

AutoCAD 2002 应用与开发系列

中文版 AutoCAD 2002/2000 Visual LISP 开发指南



康博 编著



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



(京) 新登字 158 号

内 容 简 介

本书是“AutoCAD 2002 应用与开发系列”之一,通过大量的函数举例及程序系统地讲解了在 AutoCAD 2002 和 AutoCAD 2000 环境下利用 Visual LISP 进行二次开发的技术,内容包括 Visual LISP 开发环境、应用 Visual LISP 进行程序的开发和调试、编译应用程序、程序维护、使用 ActiveX 和反应器、AutoLISP 基础、AutoCAD 通讯、操作 AutoCAD 对象、设计和管理对话框、可编程对话框参考和 AutoLISP 参考等。

本书内容深入、结构严谨、通俗易懂,涵盖了运用 Visual LISP 进行二次开发的各个方面,适用于广大 AutoCAD 2002/2000 的开发人员和中高级用户阅读。

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: 中文版 AutoCAD 2002/2000 Visual LISP 开发指南

作 者: 康 博 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: 李 阳

印 刷 者: 北京市密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 41.75 字数: 990 千字

版 次: 2001 年 8 月第 1 版 2001 年 8 月第 1 次印刷

书 号: ISBN 7-302-04654-9/TP·2761

印 数: 0001~5000

定 价: 62.00 元

丛书前言

AutoCAD 2002 版的成功推出,标志着 Autodesk 公司顺利实现了面向 Internet 设计的战略性转移,使得 AutoCAD 软件在运行速度、图形处理以及网络化设计功能等方面都达到了崭新的水平。

为了满足广大读者掌握 AutoCAD 2002 的迫切需要,我们组织了一批长期从事 AutoCAD 教学、开发和应用的专业人士,潜心测试和研究了这一软件的新增功能和特点,精心策划和编写了《AutoCAD 2002 应用与开发》系列丛书,具体书目如下:

- AutoCAD 2002 实用培训教程
- 中文版 AutoCAD 2002 实用指南
- 中文版 AutoCAD 2002 高级应用技巧
- 中文版 AutoCAD 2002 机械图形设计
- 中文版 AutoCAD 2002 建筑图形设计
- AutoCAD 2002 三维图形设计
- AutoCAD 2002/2000 VBA 开发指南
- 中文版 AutoCAD 2002/2000 Visual LISP 开发指南
- AutoCAD 2002 二次开发技术指南

本丛书既有引导初学者入门的培训教程,又有面向不同行业中高级读者的软件功能的全面展示和实际应用。既深入剖析了 AutoCAD 2002 的二次开发语言的核心技术,又以实例形式具体介绍了 AutoCAD 2002 在建筑和机械行业的实际应用。

整套丛书内容关联,自成体系,为各种层次、各行各业的用户提供了系统完整的 AutoCAD 2002 应用与开发解决方案。

愿本丛书能成为 AutoCAD 用户的良师益友,并衷心希望广大读者对丛书的不足提出宝贵的建议。

U360/01

前 言

多年来, AutoLISP 已经成为在计算机辅助设计(CAD)领域进行二次开发的标准。而 Visual LISP 语言则代表着在 AutoCAD 上做开发的下一代 LISP 语言,同时还为基于 AutoCAD 的 AutoLISP 语言增加了许多重要的功能。

AutoCAD 2002 新特点

2001 年, Autodesk 公司推出了 AutoCAD 系列软件的最新版本——AutoCAD 2002, 该软件为用户提供了强大的图形设计功能, 是一个高效、易学易用的绘图工具。与以前的版本相比, 该软件具有以下特性:

- 关联标注

AutoCAD 2002 提供了一个全真的关联标注功能, 使得已有的尺寸标注可以随着几何对象的改变而自动更新, 大大提高了绘图效率。

- 块属性管理器

块属性管理器提供在块内修改属性的简便方法, 可以使得当这些属性改变时, 能马上反映到已有的块引用上。只要选中一个已插入的块, 编辑它的属性值和特性, 所有的改变会立即显示在绘图区内, 也可以将块属性数据延伸到一个文件格式。

- 图层管理器

新的图层管理器为用户提供了将图形从一个图层结构转换成另一个图层结构的功能, 还可以使用记忆功能保存上一个图层状态。

- XML 设计

XML 设计提供了通过 Web 页传送图形几何信息的功能, 它是以 XML 传送协议来传送三维模型和图像信息的。

AutoLISP 和 Visual LISP 的关系

LISP 语言是在 20 世纪 50 年代末期诞生的一种编程语言, 最初被设计用在人工智能(AI)应用程序中, 今天该语言也仍然是人工智能应用程序的基础。

LISP 语言是唯一适合于进行 CAD 项目开发的非结构化设计的语言。20 世纪 80 年代中期, AutoCAD 在其 2.1 版本中引入了 LISP 作为二次开发 AutoCAD 的编程语言, 并基于 LISP 语言的发展形成了一套自己的应用程序编程接口语言——AutoLISP, 该编程语言可以用来扩展 AutoCAD 功能、定制 AutoCAD 和在 AutoCAD 上做二次开发。

Visual LISP(VLISP)是一个用来帮助使用 AutoLISP 语言进行程序开发的软件工具。VLISP 的集成开发环境提供了许多工具可以帮助简化诸如源代码的生成及修改、程序测试和程序调试等任务。除此之外, Visual LISP 还提供了一个新的功能,利用这个新功能可以发布独立的用 AutoLISP 语言编制的 ObjectARX 应用程序。

例如在过去,用 AutoLISP 语言开发基于 AutoCAD 的应用程序意味着要要在一个独立的文本编辑器(例如 DOS 的 edit 和 Windows 的写字板等)中输入程序代码,然后再把程序加载进 AutoCAD 并运行。而如果要进行程序调试,只能在源程序中加入一些语句来打印输出一些变量的值,并以此作为调试点,用户还必须自己判断哪些变量的值是正确的,哪些需要再检查。如果没有发现足够的信息确定是否出错,或者在什么地方出错,就不得不再返回到源程序中并加入更多的调试点。最后,当程序已经运行得很好时,还需要再把那些调试代码删除或注释掉。而 Visual LISP 的集成开发环境简化了这一切过程。

Visual LISP 提供的新功能

Visual LISP 作为一个集成开发环境,提供了以下新功能:

- 一个完整的 AutoLISP 仿真环境,具有透明的加载编译技术。
- 强大的语法检查器,可以识别出 AutoLISP 语法是否错误,以及调用 AutoLISP 函数时参数的输入是否正确。
- 提供一个文件编译器,可以改善程序的运行速度和提供安全、高效的交互平台。
- 特别为 AutoLISP 语言提供了源程序调试器。通过应用强有力的断环处理,可以为程序的调试提供最大的灵活性。这个工具也支持在一个窗口中单步运行 AutoLISP 源程序,而且实时地在 AutoCAD 的图形窗口中显示程序的运行结果。
- 支持 AutoLISP 和 DCL 语言的文本文件中的彩色代码显示功能,并支持 AutoLISP 语法的其他功能。
- 提供一个 AutoLISP 格式化器,可以设置程序代码的书写结构,使程序具有一个更容易读懂的格式。
- 综合的检验器和监视器实用工具,用户可以非常方便地在浏览和修改数据结构时获取变量和表达式的值。这些特征可以用来检验 AutoLISP 数据和 AutoCAD 图形实体。
- 为 AutoLISP 函数提供了上下文相关的帮助系统。
- 强有力的符号名查找和自动匹配特征。
- 项目管理系统,可以使维护多文件应用程序变得非常容易。
- 提供打包功能,可以把编译过的 AutoLISP 文件打包成为一个 ARX 模块。
- 桌面的存储与恢复功能,可以从其他 Visual LISP 进程中保护和再利用 Windows 环境资源。
- 智能的系统控制台使 AutoLISP 用户开发应用程序时的方便性和高效性达到了一个新的水平。Visual LISP 控制台的基本功能和 AutoCAD 文本屏幕的功能是相适应的,它还提供了一些方便的交互功能,如历史记录滚动和完全输入的行编辑功能。

Visual LISP 是一个用来开发和运行 AutoLISP 程序的集成开发环境。它包含一套自己的窗口和菜单，这些窗口及菜单与 AutoCAD 有显著的不同。但是，Visual LISP 不是一个独立的软件，它不能独立在 AutoCAD 之外运行。无论什么时候想要在 Visual LISP 中工作，都首先要启动 AutoCAD。并且有时当从 Visual LISP 环境运行 AutoLISP 程序时，还需要和 AutoCAD 进行交互工作，以便于操作 AutoCAD 图形实体。

注意

无论什么时候运行 Visual LISP，都要确保 AutoCAD 处在运行状态。

本书内容结构

本书解释了如何使用 Visual LISP 集成开发环境以及如何生成和运行 AutoLISP 应用程序，另外还介绍了 AutoLISP 语言的结构。

本书适用于具有使用 AutoCAD 的经验，并掌握 Microsoft Windows 基本用户级技能的用户。它并不要求用户具有使用 AutoLISP 的经验，但如果具有这方面的经验，可以使学习变得更容易。

本书分为以下几部分：

- “应用 Visual LISP 环境”部分说明如何使用 Visual LISP 开发并测试 AutoLISP 程序。
- “使用 AutoLISP 语言”部分是关于 AutoLISP 语言元素和结构的详细指导。
- “可编程对话框”部分说明如何在 AutoLISP 应用程序中设计和使用对话框。
- “AutoLISP 参考”部分将函数按字母顺序排列，并说明了各函数的功能、参数、返回值及具体调用。
- “附录”包括 AutoLISP 函数的分类概要和 AutoLISP 错误代码的信息。

安装 Visual LISP

在用户安装 AutoCAD 2002 时，如果选择了典型安装或者全部安装，那么 Visual LISP 也就会自动安装到当前计算机中。值得注意的是，典型安装并不会安装 Visual LISP 的教程部分，而本书的“程序维护”一章要用到教程部分，这时可以再次运行 AutoCAD 的安装程序添加此部分。

目 录

第 1 章 应用 Visual LISP 环境	1
1.1 启动 Visual LISP	2
1.2 Visual LISP 环境界面	3
1.2.1 菜单概述	4
1.2.2 控制台窗口	6
1.2.3 文本编辑器概述	7
1.2.4 加载并运行 AutoLISP 应用程序	8
1.2.5 加载 AutoLISP 扩展函数	9
1.2.6 退出 Visual LISP 环境	10
第 2 章 应用 Visual LISP 进行程序开发	11
2.1 系统控制台	12
2.1.1 系统控制台的功能	12
2.1.2 系统控制台的快捷菜单	13
2.1.3 分隔符处理	14
2.1.4 上下文相关的帮助	15
2.1.5 记录控制台事件	15
2.1.6 使用控制台历史记录	16
2.1.7 在 Visual LISP 和 AutoCAD 间切换	17
2.2 应用文本编辑器	18
2.2.1 文件操作	18
2.2.2 彩色代码显示	20
2.2.3 上下文相关的帮助	21
2.2.4 文本编辑器的快捷菜单	22
2.2.5 字段处理	23
2.2.6 快捷功能	23
2.3 格式化代码	33
2.3.1 运行格式化器	33
2.3.2 Visual LISP 格式化基础	33
2.4 语法错误检查	41
2.4.1 检查括号是否匹配	42
2.4.2 应用彩色代码功能检查语法错误	43
2.4.3 用检查命令查找语法错误	44
2.5 Visual LISP 环境和格式设置选项	45

2.5.1 窗口属性	45
2.5.2 环境选项	48
2.5.3 保存设置	52
第 3 章 调试程序	53
3.1 在 Visual LISP 中进行调试	54
3.1.1 设置断点	57
3.1.2 单步调试程序	58
3.1.3 跟踪表达式的运行结果	60
3.1.4 在程序运行时跟踪变量	60
3.1.5 继续运行程序	61
3.1.6 在“自动执行”模式下运行	62
3.2 Visual LISP 的调试功能	62
3.2.1 开始调试	63
3.2.2 中断循环	63
3.2.3 断点	65
3.3 数据检验工具	68
3.3.1 “监视”窗口	68
3.3.2 “跟踪堆栈”窗口	70
3.3.3 “符号服务”对话框	76
3.3.4 检验器	78
3.3.5 查看 AutoCAD 图形图元	84
第 4 章 编译应用程序	87
4.1 编译和链接程序	88
4.1.1 如何应用编译器	88
4.1.2 编译举例	90
4.1.3 加载并运行编译过的程序	91
4.1.4 链接函数调用	92
4.2 生成应用程序	92
4.2.1 创建新的应用程序	93
4.2.2 应用程序输出文件介绍	97
4.2.3 加载和运行 Visual LISP 应用程序	98
4.2.4 修改应用程序选项	98
4.2.5 重新编译应用程序	99
4.2.6 更新应用程序	99
4.3 针对多文档环境的设计方法	100
4.3.1 理解名称空间	100

4.3.2 在自身的名称空间中运行应用程序	102
4.3.3 在名称空间中共享数据	106
4.3.4 MDI 环境下的错误处理	108
4.3.5 在 MDI 环境下使用 AutoLISP 的限制	109
第 5 章 程序维护	110
5.1 管理多个 Visual LISP 文件	111
5.2 定义一个工程	112
5.2.1 设置工程特性	113
5.2.2 工程窗口	116
5.3 工程操作	120
5.3.1 打开一个工程	120
5.3.2 在工程文件中查找字符	120
5.3.3 由工程编译应用程序	121
5.4 优化应用程序	122
5.4.1 定义编译选项	122
5.4.2 选择编译模式	123
5.4.3 选择链接模式	124
5.4.4 安全优化	125
第 6 章 使用 ActiveX	127
6.1 在 AutoLISP 中使用 ActiveX 对象	128
6.2 AutoCAD 对象模型	132
6.2.1 对象特性	132
6.2.2 对象方法	133
6.2.3 对象集合	133
6.3 访问 AutoCAD 对象	134
6.3.1 用检验工具查看对象特性	134
6.3.2 从 AutoCAD 应用程序对象中获取其他对象	136
6.3.3 过程概要	137
6.3.4 编程技巧	138
6.4 应用 Visual LISP 函数实现 ActiveX 方法	139
6.4.1 确定所需的 Visual LISP 函数	140
6.4.2 确定如何调用一个函数	141
6.4.3 将 AutoLISP 数据类型转换为 ActiveX 数据类型	142
6.4.4 读取和更新对象特性	147
6.4.5 使用通过参数返回值的 ActiveX 方法	149
6.4.6 显示一个对象的特性和方法	150

6.4.7 对象集合操作	152
6.4.8 释放对象和释放内存	155
6.4.9 转换对象引用	155
6.4.10 处理 ActiveX 方法返回的错误	157
6.5 使用 ActiveX 与其他应用程序交互	159
6.5.1 输入类型库	159
6.5.2 建立与应用程序的连接	161
6.5.3 应用程序编码样例	161
6.5.4 不输入类型库时使用 ActiveX	165
第 7 章 使用反应器	167
7.1 理解反应器类型和事件	168
7.2 定义回调函数	170
7.3 创建反应器	172
7.3.1 使用对象反应器	173
7.3.2 将数据附着到反应器对象	175
7.4 在多重名称空间中使用反应器	175
7.5 查询、修改和删除反应器	177
7.5.1 检验反应器	177
7.5.2 使用函数调用查询反应器	178
7.5.3 修改反应器	179
7.5.4 删除反应器	180
7.6 临时反应器和永久反应器	181
7.7 反应器使用规则	181
第 8 章 AutoLISP 基础	183
8.1 AutoLISP 表达式	184
8.1.1 AutoLISP 数据类型	185
8.1.2 AutoLISP 函数语法	188
8.2 AutoLISP 程序文件	188
8.2.1 格式化 AutoLISP 程序	189
8.2.2 注释	189
8.3 AutoLISP 变量	190
8.3.1 显示变量值	190
8.3.2 nil 变量	191
8.3.3 预定义变量	191
8.4 数字操作	192
8.5 字符串操作	193

8.6 基本输出函数	194
8.6.1 显示信息	194
8.6.2 字符串中的控制字符	195
8.6.3 通配符匹配	196
8.7 比较和条件	197
8.8 表处理	198
8.8.1 点类型表	199
8.8.2 点对	201
8.9 符号和函数操作	202
8.9.1 C: XXX 函数	203
8.9.2 函数中的局部变量	205
8.9.3 带参数的函数	207
8.10 错误处理	208
8.10.1 使用 *error* 函数	209
8.10.2 捕获错误和继续执行程序	210
第9章 和 AutoCAD 通讯	213
9.1 查询和 Command 函数	214
9.1.1 Command 函数	214
9.1.2 系统和环境变量	216
9.1.3 系统配置控制	216
9.2 显示控制	217
9.2.1 控制菜单	217
9.2.2 图形和文本窗口显示控制	219
9.2.3 低级图形控制	220
9.3 获得用户输入	220
9.3.1 getxxx 函数	220
9.3.2 用户输入函数的条件控制	222
9.4 几何实用函数	225
9.4.1 对象捕捉	225
9.4.2 文本范围	226
9.5 转换	229
9.5.1 字符串转换	229
9.5.2 角度转换	231
9.5.3 ASCII 代码转换	232
9.5.4 单位转换	235
9.5.5 坐标系转换	237

9.6 文件处理	239
9.6.1 文件查找	239
9.6.2 存取帮助文件	240
9.7 设备访问和控制	241
9.7.1 获取用户输入	241
9.7.2 校准数字化仪	241
第 10 章 操作 AutoCAD 对象	244
10.1 操作选择集	245
10.1.1 选择集过滤器表	247
10.1.2 在 AutoLISP 和 ObjectARX 应用程序间传递选择集	253
10.2 对象操作	254
10.2.1 图元名称函数	254
10.2.2 图元数据函数	260
10.2.3 图元数据函数和图形屏幕	265
10.2.4 多段线和轻装多段线	266
10.2.5 非图形对象操作	267
10.3 扩展图元数据 —— xdata	268
10.3.1 扩展图元数据的组织	269
10.3.2 注册应用程序	271
10.3.3 检索扩展图元数据	272
10.3.4 扩展图元数据附着到图元上	274
10.3.5 扩展图元数据内存管理	275
10.3.6 扩展图元数据中的句柄	275
10.4 Xrecord 对象	276
10.5 获取符号表和词典	277
10.5.1 符号表	277
10.5.2 词典条目	278
第 11 章 设计对话框	280
11.1 对话框部件	281
11.2 用 DCL 定义对话框	282
11.2.1 base.DCL 和 acad.DCL 文件	282
11.2.2 引用 DCL 文件	283
11.2.3 DCL 语法	283
11.3 用 Visual LISP 显示对话框	286
11.3.1 预览错误处理	287
11.3.2 DCL 文件的语义核查	288

11.4 调整对话框的布局	289
11.4.1 在控件组中分配控件	289
11.4.2 调整控件间距	290
11.4.3 调整右端和底部的空间	290
11.4.4 调整加框行和列周围的空间	291
11.4.5 自定义退出按钮文本	292
11.5 设计指南	294
11.5.1 美观和功效	294
11.5.2 前后一致的设计和清晰明了的语言	294
11.5.3 用户控制	294
11.5.4 容错处理	296
11.5.5 提供帮助	296
11.5.6 考虑残疾用户	296
11.5.7 单词字母的大写	297
11.5.8 避免缩略语	297
11.5.9 布局	297
11.5.10 控件的大小和位置	298
11.5.11 禁用控件	298
11.5.12 嵌套对话框	298
11.5.13 隐藏对话框	298
11.5.14 提供默认值	299
11.5.15 处理键盘输入	299
11.5.16 预定义控件和控件组的指导	300
11.5.17 错误处理	302
第 12 章 管理对话框	304
12.1 使用 AutoLISP 程序控制对话框	305
12.1.1 简介	305
12.1.2 对话框打开时限制函数	306
12.2 动作表达式和回调	307
12.2.1 动作表达式	308
12.2.2 回调原因	309
12.3 控件处理	310
12.3.1 初始化模式和值	310
12.3.2 在回调时改变模式和值	310
12.3.3 处理单选控件组	312
12.3.4 处理滑块	312

12.3.5 处理编辑框	313
12.4 嵌套对话框	314
12.5 隐藏对话框	314
12.6 列表框和弹出式列表	317
12.6.1 列表操作	318
12.6.2 处理列表框	319
12.7 图像控件和按钮	320
12.7.1 创建图像	320
12.7.2 处理图像按钮	322
12.8 特定应用数据	323
12.9 DCL 错误处理	324
12.10 对话框函数概要	325
12.10.1 函数序列	325
12.10.2 样例——块定义对话框	326
第 13 章 可编程对话框参考	327
13.1 控件属性	328
13.1.1 属性类型	328
13.1.2 受限制的属性	329
13.1.3 用户定义的属性	329
13.2 预定义属性摘要	330
13.3 DCL 属性目录	331
13.4 DCL 控件功能摘要	339
13.4.1 预定义活动控件	339
13.4.2 控件组	339
13.4.3 用于装饰和说明的控件	340
13.4.4 文本控件组	340
13.4.5 对话框退出按钮和错误控件	340
13.5 DCL 控件目录	341
13.6 可编程对话框函数摘要	354
第十四章 AutoLISP 参考	356
附录 A AutoLISP 函数摘要	621
A.1 分类概要	622
A.2 基本函数	622
A.2.1 应用程序处理函数	622
A.2.2 数学函数	623
A.2.3 等量和条件函数	624

A.2.4 错误处理函数	625
A.2.5 函数处理函数	626
A.2.6 表操作函数	626
A.2.7 字符串处理函数	628
A.2.8 符号处理函数	629
A.3 实用函数	629
A.3.1 转换函数	630
A.3.2 设备访问函数	630
A.3.3 显示控制函数	630
A.3.4 文件处理函数	631
A.3.5 几何函数	632
A.3.6 查询和命令函数	633
A.3.7 用户输入函数	634
A.4 选择集、对象和符号表函数	634
A.4.1 扩展数据处理函数	634
A.4.2 对象处理函数	635
A.4.3 选择集操作函数	636
A.4.4 符号表和词典处理函数	636
A.5 内存管理函数	637
A.6 Visual LISP 的 AutoLISP 扩展函数	638
A.6.1 ActiveX 集合操作函数	638
A.6.2 ActiveX 数据转换函数	638
A.6.3 ActiveX 方法调用函数	639
A.6.4 ActiveX 对象处理函数	640
A.6.5 ActiveX 特性处理函数	640
A.6.6 曲线测量函数	641
A.6.7 词典函数	642
A.6.8 对象处理函数	642
A.7 反应器函数	643
A.8 VLX 名称空间函数	645
A.9 名称空间通信函数	646
A.10 Windows 注册表函数	647
附录 B AutoLISP 错误代码	648

第 1 章 应用 Visual LISP 环境

本章介绍一些 Visual LISP 交互开发环境(IDE)的基本内容,并且介绍如何在 Visual LISP 中运行 AutoLISP 程序。

本章内容包括:

- 启动 Visual LISP
- 开发 Visual LISP 用户界面
- Visual LISP 菜单介绍
- 控制台概述
- 文本编辑器概述
- 加载和运行 AutoLISP 程序

1.1 启动 Visual LISP

Visual LISP(VLISP)交互式开发环境是运行在 AutoCAD 外的一套窗口中的,要使用 Visual LISP 开发 AutoCAD 应用程序,首先需要启动 Visual LISP。启动 Visual LISP 的步骤如下:

(1) 启动 AutoCAD。

(2) 在 AutoCAD 的菜单里,选择“工具”→AutoLISP →“Visual LISP 编辑器”命令,或者在命令行上输入下面的命令:

```
vlisp
```

弹出的“Visual LISP 为 AutoCAD”窗口,如图 1-1 所示。

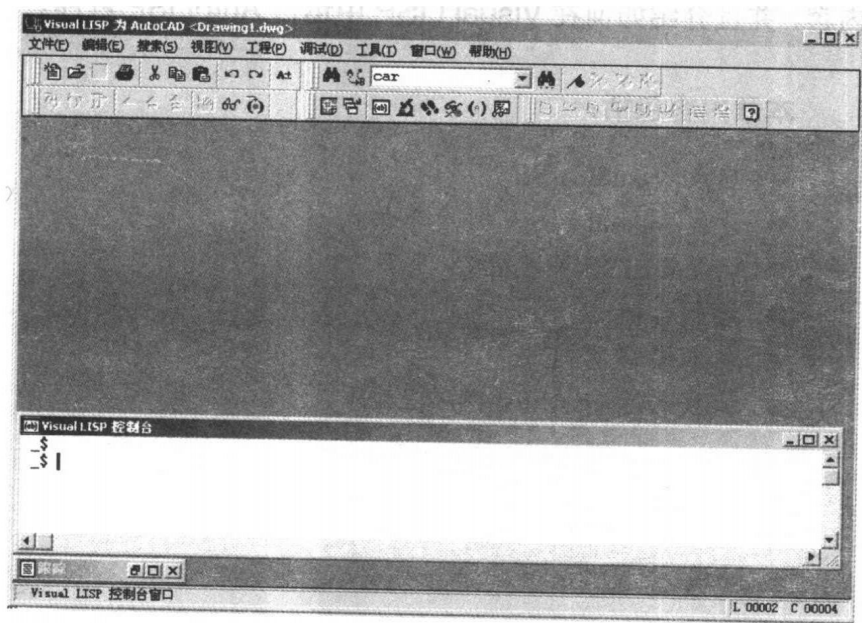


图 1-1

当加载 Visual LISP 时,一个或多个 Visual LISP 的窗口会暂时显示在屏幕上。Visual LISP 可以存储上一次退出时正在打开的窗口,在下一次运行时,会自动打开并恢复这些窗口里的内容。

在任何时候使用菜单命令或 VLISP 命令都可以从 AutoCAD 返回到 Visual LISP IDE 中。

说明

AutoCAD 也可以识别 vlide 命令来启动或返回 Visual LISP 的,这个命令是 Visual LISP 交互开发环境的意思,AutoCAD 是通过 vlide 命令来调用 Visual LISP 的,因此有时在 AutoCAD 命令窗口中会显示 VLIDE。