp 文件
	*.CNT	目录文件
	*.GID	WinHelp 配置文件。由 winhlp32 可执行文件创建的二进制文件。包含帮助文件的相关信息, 包括目录文件链接、所有涉及的帮助文件名、关键词和文件位置
CAD2002\FONTS 目录	*.SHX	编译后的文本字体文件

1.3 系统目录结构

本节主要介绍一些简化文件操作和改进操作系统目录结构的方法。

建议用户将自己定义的文件(例如 AutoLISP 应用程序和菜单文件)与 AutoCAD 的程序和支持文件分开存放。这样更容易查找可能的冲突, 并可以单独升级某个应用程序, 而不影响其他应用程序。

下面用图 1-2 所示的目录结构样例来说明组织目录和文件的方法。

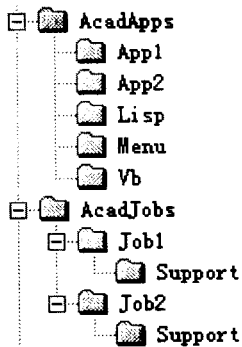


图 1-2 目录结构样例

可以为自定义的 AutoLISP 和 Visual Basic 应用程序、菜单文件和其他第三方应用程序创建一个目录（例如样例中的/AcadApps）。如果需要为相互独立的工作文件维护多个图形文件目录，则可以创建一个主目录（例如/AcadJobs），然后在其下为每项工作创建一个子目录。

1.4 小 结

本章主要介绍了定制 CAD 的基础内容——库索引路径、支持文件、目录结构，使用户对 CAD 的系统结构有一个简单的了解，接下来的几章将详细地阐述如何定制菜单、字体、工具栏、填充图案等内容。

第 2 章 定制命令

知识要点:

- ◇ 程序参数文件
- ◇ 使用 windows 系统命令
- ◇ 使用自定义命令
- ◇ 定义命令别名

本章主要介绍程序参数文件 ACAD.PGP 的基本组成,然后通过实例学习使用 windows 系统命令、使用自定义命令、定义命令别名。通过本章的学习,读者将对定制命令有一个清楚的了解,并能灵活的应用,提高 CAD 画图的工作效率。

2.1 程序参数文件

AutoCAD 软件带有程序参数文件 ACAD.PGP,该文件是 ASCII 文本文件,用于保存命令定义,可将此文件当作自定义 AutoCAD 命令的列表。当用户输入不常用的命令时,AutoCAD 在 ACAD.PGP 中查找该命令。此文件分成两个部分:第 1 部分定义操作系统命令,第 2 部分定义 AutoCAD 命令别名。当安装 AutoCAD 时,这个文件自动被拷贝到硬盘驱动器的 AutoCAD2002\SUPPORT 子目录下。

ACAD.PGP 文件允许用户从图形编辑器中直接使用操作系统命令。例如,若想要删除一个文件,只需在命令提示中输入 DEL (Command: DEL),然后 AutoCAD 将提示用户输入需要删除的文件名。

ACAD.PGP 文件也包括一些较常用的 AutoCAD 命令的命令别名。例如,LINE 命令的命令别名是 L。如果在命令提示中输入 L (Command: L),AutoCAD 将把它当成 LINE 命令。

ACAD.PGP 文件的注释可以包含任何数目的注释行,并且可以出现在文件中的任何地方。每一个注释行必须以分号 (;) 开头(这是注释行的标记)。任何以分号开头的句子在执行时都被忽略。可以使用注释行给出文件的一些相关信息,这样便于其他的 AutoCAD 用户理解、编辑或修改文件。

2.2 定义命令

2.2.1 定义外部命令

在 AutoCAD 运行时,用户可调用其他程序或实用工具,例如:

- Windows 系统命令和实用工具, 例如 start、type、dir 或 copy。
- 应用程序, 例如文本编辑器或字处理器。
- 数据库管理程序、电子表格和通讯程序。
- 用户提供的程序, 例如批处理文件或 Visual Basic 应用程序。

定义外部命令时, 需要在 AutoCAD 命令提示中指定使用的命令名称, 以及传递给操作系统的可执行命令字符串。外部命令部分中的每行都包括以逗号分隔的 5 个字段, 如下所示:

command,[executable],flags[,[*]prompt[,return_code]]

(1) command

在命令提示中输入的命令。如果此名称是 AutoCAD 内部的命令名称, 则该命令将被忽略。这个名称不区分大小写。

(2) executable

输入命令名时传递给操作系统的固定字符串, 可以是能在操作系统提示中执行的任何命令, 可包含开关或参数。该字符串是否区分大小写由运行的应用程序决定。

(3) flags

必要的位编码参数, 将这些整数值按任意组合相加以得到所需的结果。

0 = 启动应用程序并等待结束。

1 = 不等应用程序结束。

2 = 最小化运行应用程序。

4 = “隐藏”运行应用程序。

8 = 将参数字符串括在引号中。

位值 2 和 4 是互斥的, 如果同时指定了这两个位值, 那么只使用 2 位。应尽量避免不带值 1, 值 2 或值 4, 因为 AutoCAD 在应用程序运行时不能使用。

位值 8 使 del 这类命令可以正确处理包含空格的文件名。注意这将导致不能向这些命令传递用空格分隔的文件名列表。如果想使用多文件支持, 则不要使用位值 8。

(4) prompt

此字段可选。它指定显示在 AutoCAD 命令行中的提示。此提示的响应被添加到 executable 字段提供的字符串后。如果 prompt 字段的第一个字符是星号 (*), 则响应可以包含空格, 用户必须按 Enter 键结束响应; 否则, 响应可以用 Space 键或 Enter 键结束响应。如果未指定 prompt, 则不需要输入。但如果要提供返回码或提示行需要后续空格, 则必须添加逗号。

(5) return_code

可选的位编码参数。可将这些整数值按任意组合相加以得到所需的结果。例如, 如果需要值 1 和值 2, 可用值 3 作为返回码。值的定义如下 (代码 0 和 4 对窗口环境无意义, 因此不包括在此列表中):

值 1 = 加载 DXB 文件: AutoCAD 在命令结束后将名为 \$cmd.dxb 的 DXB 文件加载到图形中。加载 DXB 文件后, \$cmd.dxb 文件将被删除。此操作产生的结果与 DXBIN 命令相同。

值 2 = 用 DXB 文件构造块定义: AutoCAD 用名为 \$cmd.dxb 的 DXB 文件创建块定义。对 prompt 字段的响应将被用作块名。此名称必须是当前图形中不存在的有效块名, 因此,