

江雪松 等编著

# AutoCAD

# 2000

## 二次开发指南



1.72

国防工业出版社

51

TP431.72

744

# AutoCAD 2000 二次开发指南

江雪松 等编著

国防工业出版社

·北京·

### 图书在版编目(CIP)数据

AutoCAD 2000 二次开发指南 / 江雪松等编著. —北京:  
国防工业出版社, 2000.9

ISBN 7-118-02246-2

I. A... II. 江... III. 计算机辅助设计-应用软件,  
AutoCAD 2000-软件开发 IV. TP391.72

中国版本图书馆 CIP 数据核字(2000)第 31740 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)

北京奥隆印刷厂印刷

新华书店经售

\*

开本 787×1092 1/16 印张 13 $\frac{3}{4}$  314 千字

2000 年 9 月第 1 版 2000 年 9 月北京第 1 次印刷

印数:1—3000 册 定价:19.00 元

---

(本书如有印装错误,我社负责调换)

# 前 言

AutoCAD 2000 是美国 Autodesk 公司应当今技术的快速发展和用户的需求推出的面向 21 世纪的 CAD 软件包,它实现了向 2W(Windows/Web)的战略转移,体现了世界 CAD 技术的发展趋势。它的推出,使得已被注入到 PC 和 Windows 系统中的先进技术力量、Internet 四海一家的信息共享机制、面向对象的智能化编程及数据体系变成了 CAD 用户期盼的便利、快捷、灵巧的设计和绘图能力,它正迅速而深刻地影响着人们从事设计和绘图的基本方式。

AutoCAD 得到广泛应用的原因除了因为它功能强大、多平台支持、可支持的外部设备广泛、易学易用外,更主要的原因是它具有良好的开放体系结构,即允许用户和其他开发商对其进行扩充和改进,能最大限度地满足不同用户的特殊需求。通过二次开发,可方便地将其改造成为符合用户要求的专业软件。

AutoCAD-2000 虽然较前一版本已大大提高了设计效率,但是由于它是作为一种通用绘图系统而设计的,这就不可避免地有其局限性。于是产生了许多二次开发商,那么是否每一家工厂、企业都需要购买在 AutoCAD 上开发的增值软件呢?不,并不需要,有时候只需要做很少一些工作,也同样可以使生产效率成倍地提高。大多数时候设计工程师都在做许多的重复工作,他们在为完成一些早已让人厌烦的绘图工作而苦恼,而新加入的设计师们又在为对设计软件的使用不熟悉而苦恼,也许这时工程师们都想到了再次购买新软件,因为他们认为自己不能做二次开发,事实并非如此,您只需要掌握了 AutoCAD 基本使用方法,有一点编程基础就可以完成。本书即是以此为目的而编写的。

本书是在总结作者进行 AutoCAD 软件系统教学及开发经验基础上编写而成的,书中的实例也大多取自作者的开发实践,内容包括:AutoCAD 命令的用户化,AutoCAD 的配置与优化,线型与图案填充的开发,形、字体和 PostScript 支持,自定义菜单开发与 DIESEL 语言,工具栏的定制以及 ActiveX Automation、Visual LISP、ObjectARX 等技术。本书在内容安排上采用循序渐进、图文并茂的方式,并附有相当数量的实例和有关程序,具有较好的可操作性。

由于编者水平有限,书中难免有挂一漏万之处,恳请读者不吝指教。

本书主要由江雪松、刘德福、陈锋、欧阳光编写,同时参加编写的人员还有:薛小香、林章庆、刘元高、李春鹤、冯曙红、袁军、林振宁、周成福、陈培、邓冰、黄强等,在此表示感谢。

编著者

# 第 1 章

## AutoCAD 定制入门

本章将讲述基本的 AutoCAD 自定义功能,初次接触自定义功能的用户应先阅读本章内容,以掌握如何在 AutoCAD 中输入操作系统命令,如何创建常用命令的别名,以及如何为自己的应用程序开发联机帮助。

本章具体包括以下内容:

- AutoCAD 基本环境介绍
- AutoCAD 命令搜索过程
- AutoCAD 中可自定义的支持文件
- 如何在 AutoCAD 中定义外部命令
- 如何在 AutoCAD 中使用命令别名
- 如何在 AutoCAD 中使用 Windows 帮助文件
- 如何在 AutoCAD 中使用 HTML 文件
- 如何使用 AutoCAD 帮助文件

AutoCAD 是作为一通用绘图系统而设计的,但各行各业都有自己的行业标准,许多单位也有自己的技术规范和企标准,每个设计工程师和绘图员更有各自独特的工作方式。因而,AutoCAD 不可能完全满足每个用户的具体要求。于是通过系统提供的开放体系结构,AutoCAD 允许用户和第三方软件开发商根据各自的需求来改进和扩充 AutoCAD 的许多功能,实现对 AutoCAD 的二次开发,称为用户化或定制(Customization)。本章将讲述基本的 AutoCAD 自定义功能,引导用户进行简单的 AutoCAD 定制。

### 1.1 概 述

AutoCAD 是开放式结构的通用绘图系统,用户可以根据需要扩展和调整 AutoCAD 的功能,二次开发商也可以提供许多独立开发的应用程序及扩展,使 AutoCAD 更加符合用户


的需要。本节将对 AutoCAD 2000 的定制及其新功能作一简要介绍。

### 1.1.1 AutoCAD 2000 定制指南的变化

新版本的 AutoCAD 2000 向开发人员提供了一种新的途径,使他们可通过更加强大和高集成度的编程接口来访问其开放式结构。因此,以前属于定制指南的某些内容现在都已独立成册,故不再纳入本书。新版本的 AutoCAD 2000 的定制指南内容变化如下:

(1) 通过新的开发环境 Visual LISP、AutoLISP 的功能大大增强,AutoDesk 公司提供的三个在线文档将帮助用户开发 AutoLISP 应用程序。Visual LISP 开发人员手册解释如何使用 Visual LISP 集成开发环境 (IDE),以及如何编译和运行 AutoLISP 应用程序。AutoLISP 参考定义并说明各个 AutoLISP 函数,并且给出了样例。AutoLISP 参考说明函数的语法和返回值。Visual LISP 教程则循序渐进地指导开发者编译和使用 Visual LISP 应用程序。

(2) 可编程对话框参考原来是定制指南的一部分,在 AutoCAD 2000 中已被并入 Visual LISP 开发人员手册。

提示:新的 AutoCAD 开发环境已提供了对 MFC 的支持, ObjectARX 2000 还提供了自己的用户界面类库,开发者可用它们来建立更加友好的用户界面,如上下文敏感菜单就是可编程对话框无法提供的。

(3) DXF 组码定义了 AutoCAD 图形中的所有对象。这些组码可以在编写 AutoLISP 和 ObjectARX 应用程序的过程中使用,以维护图形对象,也可供开发者和用户进行图形数据的输入和输出。DXF Reference 原来是定制指南的一部分,现在已作为 HTML 格式的帮助文件出现,同时还提供 PDF 文件以供打印。上述文件也可以从 AutoCAD 技术文档发布站点获得,网址为: <http://www.autodesk.com/techpubs/autocad/>。

鉴于此本书将不再讲述以上内容,而将集中向用户讲述 AutoCAD 中的基本自定义技巧,这几乎不涉及编程。这些都是通过修改或创建 ASCII 文本文件来实现的。

### 1.1.2 AutoCAD 2000 中的基本自定义特性

AutoCAD 2000 中的基本自定义特性包括:

(1) 可在 AutoCAD 中运行外部程序。例如,只要在程序参数文件 acad.pgp 中加入适当的外部命令,就可以在 AutoCAD 内部复制磁盘或删除文件。对于经常使用的 AutoCAD 命令,可以创建它们的简单缩写,即别名。例如,输入 b 就可以调用 BHATCH 命令。别名也是在程序参数文件 acad.pgp 中定义的(参见 1.3 节)。

(2) 用户可为应用程序和内部过程制作帮助系统。通过使用 AutoLISP、VBA 和 Object ARX 程序,可以从菜单、工具栏获得帮助。用户通过创建帮助文件,可以自定义 AutoCAD 命令的帮助,也可以为自己的应用程序定义帮助(参见 1.4 节)。

(3) 创建自定义点划线型、填充图案、形和文字样式。用户可以创建符合企业标准和工作需要的填充图案、形和字体(参见第 3 章、第 4 章)。

(4) 创建自定义菜单和工具栏,以修改 AutoCAD 用户界面。菜单在很大程度上决定了 AutoCAD 界面的外观,同时也决定了用户的工作效率。将最常用的命令放在菜单上最容易选取的位置,并将命令分组,合并到单个菜单项中,工作效率会显著提高。通过菜单自定义

功能,用户可以根据实际需要调整 AutoCAD 界面。菜单文件定义了选择定点设备按钮后执行的操作。同样,下拉菜单、屏幕、数字化仪和图像控件菜单的功能和外观,以及工具栏和加速键,均在菜单文件定义(参见第 5 章)。

(5) 自定义状态行。可以使用 DIESEL 字符串表达式语言和 MODEMACRO 系统变量在状态行中显示各种信息,如日期和时间、系统变量设置或其他可以通过 AutoLISP 提取的信息(参见第 6 章)。

(6) 编写脚本自动完成重复性任务。脚本是包含 AutoCAD 命令的 ASCII 文本文件,其运行方式类似于批处理文件。例如,如果需要以某种方式打印一组图形,则可以编写脚本来依次打开每个图形,打开或关闭其中不同的图层并发出 PLOT 命令,然后让它在夜间无人看管时自动运行。还可以用脚本和幻灯片创建展示会中那种自动连续播放的演示。幻灯片是不可编辑的 AutoCAD 绘图区域“快照”。在图片菜单和对话框中,幻灯片还用于概括图形信息(参见第 7 章)。


(7) 可在命令提示或 AutoLISP 及 ObjectARX 程序中重定义或禁用选定的 AutoCAD 命令。例如,重新定义某些 AutoCAD 命令以提供辅助信息和提示;或者创建图形管理系统,在其中重新定义 QUIT 命令,在退出编辑任务之前将某些信息写入日志文件(参见第 7 章)。

本书将详细讲述以上全部内容,此外本书还介绍了 AutoCAD 应用程序编程接口(API)(参见第 7 章)。AutoLISP 是 LISP 编程语言的一种特定实现方式,是 AutoCAD 的组成部分。借助 AutoLISP,可以自动执行重复性任务或创建新的 AutoCAD 命令,使 AutoCAD 更好地满足用户自己的需要。用户可以编写自己的 AutoLISP 程序,也可以使用第三方程序和共享程序。ActiveX Automation(以前称为 OLE Automation)可替代 AutoLISP 和 DCL,并更为先进。从作为 Automation 控制器的应用程序中,如 Visual Basic 或 Microsoft Office 97 应用程序,用户能够访问和控制 AutoCAD 对象。通过 ObjectARX 编程接口,可以用 Microsoft Visual C++ 编程语言自定义 AutoCAD。用户既可以运行第三方 ObjectARX 应用程序,也可以自己开发。


### 1.1.3 AutoCAD 2000 的改进和增强

在 AutoCAD 2000 中,许多 AutoCAD 的自定义特性得到了改进和增强。影响自定义特性的改进包括:

(1) 在同一个 AutoCAD 任务中可以打开多个图形。

 **注意:**要保证与早期版本中脚本的兼容,请读者将新的 SDI(单文档界面)系统变量设为 0。

(2) acad.pgp 文件中外部命令部分的格式有所变化。

 **注意:**已废弃的 memory 字段现在称为 flags,用于控制外部应用程序。

(3) 词典对象和符号表对象(包括图层、线型和文字样式)的名称最长可达 255 个字符,其中包括扩展字符。为了保证向后兼容性,EXTNAMES 系统变量仍旧关闭扩展符号名,但在 AutoCAD 的后续版本中将被删除。

(4) 可以设置是否打印图层。

(5) 可以在“特性”窗口中查看和修改单一和多个选择集的特性。“特性”窗口用简单的

界面替代了 DDMODIFY 和 DDCHPROP 对话框。如果要在命令行中修改选择集,可以使用新的 PROPERTIES 命令。

(6) Visual LISP 提供了集成开发环境。由于 Visual LISP 还具备其他功能,所以脚本的表现可能与 AutoLISP 不同。

(7) DXFOUT 和 DXFIN 命令已被废弃,但仍旧可以用脚本、AutoLISP、ActiveX Automation 和 ObjectARX 接口来获得。现在,可以使用 SAVE、SAVEAS 和 WBLOCK 命令写 DXF 文件,还可以用 OPEN 和 INSERT 命令打开 DXF 或将其插入图形文件。在后续版本中可能不再提供对 DXFOUT 和 DXFIN 命令的支持。

(8) 现在 DXF 文件也有了相关的预览图像。使用 SAVE、SAVEAS 和 WBLOCK 命令时,可以选择创建 DXF 文件的略图。在使用 OPEN 和 INSERT 命令时可以预览该图像。

(9) 菜单文件得到增强,可以为用户提供针对特定命令和特定对象的快捷菜单。

(10) 在工具栏中增加了新的控制元素。

(11) 由于在 AutoCAD 中引入了多文档设计环境 (MDE),所以引入了新的启动 AutoLISP 文件 acadoc.lsp。acad.lsp 文件现在用于应用程序启动功能,而 acadoc.lsp 文件与文档(图形)的初始化相关。


(12) 相对(全局)菜单交换现已无效。例如,下列语法现已无效:

```
(menucmd "Gmenugroup1.menuname1 = menugroup2.menuname2")
```

但其他相对菜单参照仍旧有效。

## 1.2 基本自定义实现

本指南中所描述的大多数自定义功能都需要编辑或创建 ASCII 文本文件。因此,读者应该能熟练使用一种文本编辑器(例如 Microsoft Windows 记事本)或能将文件保存为 ASCII 格式的字处理器(例如 Microsoft Word)。

 **注意:**在对 AutoCAD 做出修改(例如菜单文件)之前请预先进行备份,以便在出现错误时恢复。

### 1.2.1 AutoCAD 环境

AutoCAD 的缺省目录结构将程序和支持文件按逻辑关系分组,这种结构可按照用户的需要修改。但有些应用程序需要在特定位置查找某些文件,用户进行修改时应避免与这些应用程序的要求冲突。如果未指定驱动器和目录,AutoCAD 只在库搜索路径中查找文件。下面将分四部分讲述与 AutoCAD 系统环境相关的知识。

#### 1. AutoCAD 库搜索路径

缺省状态下 AutoCAD 按照库路径中指定的次序搜索支持文件,具体次序如下所示:

- 当前目录(这通常由快捷图标的“开始”设置决定)。
- 包含当前图形文件的目录。
- OPTIONS 中指定的搜索路径中所列出的目录。
- 包含 AutoCAD 程序文件的目录。

上述各目录也有可能相同,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,如果要将在库搜索路径中的图形 my.dwg 插入到当前图形中,则必须如下指定其完整的路径名,具体的使用方法如下:

命令: insert

输入块名或 [?]: /files2/olddwg/my

执行该命令将把 files2/olddwg/目录下的 my.dwg 插入到当前图形中。如果该图形确实位于指定的位置,AutoCAD 将像通常一样提示完成 INSERT 命令。此外如在库搜索路径和非库搜索路径下都存在 my.dwg,那么系统将优先查找库搜索路径下的文件。

## 2. AutoCAD 目录结构样例

AutoCAD 使用树状结构的目录和子目录,本节介绍一些简化文件操作和改进操作系统目录结构的方法。建议读者将自己定义的文件(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。

下面几节将用图 1.1 所示的目录结构图例来说明组织目录和文件的方法。

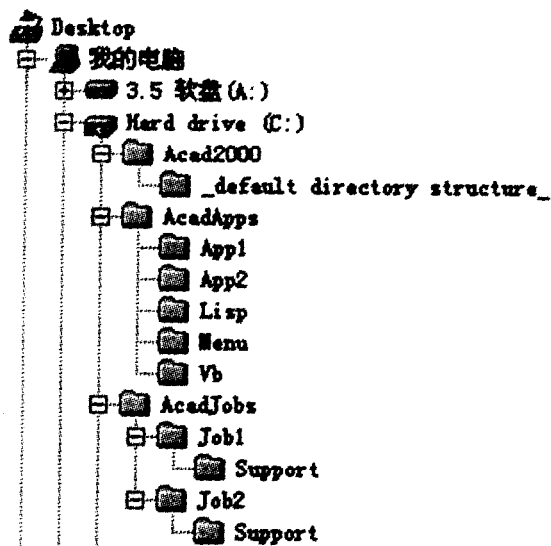


图 1.1 AutoCAD 树状结构图例

这里建立了两个用户目录,AcadApps、AcadJobs 分别存放用户开发的应用程序和工作图形。读者可以为自定义的 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,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。

## 3. 多配置

用户一般只需要一个配置,但有时也会需要多个配置。例如,用户的大多数工作使用鼠标即可完成,但偶尔也需要使用数字化仪。在这种情况下,应该将系统设置为处理多个配置,而不是在每次更换设备时重新进行配置。AutoCAD 为使用定点设备和打印设备进行配置时,所提供的配置信息记录在配置文件中。配置文件 acad2000.cfg 缺省放在包含 AutoCAD 程序文件的目录中,但用户可以指定替换的路径或文件名。

许多 AutoCAD 系统变量的值和“选项”对话框中的配置选项都保存在配置文件中。如果需要使用这些变量和操作参数的不同设置,可将它们相应的值保存在不同的配置文件中。用户必须将 AutoCAD 设置为使用不同的配置文件才能利用多配置的优点。这一功能可通过使用 /c 命令行开关指定替换的配置文件来实现(具体实现请参见 2.1 节)。

#### 4. 多个图形文件目录

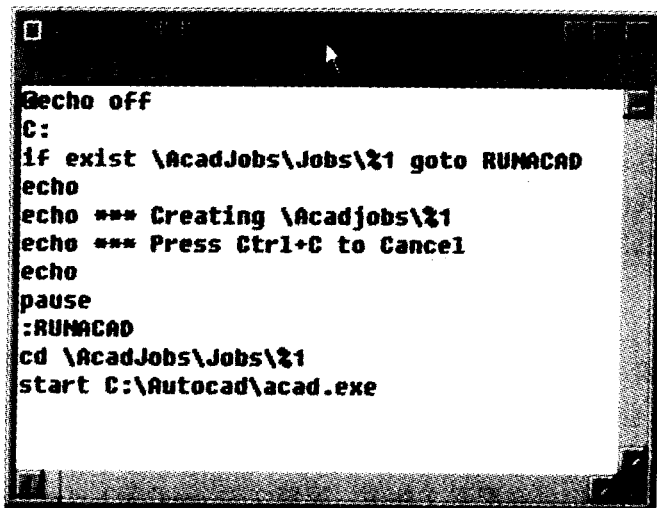
维护多个图形文件目录不仅是一种约定,而且经常是工作中所必须的。将图形和其他相关文件保存在独立的目录中可以简化基本的文件维护工作,这正如 Microsoft 的 ActiveX 技术中所提到的基于文档的管理。本节所讲述内容都是基于前面的示例目录结构,但读者也可以根据需要进行扩展或修改。

读者可将自己的图形子目录都放在 /AcadJobs 目录中。图形子目录仍然可以包含其他子目录,用户可在这些子目录中保存与特定的图形类型或工作相关的支持文件。/AcadJobs/Job1/Support 可包含 /AcadJobs/Job1 中图形文件专用的块和 AutoLISP 文件。在支持路径中指定 Support(不带路径前缀)将把当前目录的 Support 子目录添加到支持路径中。

**注意:**用“选项”对话框指定目录时,AutoCAD 将使用绝对路径。要使用上述的相对路径命名惯例,必须在命令行中用 /s 开关指定支持路径。请参见本书第 3 章。

在启动 AutoCAD 时,如果所需的图形文件目录是当前目录,则该目录中的所有文件和子目录都可以直接访问。读者可以为每项工作创建程序图标或“开始”菜单项,并指定正确的工作目录,也可使用批处理程序来代替图标或菜单,用批处理程序可自动创建新的工作目录。下面的批处理程序先验证是否存在指定的目录,再将该目录设为当前目录,然后运行 AutoCAD。

用记事本或其他 ASCII 文本编辑器将此批处理程序保存为文件 acad.bat,如图 1.2 所示以后就可以随时调用此批处理程序了。当然读者还需要按实际情况更改目录和驱动器的名称。将此文件放在系统搜索路径指定的目录(例如 C:\winnt)中,然后就可以用“开始”菜单上的“运行”命令,或者在资源管理器中双击,来运行此批处理程序。如果将该文件保存为 acad.bat,那么在使用时需要按照如下语法:




```
Echo off
C:
if exist \AcadJobs\Jobs\%1 goto RUNACAD
echo
echo *** Creating \Acadjobs\%1
echo *** Press Ctrl+C to Cancel
echo
pause
:RUNACAD
cd \AcadJobs\Jobs\%1
start C:\Autocad\acad.exe
```

图 1.2 新建 acad.pgp 文件

acad jobname

其中,jobname 是要置为当前工作目录的名称。

 **注意:**此时应在系统中设定了 acad.bat 的搜索路径。

### 1.2.2 AutoCAD 命令搜索过程

通常 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(在下一节中讲述),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输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 Spacebar 键或 Enter 键),则 AutoCAD 调用上次执行的命令。


 **提示:**HELP 是缺省命令。

(2) 如果输入非空,那么 AutoCAD 将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。

 **提示:**自动加载又称按需加载,即 AutoCAD 可以自动寻找未驻留内存的 ARX、AutoLisp、Visual Lisp 程序,利用了这一特性的程序具有可限制代理对象、使 ARX 应用程序加载更具有灵活性、仅在需要调用应用程序时才加载等优点。

(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。

### 1.2.3 AutoCAD 2000 中可自定义的支持文件

AutoCAD 的二次开发主要通过两个途径,即文件开发和程序开发,要进行文件开发就必须了解 AutoCAD 中可自定义的支持文件。本节将向读者介绍 AutoCAD 支持文件。

AutoCAD 用支持文件保存菜单定义、加载 AutoLISP 和 ObjectARX 应用程序以及描述


文字字体等。许多支持文件是文本文件,可以用文本编辑器进行修改。下面列出了可编辑的 AutoCAD 支持文件(大多数在本书中都加以讨论)。这些支持文件按照扩展名的字母顺序列出。

 **注意:**记住在对这些文件进行修改之前先制作备份。

文件	说 明
* .ahp	AutoCAD 帮助文件。相关的帮助索引文件的扩展名为 .hdx
asi.ini	数据库连接的转换映射文件
* .dcl	AutoCAD 对话框控制语言 (DCL) 程序文件
* .lin	AutoCAD 线型定义文件
acad.lin	标准 AutoCAD 线型库文件
acadiso.lin	标准 AutoCAD ISO 线型库文件
* .lsp	AutoLISP 程序文件
acad.lsp	每次启动 AutoCAD 时加载的用户自定义 AutoLISP 例行程序
acaddoc.lsp	每次开始一个图形时加载的用户自定义 AutoLISP 例行程序
* .mln	多线库文件。
* .mnl	AutoCAD 菜单使用的 AutoLISP 例行程序。MNL 文件必须和它所支持的 MNU 文件同名
acad.mnl	标准 AutoCAD 菜单使用的 AutoLISP 例行程序
* .mns	AutoCAD 生成的菜单源文件。包含定义 AutoCAD 菜单的命令字符串和宏语法
acad.mns	标准 AutoCAD 菜单的源文件
* .mnu	AutoCAD 菜单源文件。包含定义 AutoCAD 菜单的命令字符串和宏语法
acad.mnu	标准 AutoCAD 菜单的源文件
* .pat	AutoCAD 填充图案定义文件
acad.pat	标准 AutoCAD 填充图案库文件
acadiso.pat	标准 AutoCAD ISO 填充图案库文件
acad.pgp	AutoCAD 程序参数文件。包含外部命令和命令别名的定义
fontmap.ps	AutoCAD 字体映射文件。由 PSIN 使用,是 AutoCAD PostScript 解释器能识别的全部字体的目录(或字体映射)
acad.psf	AutoCAD PostScript 支持文件,PSOUT 和 PSFILL 命令的主要支持文件
acad.rx	列出启动 AutoCAD 时加载的 ObjectARX 应用程序
* .scr	AutoCAD 脚本文件。脚本文件包含一组 AutoCAD 命令,其运行方式与批处理文件相似
* .shp	AutoCAD 形/字体定义文件。编译后形/字体文件的扩展名为 .shx
acad.unt	AutoCAD 单位定义文件。包含进行单位换算所需的数据

## 1.3 acad.pgp——程序参数文件

AutoCAD 程序参数文件 acad.pgp 是一个 ASCII 文本文件,它用于保存命令定义,读者也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 将在 acad.pgp 中查找该命令。此文件分成两个部分:第一部分定义外部命令,第二部分定义命令别名,下面将分别讲述这两部分内容。

 **提示:**此文件可以用分号 (;) 引入说明文字。

### 1.3.1 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,具体包括:

- 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、电子表格和通信程序。
- 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,需要指定在 AutoCAD 命令提示中使用的命令名称,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五个字段,具体语法如下所示:

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

以上各参数的说明如下:


#### 1. command

在命令提示中输入的命令,也就是你在 AutoCAD 中输入的命令。

 **注意:**如果此名称是 AutoCAD 内部命令名称,则该命令将被忽略,此外这个名称不区分大小写,而且命令中还不能包含空格,这是因为用户输入命令时,AutoCAD 把空格看成回车,即命令结束。

#### 2. executable

输入命令名时传递给操作系统的固定字符串,它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。在大多数情况下,AutoCAD 的命令字段和操作系统命令字段是相同的,在某些情况下则并非如此。这样主要是为了避免 AutoCAD 与该操作系统命令冲突。例如,用户要建立一个使用 DOS 的 COPY 命令的外部命令定义,而 AutoCAD 本身又定义了 COPY 命令,这时二者为了区别就应当不同。

 **提示:**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,按任意组合将这些整数值相加,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。

其中位值 2 和 4 是互斥的,如果同时指定了这两个位值,那么只使用位值 2。应尽量避免不带值 1 的值 2 或 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,用户必须按回车键结束响应;否则,响应可以用空格键或回车键结束响应。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 2,可用 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):

(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 文件被删除。此操作产生的结果与 DXBIN 命令相同。

(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,这种模式不能重定义以前定义的块。AutoCAD 载入该 DXB 文件后,\$ cmd.dxb 文件被删除。INSERT 命令的缺省名称被设置为新定义的块的名称。

下面给出外部命令定义的两个示例。

[例 1] 用于命令窗口的外部命令定义:

```
; Examples of external commands for command windows
CATALOG, DIR /W, 8, 指定文件: ,
DEL, DEL, 8, 要删除的文件: ,
DIR, DIR, 8, 指定文件: ,
EDIT, START EDIT, 9, 要编辑的文件: ,
SH, , 1, * 操作系统命令: ,
SHELL, , 1, * 操作系统命令: ,
START, START, 1, * 要启动的应用程序: ,
TYPE, TYPE, 8, 要列出的文件: ,
```

例 1 给出了命令窗口中的 DOS 命令在 AutoCAD 2000 中的使用方法,如在 DOS 命令行状态下的“DIR /W”在 AutoCAD 2000 中就用“CATALOG”来实现,其余类似,读者也可以试着建立别的外部命令。

[例 2] 用于 Windows 的外部命令:

```
; Examples of external commands for Windows
EXPLORER, START EXPLORER, 1, ,
```

NOTEPAD, START NOTEPAD, 1, \* 要编辑的文件: ,

PBRUSH, START PBRUSH, 1, ,

这里分别定义了 AutoCAD 2000 中使用浏览器、记事本、画笔的外部命令,读者还可以定义别的命令,比如启动附件中的计算器。

### 1.3.2 使用外部命令

AutoCAD 启动时,将搜索支持路径并把它发现的第一个 ACAD.PGP 文件读入。当用户开始绘制一张新图或是调入一张旧图时,AutoCAD 都将对 ACAD.PGP 进行重新初始化。如果用户在修改 ACAD.PGP 文件后想立即使用自己定义的外部命令就必须先对 ACAD.PGP 重新初始化,这可使用 REINIT 命令来实现。执行 REINIT 命令所显示的对话框如图 1.3 所示。

#### 1. 使用 Windows 系统命令

定义外部命令时,Windows 的 start 和 cmd 系统命令非常有用。如果指定的可执行字符串中未使用 start 或 cmd 命令,则必须等该窗口关闭后才能继续使用 AutoCAD。

start 命令启动独立的窗口并运行指定的程序或命令。不带参数的 start 命令将打开一个新的命令提示窗口。start 命令有许多影响新窗口显示的命令行开关。要想运行 Windows 应用程序,建议读者使用不带开关的 start 命令。start 命令也可用于启动与某个应用程序相关联的文档。

[例 3] 用 start 命令直接打开用字处理器创建的文档或 HTML 文件。

首先在命令行输入 START 命令,然后系统提示输入要启动的应用程序,如图 1.4、图 1.5 所示。此时用户可根据需要启动相应程序,如启动 IE 则输入 IEXPLORE。

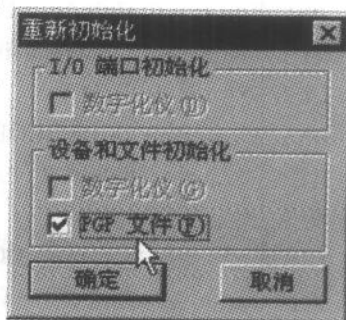


图 1.3 “重新初始化”对话框

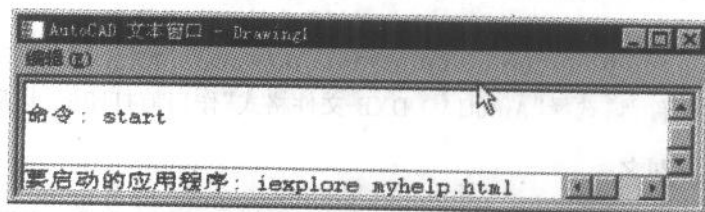


图 1.4 用 start 命令打开 HTML 文件

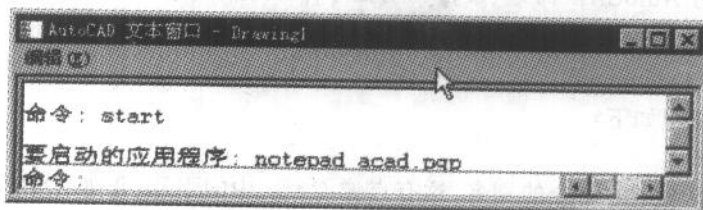



图 1.5 用 start 命令打开 TXT 文件

cmd 命令打开一个命令提示窗口,用作 AutoCAD 的命令解释器。在将控制权返回给 AutoCAD 命令提示之前,必须关闭此窗口。/c 和 /k 命令行开关对外部命令相当有用。/c

执行指定的命令,然后停止(关闭窗口);/k 开关执行指定的命令,然后继续(保持窗口打开)。使用 /k 开关时,用户必须自己关闭命令窗口(用 exit 命令)。

 **提示:**通常,用 start 启动与 AutoCAD 相互独立的新窗口或应用程序;而用 cmd 运行不创建独立窗口的批处理文件或命令脚本,或者创建将控制权返回给 AutoCAD 之前必须关闭的窗口。

## 2. 使用自定义命令

下例定义了三个新命令:RUN、LISTSET 和 DXB2BLK。

RUN, cmd /c,0, \* Batch file to run: ,

LISTSET,cmd /k SET,0

DXB2BLK,cmd /c DXBCOPY,0,DXB file: ,2

RUN 命令运行批处理文件或命令脚本。带 /c 开关的 cmd 命令打开命令窗口,运行批处理文件,然后关闭。

LISTSET 命令显示当前 DOS 环境变量设置。因为此例使用 cmd /k,而不是 start,所以在返回 AutoCAD 之前,用户必须自己关闭命令窗口。要使此窗口保持活动,请使用 start /realtime。关于这些命令和开关的详细信息,请参见 Windows 系统命令文档。

DXB2BLK 命令用指定的 DXB 文件创建块定义。DXB 文件将所有对象转换为直线。此过程有一个非常有用的应用:它提供了一种将文字对象分解为直线的简单方法。

DXB2BLK 将指定的 DXB 文件名传递给 dxbcopy 批处理文件,该批处理文件将该 DXB 文件复制为 \$ cmd.dxb。然后,AutoCAD 用指定的 DXB 文件创建块。答复“DXB file:”提示的名称被用作新块的名称。要创建 dxbcopy.cmd 文件,请在 DOS 提示下输入如下命令:

```
echo copy %1.dxb $ cmd.dxb > dxbcopy.cmd
```

此命令在当前目录中创建 dxbcopy.cmd 文件。将此文件移至 DOS 路径所包含的目录中,或者在 acad.pgp 文件中显式地指出该文件的位置。例如,如果 dxbcopy.cmd 文件在目录 D:\cad 中,请在 acad.pgp 文件的外部命令部分中输入下列语句:

```
DXB2BLK, cmd /c D:\CAD\DXBCOPY,0,DXB file: ,2
```

要创建 DXB 文件,请选择“AutoCAD DXB 文件格式”作当前打印机,然后打印到文件。

### 1.3.3 命令别名

在 acad.pgp 的命令别名部分中为常用的 AutoCAD 命令定义别名,这样可以缩短这些命令。可以为任何 AutoCAD 命令、设备驱动程序命令或外部命令创建别名。

在 acad.pgp 文件中,由逗号分隔的两个字段定义命令别名:

```
abbreviation, * command
```

以上参数的说明如下:

- abbreviation

在命令提示中输入的命令的别名,这有点像 C++ 中的引用,正如在学校你的学号和姓名都代表你一样。

- command

创建别名的 AutoCAD 命令。必须在命令名前输入星号(\*)来表示该行为命令别名。如果一个命令可以透明地输入,则其别名也可以透明地输入。当用户输入命令别名时,Au-



toCAD 在命令提示中显示完整的命令名并执行该命令。

下列的命令别名部分定义的命令别名为标准 acad.pgp 文件的一部分。

A, \* ARC

C, \* CIRCLE

CP, \* COPY

在定义了命令别名后,用户就可以在输入 A、C、CP 时分别代表 ARC、CIRCLE、COPY 命令了。命令别名部分可包含带特殊的负号(-)前缀的命令,如下所示。这使用户可以创建别名来访问某些命令的命令行形式。

BH, \* -BHATCH

BD, \* -BOUNDARY

学习完本节后,读者完全可以按照自己的习惯定义许多命令别名,比如前面的画圆命令 CIRCLE,你只需要输入“C”就可以达到目的,效率当然会有极大的提高,表 1.1 给出了 AutoCAD 2000 中系统定义的命令别名,希望读者熟记。

表 1.1 AutoCAD 2000 中系统定义的命令别名

命令别名	AutoCAD 命令名	命令别名	AutoCAD 命令名
3A	* 3DARRAY	3DO	* 3DORBIT
3F	* 3DFACE	3P	* 3DPOLY
A	* ARC	ADC	* ADCEnter
AA	* AREA	AL	* ALIGN
AP	* APLOAD	AR	* ARRAY
ATT	* ATTDEF	- ATT	* - ATTDEF
ATE	* ATTEDIT	- ATE	* - ATTEDIT
ATTE	* - ATTEDIT	B	* BLOCK
- B	* - BLOCK	BH	* BHATCH
BO	* BOUNDARY	- BO	* - BOUNDARY
BR	* BREAK	C	* CIRCLE
CH	* PROPERTIES	- CH	* CHANGE
CHA	* CHAMFER	COL	* COLOR
COLOUR	* COLOR	CO	* COPY
D	* DIMSTYLE	DAL	* DIMALIGNED
DAN	* DIMANGULAR	DBA	* DIMBASELINE
DBC	* DBCONNECT	DCE	* DIMCEnter
DCO	* DIMCONTINUE	DDI	* DIMDIAMETER
DED	* DIMEDIT	DI	* DIST
DIV	* DIVIDE	DLI	* DIMLINEAR
DO	* DONUT	DOR	* DIMORDINATE
DOV	* DIMOVERRIDE	DR	* DRAWORDER
DRA	* DIMRADIUS	DS	* DSETTINGS
DST	* DIMSTYLE	DT	* DTEXT
DV	* DVIEW	E	* ERASE