

AutoCAD

Visual LISP

TP312/L1
及社

AutoCAD

2000

应用系列之三

附磁盘



Visual LISP

实用教程



康博创作室 编著

曹康 审校

人民邮电出版社

AutoCAD 2000 应用系列之三

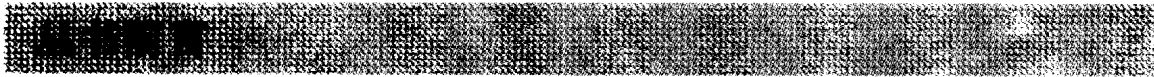
Visual LISP 实用教程

康博创作室 编著
曹 康 审校

人民邮电出版社

Visual LISP 是 Autodesk 公司在 AutoLISP 语言基础上开发出的新一代可视化的 LISP 语言,而且增加了诸如 ActiveX 和反应器之类的新功能。本书全面系统地介绍了运用 Visual LISP 语言进行二次开发的各个方面。书中第一章至第四章讲解了 Visual LISP 的基础知识和集成开发环境;第五章至第八章介绍了利用 Visual LISP 生成、调试以及维护应用程序的方法;第九章和第十章展示了 Visual LISP 的高级功能、ActiveX 和反应器;第十一章按字母顺序列出了所有 Visual LISP 函数的使用方法,同时也包括了传统的 AutoLISP 函数。

本书内容全面,实用性强,是一本指导专业开发人员用 Visual LISP 语言进行程序开发的优秀教程,同时也可作为对 AutoCAD 二次开发感兴趣的读者的自学用书。



AutoCAD 2000 的成功推出,标志着 Autodesk 公司顺利实现了其 CAD 软件技术向 Windows/Objects/Web 的跨世纪技术战略转移,使 CAD 用户能够更加方便和高效地将其应用于工程设计、教育、科研和其他各领域。

为了让各行各业的用户都能在较短的时间内学会和掌握 AutoCAD 2000 这一优秀的设计软件,以推动我国 CAD/CAM 领域的应用发展,康博创作室组织了一批长期从事 AutoCAD 教学、研究和应用的专家学者,精心策划编写了《AutoCAD 2000 应用系列》,本套丛书共包括以下 6 本:

- 《AutoCAD 2000 实用教程》
- 《中文版 AutoCAD 2000 实用教程》
- 《Visual LISP 实用教程》
- 《AutoCAD 2000 定制与开发》
- 《中文版 AutoCAD 2000 平面与三维造型设计技巧》
- 《中文版 AutoCAD 2000 高级应用技巧》

本丛书在内容编写和结构编排上充分考虑到国内用户的阅读习惯和应用实际。由浅入深、循序渐进、相互补充、配套,为用户从入门到精通提供了一揽子解决方案。

本丛书既有面向初、中级读者的软件功能的全面展示,也有面向中、高级读者的高级应用技巧;既介绍了 AutoCAD 2000 定制与开发的原理与方法,又深入剖析了全新版本的可视化二次开发工具 Visual LISP;同时,提供了 AutoCAD 2000 在机械、建筑等主要领域的平面与三维图形设计技巧。

愿《AutoCAD 2000 应用系列》能成为广大读者的良师益友,并衷心希望广大读者对本丛书的不足或缺点提出批评和建议,以便我们在今后的工作中不断改进。

康博创作室

1999 年 8 月

自 AutoCAD 引入 AutoLISP 编程语言以来,AutoLISP 已经成为广大 AutoCAD 用户进行二次开发、程序定制的主要工具。但它本身也有一些局限性,例如,作为解释程序语言,其运行速度较慢,没有一个完整的编程环境,很难进行程序调试和加密等等。

随着 Visual LISP 语言的面世,原来 AutoLISP 语言的局限性已经不复存在,同时还为基于 AutoCAD 的 AutoLISP 语言增加了许多重要的功能。例如,Visual LISP 扩展了通过 Microsoft 的 ActiveX Automation 接口和 AutoCAD 内部对象进行交互的能力,还增加了使 AutoLISP 语言通过对象反应器对 AutoCAD 内部对象进行反应的能力。作为一个开发工具,Visual LISP 提供了一个完整的集成开发环境 (IDE),包括编译器、调试器和其他一些能够更快捷、更方便地用来在 AutoCAD 上做用户化工作的增强工具。

AutoLISP 语言相对于常用的 Visual Basic 和 Visual C++ 等语言而言是一种难度较大的 AutoCAD 专用开发语言,即使您已经掌握了 AutoLISP 语言,初次使用 Visual LISP 时也会感到陌生,甚至有无从下手的感觉。为了满足广大读者迅速学会 Visual LISP 语言的需要,康博创作室组织了几位长期从事 AutoCAD 应用与开发的博士编写了此书。本书独到的组织结构和内容编排方式,以及丰富的开发示例,定会引导读者高效地掌握这一开发工具。

本书配套磁盘中收录了书中所有示例的源代码以及其他一些常用的代码示例,这些程序都经过了精心筛选和严格调试,供读者上机练习之用,盘中代码的版权属康博创作室所有,读者不得将其用于商业目的。

为了表达上的简洁性,Visual LISP 有时在正文中简略为 LISP;书中用“菜单

名→菜单选项(第 1 层)→菜单选项(第 2 层)……→菜单选项(第 N 层)”来描述复杂的菜单选项操作。在以上描述中,所选择的菜单命令以所写的顺序从菜单中选出。例如“Draw→Point→Divide”表示在“Draw”菜单的“Point”子菜单中选择“Divide”命令。另外,书中用小五号书宋体表示 Visual LISP 程序和系统弹出的命令行及用户的相应输入。

本书由康博创作室策划完成,曹康博士审校。在本书编写过程中得到了几家 CAD 应用与开发公司的技术人员的大力支持和协助,谨向他们致以衷心的感谢。

由于笔者水平与经验有限,不妥之处在所难免,欢迎广大读者多提宝贵意见。您的意见和建议可发 E-mail 至 kang-bo@263.net。我们将认真答复您所提出的问题,并衷心感谢您对我们工作的关心和厚爱。

作 者

1999 年 6 月

第一章 Visual LISP 简介	1
1.1 AutoLISP 和 Visual LISP 的关系	1
1.2 Visual LISP 提供的新功能	2
1.3 安装 Visual LISP	3
1.4 启动 Visual LISP	5
1.5 熟悉 Visual LISP 用户界面	6
1.5.1 菜单概述	8
1.5.2 控制台窗口	11
1.5.3 文本编辑器概述	12
1.5.4 加载并运行 AutoLISP 应用程序	13
1.5.5 退出 Visual LISP 环境	14
第二章 AutoLISP 基础	16
2.1 AutoLISP 表达式	16
2.1.1 AutoLISP 数据类型	17
2.1.2 AutoLISP 函数语法	20
2.2 AutoLISP 程序文件	20
2.2.1 格式化 AutoLISP 程序	21
2.2.2 注释	21
2.2.3 彩色化代码显示	22
2.3 AutoLISP 变量	22
2.3.1 显示变量值	23
2.3.2 Nil 变量	23
2.3.3 预定义变量	23
2.4 数字操作	24
2.5 字符串操作	24
2.6 基本输出函数	26
2.6.1 显示信息	26
2.6.2 字符串中的控制字符	27
2.6.3 通配符匹配	28
2.7 比较和条件	29
2.8 表处理	29

2.8.1	点类型表	31
2.8.2	点对	32
2.9	符号和函数操作	33
2.9.1	C: XXX 函数	34
2.9.2	函数中的局部变量	36
2.9.3	带参数的函数	38
第三章	与 AutoCAD 通信	39
3.1	查询和 Command 函数	39
3.1.1	Command 函数	40
3.1.2	系统和环境变量	42
3.1.3	系统配置控制	42
3.2	显示控制	42
3.2.1	控制菜单	43
3.2.2	图形和文本窗口显示控制	45
3.2.3	低级图形控制	45
3.3	获得用户输入	46
3.3.1	getxxx 函数	46
3.3.2	用户输入函数的条件控制	48
3.4	几何实用函数	50
3.4.1	对象捕捉	51
3.4.2	文本范围	51
3.5	转换	54
3.5.1	字符串转换	54
3.5.2	角度转换	56
3.5.3	ASCII 代码转换	57
3.5.4	单位转换	59
3.5.5	坐标系转换	62
3.6	文件操作	63
3.6.1	文件查找	64
3.6.2	存取帮助文件	65
3.7	设备获取和控制	65
3.7.1	获取用户输入	65
3.7.2	校准数字化仪	65
第四章	操作 AutoCAD 对象	68
4.1	操作选择集	68

4.1.1 选择集过滤器表	71
4.1.2 在 AutoLISP 和 ObjectARX 应用程序间传递选择集	76
4.2 对象操作	77
4.2.1 实体名函数	77
4.2.2 实体数据函数	82
4.2.3 实体数据函数和图形屏幕	87
4.2.4 多义线和轻装多义线	88
4.2.5 非图形对象操作	89
4.3 扩展实体数据——xdata	90
4.3.1 扩展实体数据的组织	91
4.3.2 注册应用程序	93
4.3.3 检索扩展实体数据	94
4.3.4 把扩展实体数据附着到实体上	96
4.3.5 扩展实体数据内存管理	96
4.3.6 扩展实体数据中的句柄	97
4.4 Xrecord 对象	97
4.5 获取符号表和词典	98
4.5.1 符号表	98
4.5.2 词典条目	100
第五章 应用 Visual LISP 进行程序开发	102
5.1 系统控制台	102
5.1.1 系统控制台的功能	103
5.1.2 系统控制台的快捷菜单	104
5.1.3 分隔符处理	105
5.1.4 控制台中的彩色代码显示	105
5.1.5 上下文相关的帮助	106
5.1.6 记录控制台事件	106
5.1.7 在 Visual LISP 中执行 AutoLISP 函数	107
5.1.8 在 Visual LISP 和 AutoCAD 间切换	108
5.2 应用文本编辑器	108
5.2.1 文件操作	108
5.2.2 彩色代码显示	111
5.2.3 上下文相关的帮助	112
5.2.4 文本编辑器的快捷菜单	113
5.2.5 字段处理	114
5.2.6 快捷功能	114

5.3 格式化代码	124
5.3.1 运行格式化器	124
5.3.2 Visual LISP 格式化基础	125
5.4 语法错误检查	133
5.4.1 检查括号是否匹配	133
5.4.2 应用彩色代码功能检查语法错误	134
5.4.3 用检查命令来查找语法错误	135
5.5 运行程序	136
第六章 调试程序	137
6.1 在 Visual LISP 中进行调试	137
6.1.1 设置断点	140
6.1.2 调试程序	141
6.1.3 跟踪表达式的运行结果	142
6.1.4 在程序运行时跟踪变量	143
6.1.5 继续运行程序	144
6.2 调试特征	144
6.2.1 开始调试	145
6.2.2 断环	145
6.2.3 断点	147
6.3 Visual LISP 数据检验工具	149
6.3.1 监视窗口	150
6.3.2 跟踪堆栈窗口	151
6.3.3 符号服务	158
6.3.4 检验器	159
6.3.5 查看 AutoCAD 图形实体	165
第七章 构造应用程序	168
7.1 编译和连接程序	168
7.1.1 如何应用编译器	169
7.1.2 编译一个程序文件	169
7.1.3 编译举例	170
7.1.4 加载并运行编译过的程序	171
7.1.5 连接函数调用	172
7.2 构造独立的应用程序	172
7.2.1 在 AutoCAD 中声明函数	173
7.2.2 Visual LISP 运行时系统	175

7.3 生成应用程序模块	179
7.3.1 应用程序向导	180
7.3.2 加载和运行应用程序	189
7.3.3 重新构造应用程序	190
第八章 维护 Visual LISP 应用程序	191
8.1 管理多个 Visual LISP 文件	191
8.1.1 Visual LISP 工程介绍	191
8.1.2 定义一个工程	193
8.1.3 设置工程属性	194
8.1.4 工程窗口	197
8.1.5 工程操作	201
8.2 优化应用程序	204
8.2.1 定义构造选项	204
8.2.2 选择编译模式	205
8.2.3 理解工程构造选项	206
第九章 在 Visual LISP 中应用 ActiveX 对象	210
9.1 AutoCAD 对象模型	213
9.1.1 对象属性	214
9.1.2 对象功能	215
9.1.3 对象集合	215
9.2 获取 AutoCAD 对象	215
9.2.1 用检验工具来查看对象属性	216
9.2.2 从 AutoCAD 应用程序对象中获取其他对象	217
9.2.3 过程概要	218
9.3 应用 Visual LISP 函数实现 ActiveX 功能	219
9.3.1 确定所需的 Visual LISP 函数	219
9.3.2 确定如何调用一个函数	220
9.3.3 转换 Visual Basic 参数到 Visual LISP 函数中	221
9.3.4 观察和更新对象属性	223
9.3.5 判断是否可以获取一个对象	225
9.3.6 在形式参数中返回数值	226
9.4 显示一个对象的属性和功能	226
9.5 对象集合操作	228
9.6 参数转换	231
9.7 释放对象和内存	231

9.8 转换对象引用	232
第十章 向 AutoCAD 图形附着反应器	234
10.1 反应器类型和事件	234
10.2 定义回调函数	238
10.2.1 定义链接、编辑和数据库反应器	238
10.2.2 定义对象反应器回调函数	238
10.2.3 应用预定义的回调函数	238
10.3 生成反应器	239
10.3.1 应用对象反应器	240
10.3.2 在一个实体上附着反应器	242
10.4 查询、修改和删除反应器	242
10.4.1 检验反应器	243
10.4.2 查询反应器	244
10.4.3 修改反应器	244
10.4.4 禁用反应器	245
10.5 临时和永久的反应器	246
第十一章 Visual LISP 函数参考	248
附录 错误代码	390

Visual LISP 简介

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。本章着重介绍 Visual LISP 新的集成开发环境和新增功能。

本章将要介绍的主要内容有:

- Visual LISP 与 AutoLISP 的关系。
- Visual LISP 的新功能。
- Visual LISP 软件安装。
- 如何启动 Visual LISP。
- 用户界面概述。
- 菜单概述。
- 控制台概述。
- 文本编辑器概述。
- 加载和运行 AutoLISP 程序。

1.1 AutoLISP 和 Visual LISP 的关系

AutoLISP 是一种用来扩展 AutoCAD 功能、定制 AutoCAD 和在 AutoCAD 上做开发的编程语言,它是基于 LISP 语言发展起来的。LISP 语言是在 20 世纪 50 年代末期诞生的,最初被设计用在人工智能(AI)应用程序中,而且现在也仍然是人工智能应用程序的基础。

20 世纪 80 年代中期,AutoCAD 在其 2.1 版本中引入了 AutoLISP 作为应用程序编程接口。之所以最初选择 LISP 作为二次开发 AutoCAD 的编程语言,主要是因为它是唯一适合于进行 CAD 项目开发的非结构化设计的语言。

Visual LISP(VLISP)是一个用来帮助用 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特征可以帮助简化诸如源代码生成及修改、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。

在过去,用 AutoLISP 语言开发基于 AutoCAD 的应用程序意味着要在一个文本编辑器(用户必须自己提供,如 DOS 的 edit 和 Windows 的 Notepad 等)中输入程序代码,然后再把程序加载进 AutoCAD 并运行。要想进行软件调试,只能在源程序中加入一些语句来打印出变量的内容,并以此作为调试点,另外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,就不得不再返回源程序中并加入更多的调试点。最后,当程序已经工作得很好时,还需要再把那些调试代码删除或注释掉。而 Visual LISP 简化了这一切过程。

下面让我们来看看 Visual LISP 提供了哪些功能。

1.2 Visual LISP 提供的新功能

Visual LISP 提供了以下新功能:

- 完整的 AutoLISP 仿真环境,具有透明的加载时编译技术。
- 强大的语法检查器,可以识别出 AutoLISP 语法错误及调用 AutoLISP 函数时不正确的参数输入。
- 文件编译器,可以改善程序的运行速度和提供安全、高效的交付平台。
- 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,可以为调试提供最大的灵活性。这个特征也支持在一个窗口中单步运行 AutoLISP 源程序,而且同时在 AutoCAD 的图形窗口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。
- AutoLISP 格式化器,可以改变程序结构,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。
- 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。它包含着一套自己的窗口和菜单,这些窗口及菜单和 AutoCAD 是显著不同的。但是,Visual LISP 是不能独立于 AutoCAD 运行的,无论什么时候想要在 Visual LISP 中工作,都首先要将 AutoCAD 运行起来。并且有时当从 Visual LISP 环境运行 AutoLISP 程序的时候,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。

1.3 安装 Visual LISP

在安装向导的提示下安装 Visual LISP 是很简单的,安装程序会自动查找到 AutoCAD 的安装位置,并把 Visual LISP 的默认安装位置设置为 AutoCAD 安装目录下的 VLISP 目录。安装完成后不用重新启动计算机,再一次进入 AutoCAD 后,按本书 1.4 节所述方法即可启动 Visual LISP。

整个程序的安装过程基本上分以下几个步骤:

① 确定程序安装位置。建议首先安装 AutoCAD,再安装 Visual LISP,这样 Visual LISP 的安装向导就会自动寻找 AutoCAD 的安装目录,并以此目录下的 VLISP 作为默认的安装目录,如图 1.1 所示。

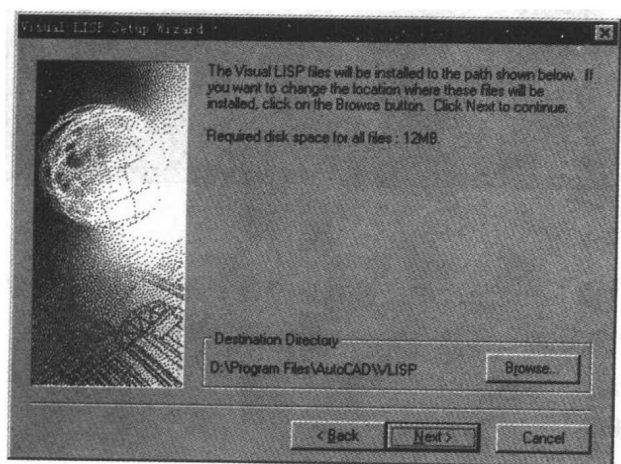


图 1.1

② 指定程序组名。虽然 Visual LISP 不能像其他 Windows 的应用程序一样独立运行,但安装向导还会为 Visual LISP 在“开始”菜单中建立一程序组,以供用户获取帮助信息和卸载程序文件,如图 1.2 所示。

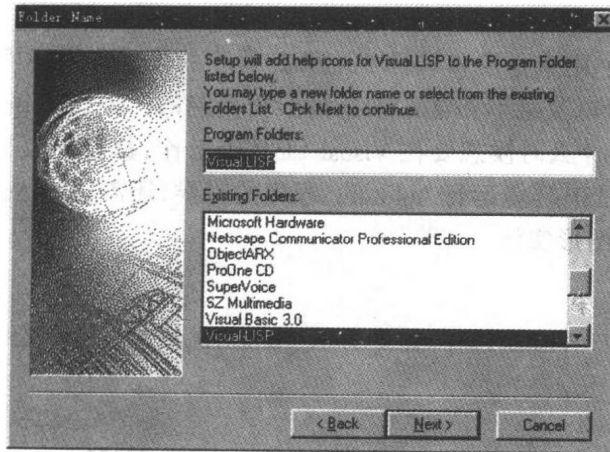


图 1.2

③ 开始安装。在设置完安装路径和程序组名后,安装向导会再一次让用户确认是否接受所做的设置,如图 1.3 所示。单击 Next 按钮开始复制文件。

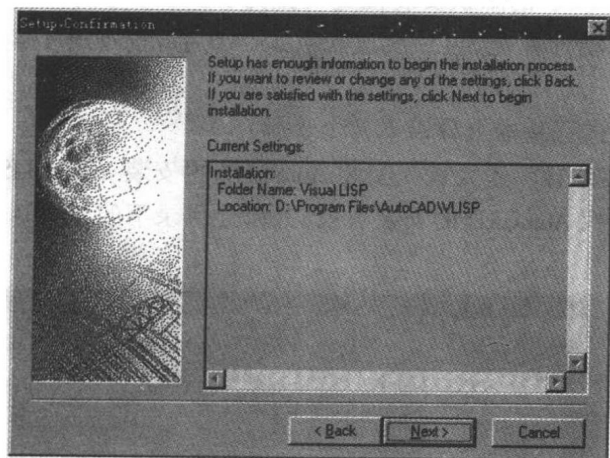


图 1.3

④ 所有程序文件安装完成后,用户可以选择是否阅读 Visual LISP 的自述文件,如图 1.4 所示。选中图中的复选框即可阅读 Visual LISP 自述文件。

⑤ 至此已经完成了 Visual LISP 的安装,再一次进入 AutoCAD 后即可应用下一节讲述的方法来启动 Visual LISP。

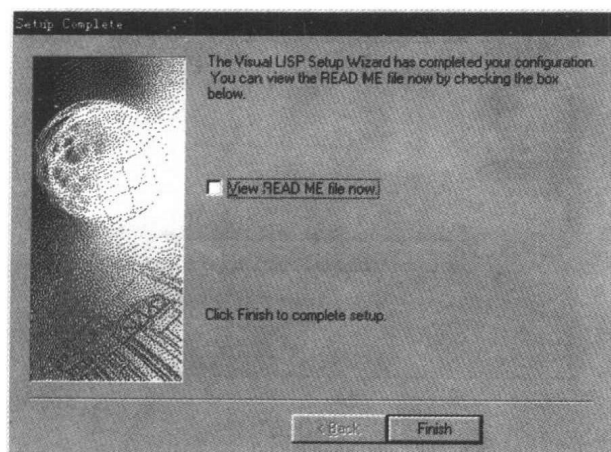


图 1.4

1.4 启动 Visual LISP

在开始启动 Visual LISP 之前,要注意 Visual LISP 程序被安装在计算机的哪个目录里。默认情况下是安装在 AutoCAD 安装目录下的 VLISP 子目录下。

Visual LISP(VLISP)是作为一个 AutoCAD 的 ObjectARX 应用程序运行的。

要应用 Visual LISP,必须先启动 AutoCAD,然后按下面步骤进行:

① 在 AutoCAD 的菜单里,选择 Tools→Load Application 命令,弹出“Load AutoLISP, ADS, and ARX Files”对话框,如图 1.5 所示。

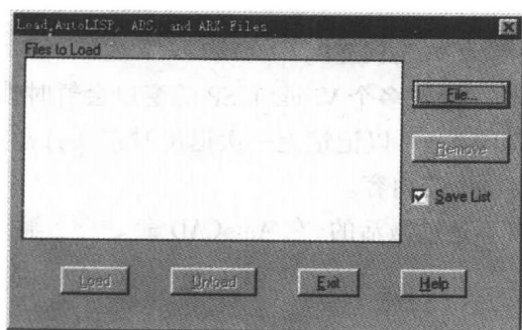


图 1.5

② 在图 1.5 中单击 File 按钮,弹出“Select AutoLISP, ADS, or ARX File”对话框,如图 1.6 所示。

③ 在该对话框中,改变目录到包含 Visual LISP 程序文件的文件夹,找到 Vlisp.arx 文件,并双击该文件名。在默认情况下,选择文件的对话框里只会显示 LSP 文件,而不会显示 Vlisp.arx 文件,这时应该在“文件类型”下拉列表中选择 ARX。