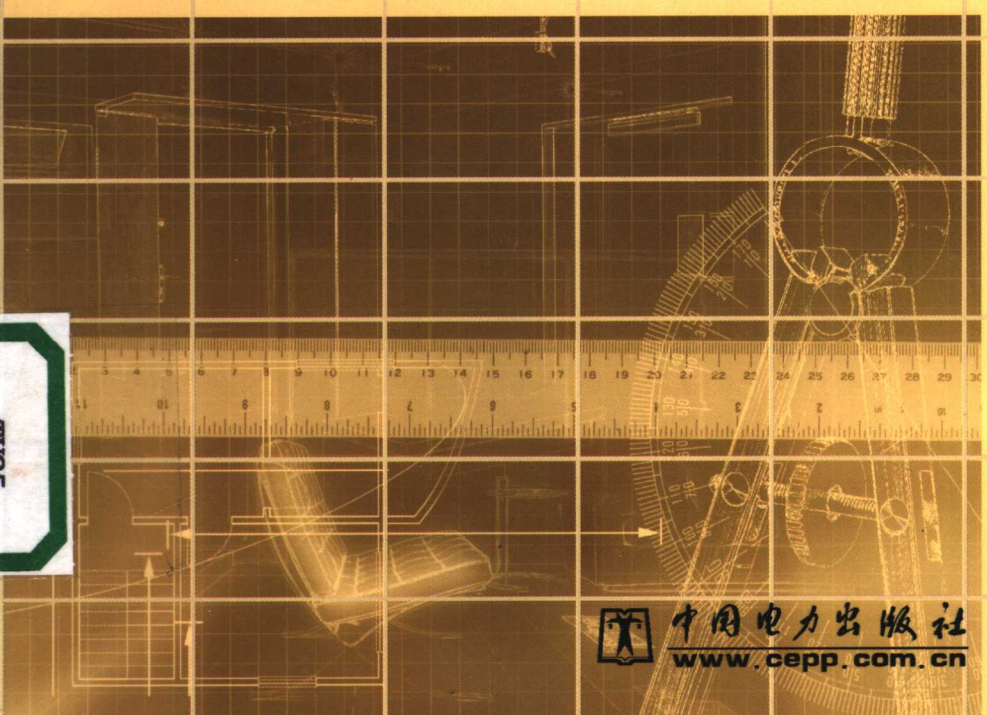
王　佳　主编

建筑电气CAD



中国电力出版社
www.cepp.com.cn

建筑电气CAD

王　佳　主编

中国电力出版社
www.cepp.com.cn

内容提要

本书是一本以 AutoCAD 为蓝本，结合建筑电气工程实例，面向 CAD 初学者编写的图书。

全书包括八章内容，分别是：AutoCAD 程序概论、照明和动力电气工程图的绘制、建筑防雷与接地电气工程图的绘制、弱电系统图及电子线路图的绘制、变配电所电气工程图的绘制、建筑设备电气控制工程图的绘制、建筑电气计算以及设置电气设计软件。

本书适合作为建筑类高等院校的 CAD 教材，也可作为从事电气工程设计人员的参考书，还可供电类在职职工岗位培训、社会培训或自学使用。

图书在版编目（CIP）数据

建筑电气 CAD / 王佳主编. —北京：中国电力出版社，2005
ISBN 7-5083-2646-6

Ⅰ. 建… Ⅱ. 王… Ⅲ. 房屋建筑设备：电气设备-建筑设计：计算机辅助设计-应用软件，AutoCAD Ⅳ. TU85-39

中国版本图书馆 CIP 数据核字（2004）第 135948 号

中国电力出版社出版、发行
（北京三里河路 6 号 100044 http://www.cepp.com.cn）
航远印刷有限公司印刷
各地新华书店经售
*
2005 年 2 月第一版 2005 年 2 月北京第一次印刷
787 毫米×1092 毫米 16 开本 9.5 印张 230 千字
印数 0001—4000 册 定价 **16.00** 元

前言

图纸是工程师的语言，计算机辅助设计和绘图技术已成为设计人员的必备技能之一，本书的作者都是讲授建筑电气设计和 CAD 技术相关课程多年的资深教师和专业权威人士，他们从工程设计的操作实际出发，结合当今最为流行的电气设计软件和典型范例，教授您建筑电气 CAD 辅助设计和绘图的专业技能，通俗易懂，简单明了，旨在提高您的专业竞争力，帮助您提高工作效率，成为电气设计和绘图的职业高手，从而在充满机会的职场中，获得最佳的位置。同时，本书也是大、专院校学生学习建筑电气 CAD 技术的极好教材，可以使学生通过轻松快捷的方式很快地掌握这项技术，为他们进入设计院从事电气设计工作铺平了道路。

全书共分为 8 章，第 1 章介绍了 AutoCAD 程序的起源、发展和简单进行二次开发的知识；第 2 章是全书的重点，详细地介绍了绘制照明和动力平面图和系统图的方法，大量的图片使您轻而易举地就可以按步操作，翔实的操作说明，使学习过程更加轻松有效；第 3 章教您如何绘制建筑防雷与接地电气工程图，还告诉您如何通过立体演示直观地检测避雷区域的方法；第 4 章讲授的是绘制弱电电气工程图的方法，图文并茂地教授了消防和电视电话平面系统图的绘制方法；第 5 章是变配电所电气工程图的绘制方法和步骤；第 6 章介绍了绘制建筑设备电气控制工程图的方法，第 7 章是关于建筑电气计算的内容，详细介绍了通过专业的电气设计软件进行照度计算、负荷计算、电压损失计算和短路电流计算的方法；最后一章还介绍了设置电气设计软件的方法。全书的内容基本上涉及到了建筑电气设计中方方面面的内容，极其全面，是您进入建筑电气 CAD 技术领域的良师益友。

本书由王佳主编，第 1 章由刘洋编写，第 2、3、6、7、8 章由王佳、沈睿编写，第 4 章由张少军编写，第 5 章由何伟良编写。由于时间仓促，书中不足之处，敬请读者批评指正。

编　者

2004 年 12 月

目录

AutoCAD程序概论

1.1 AutoCAD 程序的起源及发展

AutoCAD 是由美国 Autodesk 公司于 20 世纪 80 年代初为微机上应用 CAD 技术而开发的绘图程序软件包。经过不断的完善，AutoCAD 现已经成为国际上流行的绘图工具。

AutoCAD 可以绘制任意二维和三维图形。同传统的手工绘图相比，用 AutoCAD 绘图速度更快、精度更高，而且便于个性，因此，它已经在航空航天、造船、建筑、机械、电子、化工、美工及轻纺等很多领域得到了广泛应用，并取得了丰硕的成果和巨大的经济效益。

AutoCAD 具有良好的用户界面。通过交互菜单或命令行方式便可以进行各种操作。它的多文档设计环境让非计算机专业人员也能很快地学会使用，并且能在不断实践的过程中更好地掌握它的各种应用和开发技巧，不断提高工作效率。

AutoCAD 具有广泛的适应性。它可以在各种操作系统支持的微型计算机和工作站上运行，不仅支持 40 多种图形显示设备，分辨率可由 320×200 到 2048×1024，还支持 30 多种数字仪和鼠标器，数十种绘图仪和打印机，为 AutoCAD 的普及创造了条件。

AutoCAD 的发展过程可分为初级阶段、发展阶段、高级发展阶段、完善阶段及进一步完善阶段五个阶段。

（1）初级阶段。在初级阶段里 AutoCAD 更新了五个版本，它们分别是：于 1982 年 11 月首次推出的 AutoCAD 1.0 版本；于 1983 年 4 月推出的 AutoCAD 1.2 版本；于 1983 年 8 月推出的 AutoCAD 1.3 版本；于 1983 年 10 月推出的 AutoCAD 1.4 版本；以及于 1984 年 10 月推出的 AutoCAD 2.0 版本。

（2）发展阶段。在发展阶段里，AutoCAD 更新了以下版本：于 1985 年 5 月推出的 AutoCAD 2.17 版本和 2.18 版本；于 1986 年 6 月推出的 AutoCAD 2.5 版本；以及于 1987 年 9 月后陆续推出的 AutoCAD 9.0 版本和 9.03 版本。

（3）高级发展阶段。在高级发展阶段里，AutoCAD 经历了三个版本，使 AutoCAD 的高级协助设计功能逐步完善。它们是 1988 年 8 月推出的 AutoCAD 10.0 版本、1990 年推出的 11.0 版本和 1992 年推出的 12.0 版本。

（4）完善阶段。在完善阶段中，AutoCAD 经历了三个版本，逐步由 DOS 平台转向 Windows 平台。1996 年 6 月，AutoCAD R13 版本问世；1998 年 1 月，推出了划时代的 AutoCAD R14 版本；1999 年 1 月，AutoCAD 公司推出了 AutoCAD 2000 版本。

（5）进一步完善阶段。在进一步完善阶段中，AutoCAD 经历了两个版本，功能逐渐加

强。2001 年 9 月 Autodesk 公司向用户发布了 AutoCAD 2002 版本。2003 年 5 月，Autodesk 公司在北京正式宣布推出其 AutoCAD 软件的划时代版本——AutoCAD 2004 简体中文版。

AutoCAD 有几种编程接口，最原始的一种是 AutoLISP。它是一种解释性的编程语言，最初出现于 1985 年发行的 AutoCAD 2.5 版中。4 年以后，在 AutoCAD R10 中增加了称为 ADS 的 C 语言编程能力。然后，在 AutoCAD R13 中又增添了 ARX(AutoCAD 运行扩展)编程接口，它是新一代的基于 C++的应用程序接口，可以为应用程序扩展 AutoCAD 的功能提供前所未有的能力。AutoCAD R13 中使用的 ARX 是 1996 年 1 月发行的 1.1 版。在 AutoCAD R14 中，通过新的 API 和其他改进，ARX 的功能又有了新的扩充，已经被重新命名为 ObjectARX。ObjectARX 接口的功能十分强大，AutoCAD R14 自身的很大一部分就是用 ObjectARX 实现的。例如，处理光栅图像的子系统就几乎没有向 AutoCAD 核心系统增添什么新代码。其结果是，ObjectARX 使得 AutoCAD 成为了一个更加模块化的系统。ObjectARX 并没有取代 LISP 和 ADS，在 AutoCAD R14 中，LISP 和 ADS 仍然存在并有所扩展，并且 AutoCAD R14 中的 ObjectARX 与 AutoCAD R13 中的 ARX 具有高度的向上兼容关系。AutoCAD R13 上的 ARX 程序只需要重新编译就可以在 AutoCAD R14 上运行。

1.2 AutoCAD 开发程序的历史（AutoCAD 的定制开发）

AutoCAD 是一个为通用绘图系统而设计的软件。但每个行业和专业都有自己的行业和专业标准，用户也会有自己独特的工作方式，因而，AutoCAD 不可能完全满足每个用户的具体要求。于是通过系统的开放式体系结构，AutoCAD 允许用户和第三方软件开发商根据各自的需求改进和扩充 AutoCAD 的许多功能，实现对 AutoCAD 的定制和开发，使 AutoCAD 更加符合用户的需求。

1.2.1 开发的必要性

AutoCAD 是目前 Windows95/98/NT/2000 环境下应用最广泛、使用人数最多的 CAD 软件。但是 AutoCAD 所提供的只是一般的通用的 CAD 功能，如造型、编辑、注释等。如果使用 AutoCAD 开发系统，则我们可以将计算和绘图通过高级语言编写相应的程序，在需要设计时，只一个命令便可以运行该程序，计算和绘图过程自动完成。显而易见，这不仅大大提高了设计效率，而且通过开发系统可以定制出某些专业化模块，甚至大型设计绘图软件。国内的建筑行业天正 CAD 系统、机械行业的大恒 CAD 系统等，均是对 AutoCAD 开发定制而实现的。

1.2.2 定制开发的工具

AutoCAD 系统的定制开发工具又称为 AutoCAD 开发工具，有时称作 AutoCAD API（Application Programming Interface 应用编程接口）。它是将 AutoCAD 环境客户化的基本手段。在 AutoCAD2002 中，我们使用的开发工具主要有 ObjectARX、AutoLISP、Visual LISP、Java、VisualBASIC 及 Delphi 等。

1.2.3 定制开发的主要内容和方法

AutoCAD2002 定制开发的内容很多，但最主要的是脚本文件（SCR 文件）定制、程序参数文件（ACAD、PGP）定制、菜单文件定制及 AutoCAD 命令的定制。

（1）脚本文件（SCR 文件）定制。AutoCAD 提供了一个叫 Script File（脚本文件）的工

具，它允许不同的 AutoCAD 命令组合起来，并按照预先确定的顺序执行。这些命令可以用任何一种文字编辑器（如 Notepade、记事本等）编写成文本文件，其扩展名为.SCR（如 PLOT1.SCR）。脚本文件用 AutoCAD 中的 SCR 命令来执行。

用户在对 AutoCAD 进行开发时，可以使用任何一种高级语言（如 VC、VB、Delphi、Java 等）设计用户交互界面，对绘图所需要参数进行计算生成，然后确定 AutoCAD 命令、命令选项、命令序列等，最后生成扩展名为 SCR 的 AutoCAD 脚本文件。在 AutoCAD 中用 SCR 命令来执行脚本文件，完成所需操作。脚本文件具有 AutoCAD 的所有功能，包括图形初始化、编辑、修改、绘图、输出等。但脚本文件有一些限制，它不能使用对话框和菜单，当从脚本文件中发出打开文件、文件存盘、打印等命令时，AutoCAD 执行命令行中的命令含义而不打开对话框，无法提供用户交互功能。

运用脚本文件定制 AutoCAD，只需要熟悉一门高级编程语言和 AutoCAD 常用命令、命令选项及命令序列即可。此方法，对编程技术要求不高，简单实用。它曾经作为最原始的定制开发手段，一直保留至今。实践证明，它可以取得很好的令人满意的效果。目前，仍有许多应用软件还在使用这项技术。

（2）程序参数文件（ACAD.PGP）文件定制。AutoCAD 软件带有程序参数文件 ACAD.PGP，该文件分为注释、外部命令及命令别名三个部分。

1）注释。文件中的注释可以包含任何数目的注释行，并且可以出现在文件任何地方。每一个注释行用（;）开头。任何用分号开头的句子在执行时都被忽略。

2）外部命令。ACAD.PGP 允许用户从图形编辑器中直接使用操作系统命令。例如：想要删除一个文件，只需要在命令行输入 DEL (COMMAND: DEL)，然后根据提示输入需要删除的文件名。

3）命令别名。ACAD.PGP 定义了一些 AutoCAD 命令的别名。例如，LINE 命令的别名是 L。如果在命令行输入 L (COMMAND: L)，AutoCAD 将把它当作 LINE 命令。用户可根据自己使用需要定制 ACAD.PGP 文件，扩充操作系统命令和 AutoCAD 命令别名。

（3）菜单文件定制。AutoCAD 软件提供了一个定制 AutoCAD 的强大工具。AutoCAD 软件带有一个名为 ACDA.MNU 的标准菜单文件。在启动 AutoCAD 时，ACDA.MNU 菜单文件自动装载。菜单文件中包含 AutoCAD 命令，用户可以修改、排列这些常用命令。它允许用户删除不常用的命令，并定义新的命令，通过编辑 ACDA.MNU 菜单文件或者编写新的菜单文件来实现。可以为每一个应用程序编写一个单独的菜单文件。使用 AutoCAD 的 MENU 命令，可以随时装载这些菜单。

菜单文件的扩展名为 MNU，可以用任何文本编辑器进行编辑。菜单文件中可以定制屏幕菜单、下拉菜单、工具条、快捷菜单、上下文菜单、图像菜单等内容。

用户开发过程中，可以参照标准菜单文件 ACAD.MNU 编写自己的专用菜单文件。

（4）AutoCAD 命令的定制。AutoCAD 命令的定制即根据用户特定专业领域的问题，采用开发工具，通过编程定制自己的 AutoCAD 专用功能模块。

在各种开发工具中，直接与 AutoCAD 通信的 API 比利用 IPC 通信的 API 在速度上要快。因此，ObjectARX 的速度最快，AutoLISP 速度最慢。但是在程序稳定性上，采用 AutoLISP 开发的应用程序一旦失败，并不危害 AutoCAD 自身进程。而由于 ObjectARX 应用程序共享的 AutoCAD 地址空间，一旦失败，AutoCAD 进程也随之崩溃。在技术难度上，AutoLISP

和 VisualBASIC 均为解释型语言，方便易学，开发周期短。所以下面章节主要介绍这两种编程语言。

1.3 AutoLISP 编程简介

1.3.1 Visual LISP 入门

AutoLISP 是为了扩展和自定义 AutoCAD 功能而设计的一种程序语言，在学习 AutoLISP 之前，我们需要对开发程序的环境有一些了解，即了解 Visual LISP 的集成开发环境（IDE），并告诉您如何在 Visual LISP 中运行 AutoLISP。

尽管 Visual LISP 的运行需要 AutoCAD 的支持，但是 Visual LISP 交互式环境并不与 AutoCAD 其他部分在同一个窗口集中运行，必须通过与行命令启动 Visual LISP，才能在他的交互式开发环境中工作。

启动 AutoCAD 后选择“工具”菜单，“AutoLISP”子菜单上的“Visual LISP 编辑器”命令，即可启动 Visual LISP 编辑器。

Visual LISP 有标准、搜索、视图及工具等工具栏，各自代表不同功能的 Visual LISP 命令组。如果将鼠标在工具栏的某按钮上停留几秒钟，Visual LISP 将会在屏幕底部的状态栏上显示更相信的描述。

位于屏幕底部的状态栏中显示的信息会因 Visual LISP 中所作的工作不同而异。其同时会看到一个最小化的跟踪窗口(Trace)。该窗口会包含 Visual LISP 当前版本的信息，如果 Visual LISP 启动遇到错误，他还会包含相应的错误信息。

“Visual LISP 控制台”窗口是 Visual LISP 主窗口的一个独立的可滚动窗口。在“Visual LISP 控制台”窗口可以输入 AutoLISP 命令，方式与在 AutoCAD 命令行中类似；也可以不用菜单或工具栏而直接在控制台窗口发出 Visual LISP 命令。

在 Visual LISP 控制台的窗口中输入命令或者运行从文本编辑器中加载的程序时，可能需要在 Visual LISP 和 AutoCAD 窗口间切换，可以通过按下“视图”工具栏中“激活 AutoCAD”按钮来激活 AutoCAD；如果在 AutoCAD 中想返回 Visual LISP 环境，则可以在命令提示下输入 VLSIP 命令。

在完成 Visual LISP 任务后，可以选择“文件”菜单中“退出”命令或单击标题栏的“关闭”来关闭程序，这时 AutoCAD 并没有完全卸载 Visual LISP，而是存了退出时的状态，在下一次启动 Visual LISP 任务时会自动打开上次退出时的文件和窗口。

1.3.2 AutoLISP 语言基础

要编写完整的应用程序，首先要定义完成怎样的工作，然后利用 Visual LISP 开发环境创建 LISP 文件，最后为这个应用程序编写 AutoLISP 代码。

LISP 程序是一种计算机的表处理语言，是人工智能领域中广泛应用的一种程序设计语言。开发 AutoLISP 程序的出发点就是为了实现某些 AutoCAD 操作的自动化，加快重复性绘图工作的步伐，简化一系列复杂操作。

AutoLISP 程序由一系列表达式组成，AutoLISP 表达式的格式为“(函数 参数)”，每个表达式都以一个左括号开始，由一个函数名和一个该函数的可选参数组成，并且每个参数都可以是一个表达式，表达式以右括号结束。例如

（fun1(fun2 参数)（fun3 参数））

该代码样例调用了三个函数：第一个函数 fun1 有两个参数，另两个函数 fun2 和 fun3 各有一个参数。函数 fun2 和 fun3 被函数 fun1 所包涵，因此他们的返回值作为参数传递给 fun1，也就是说，嵌套在其他表达式中的表达式将他们的结果返回给外层表达式。

如果输入的右括号数量不对，AutoLISP 将显示提示。提示中左括号的数目表明有多少层左括号没有闭合。如果出现此提示，用户必须输入所需数目的右括号才能对表达式求值。

除了表达式，Auto LISP 的另一个基础内容就是基本输出函数，AutoLISP 包含 AutoCAD 显示的函数，包括文本和图形窗口。某些函数可以在 Visual LISP 控制台窗口中显示信息。如果定义了名称格式为 C：XXX 的 AutoLISP 函数，则该函数可以像内置的 AutoCAD 的命令一样在 AutoCAD 命令提示行中提示使用。无论是在 Visual LISP 中还是在 AutoCAD 命令提示下定义和加载此函数，情况都是一样。

1.3.3 使用局部变量和关联表

从这部分起将深入到 AutoLISP 语言的内部，比如一些有用的可以加快 AutoLISP 程序开发速度的 Visual LISP 调试工具，还将介绍局部和全局变量的不同之处，以及在何时使用它们。同时还将使用 AutoLISP 的表来管理数据。

首先介绍 AutoLISP 中的局部变量和全局变量，并介绍如何使用局部变量和全局文档变量。在文档中加载的所有函数都能访问的变量是全局变量，这些变量再定义他们的程序结束以后还会保持它们的值。

局部变量值在定他们的函数运行时才会有它们的值。函数运行完毕局部变量的值会被自动放弃，同时系统还会释放变量使用的内存空间。这被称为“自动无用数据收集”，大多数 LISP 开发环境都会提供该功能。

有时用户只修改一部分内容，想要专门对修改的部分进行调试。在 Visual LISP 中可以进行程序的局部调试。在 Visual LISP 控制台窗口输入需要调试函数全部代码，将控制台窗口光标放在代码块的最后一个括号，然后按回车键。在控制台输入函数，然后执行。当运行该函数时，Visual LISP 会自动把控制传给 AutoCAD，最后函数会在控制台窗口显示返回值的表。

如果想查看已设置的变量值，可以在控制台窗口提示出输入其名称。

1.3.4 通过监视和断点检查程序

任何程序不可能一次编写成功，一个程序的大部分时间可能都是花在调试上。因此，要判断一个程序语言以及其开发环境的优劣，就要看他是否具有强大的调试功能。Visual LISP 便为用户提供了用于编写和调试程序的全套工具。

（1）监视工具。监视工具是在其中最有价值的调试工具之一。监视工具可以让用户察看变量的信息，远比在 Visual LISP 控制台窗口显示的信息更详细。用户还可以在函数执行时间是函数的局部变量。选择“调试”菜单中的“添加监视”命令，Visual LISP 将显示一个标题为“添加监视”的对话框。输入所要监视的变量名，Visual LISP 会显示一个监视窗口，此时这个变量还没有值，需要将所要调试的函数代码复制到控制台窗口输入（函数名）并运行该函数，给变量赋值。监视窗口中会将变量值显示在一行中。双击监视窗口中的变量名，可打开一个检验窗口，检验窗口指出用户所检查变量的数据类型，以及改变量的值。如果变量是表，将检验每行显示表的一个条目。

（2）调试工具。下面介绍 Visual LISP 提供的调试工具栏，如图 1-1 所示。调试工具栏中包含了几个工具，但大多数在平时都是禁用的，只有在调试模式下运行是他们才会变为可用状态。调试工具栏的按钮分为三组。前面，三个按钮用来单步执行程序代码；中间的三个按钮用来确定断点处或因错误而暂停时，下一步该如何继续。后面的三个按钮用来设置或删除断点、添加监视以及转跳到程序代码中的最近一次运行暂停处。调试工具栏上的最后一个按钮是一个单步调试指示器，它不执行任何操作，但提供了一个可视化的指示器，用于在单步执行代码时指示光标的位置。不是运行在调试模式下时，该按钮显示为空白。

图 1-1 调试工具栏

断点是放在源代码中的一种符号，它用来制定程序暂停运行的位置。当运行代码时，Visual LISP 会正常地运行代码，直到碰到断点为止。在断点处，Visual LISP 会暂停程序的运行，等待指令来决定下一步的工作。在指定了断点之后，Visual LISP 会暂停程序的执行，并将焦点返回到文本编辑的窗口，亮显断点处那行代码。在调试工具栏上，原来空白的“单步调试指示器”图标显示为一个红的光标在一对括号的面前，这表示 Visual LISP 调试器是暂停在表达式之前，如图 1-2 所示。当程序暂停时，可以单步执行代码，一个函数接一个函数或者一个表达式的运行；也可以在任意处使程序恢复正常运行；可以动态修改变量的值，也可以修改正在运行的程序的结果。程序暂停后就可以对程序进行单步调试了。当单步执行程序时，可以将变量添加到监视窗口中，并改变他们的值。如果没有看到监视窗口，只需单击工具栏上的“监视窗口”按钮就可以显示该窗口，如果窗口内仍包括变量，请单击监视窗口顶部“清除窗口”按钮。如果调试的程序运行不正确，可结合使用断点和监视工具来确定变量的值是否和预想的一致。如果变量的值和预想的不一致，可以改变它的值以观察它对程序的影响。选择“调试”菜单的“清除所有断点”按钮，Visual LISP 提示时选择“是”，将删除代码中所有断点。

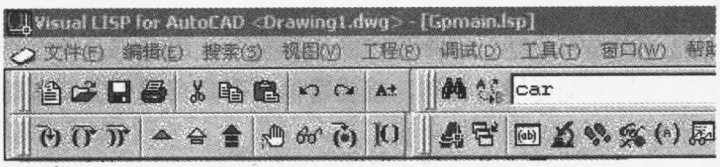


图 1-2 调试工具栏的编辑窗口

1.3.5 编写通用函数

扩展程序可创建一些工具函数，它们并非只能用于该程序，而是通用函数，以后还会用上它们。

编程中一些较为常用的功能会被反复用到。如果能够用函数将这样的功能实现，就能够提高现有的效率。通常在编写较大程序之前，都要编写这个程序的常用工具集。调用函数时，必须给它提供一个函数作为参数。也可将变量传给函数作为参数。

如果在程序中使用该函数，只需将函数定义从控制台窗口复制到 LISP 文件中即可。可以将它粘贴到文件任意位置，只要不将它粘贴到函数中间就行。

1.3.6 分解关联表

将信息从一个函数传给另一个函数，更好的方法是将参数传给被调用的函数，可以在设计函数时就确定要传给它那些参数。通过这种方式可以将变量中的关联表传给函数。

这种方法虽然简单，但还需要解决检验存储在该关联表中的信息问题。Visual LISP 的检验功能可以帮用户确定怎么做。在文本编辑窗口加载代码，在控制台提示处输入画面显示的表达式。

[检验] 窗口只能显示关联表中的所有子表，如果想引用关联表中的单个子表，就要使用 assoc 函数。assoc 函数能够把组码和数据放在一起返回。如果想取得纯粹的数据还要使用函数 cdr。cdr 函数能返回表中除掉第一个元素之外的所有其他元素，这样组码被去掉，数据值被返回。这样参数中的有用数据都被分解出来了，如果把使用中的变量都放在程序中，程序的可读性就要好很多，编译的时候也会节省一点时间。

1.3.7 使用 ActiveX 创建图元

图形是由图元组成的，AutoLISP 语言提供的绘制图元有 ActiveX 函数、entmake 函数及 command 函数。

用 ActiveX 函数创建图元和用 entmake 函数与 command 函数创建图元相比，有两个优点：①用 ActiveX 函数创建图元的速度更快；②ActiveX 函数的函数名本身就说明他们所完成的操作，所以写出的代码可读性更好，更易于维护和修改错误。

AutoCAD 或 Visual LISP 启动时并没有自动加载 ActiveX 功能，所以，如果要使用 ActiveX，就必须确保已加载了 ActiveX。加载 ActiveX 以后，在函数中添加画面显示语句，可以使函数可以调用 ActiveX 方法在 AutoCAD 中显示多段线。

通过 ActiveX 添加图元时，需要制定图元时插在模型空间还是图纸空间。为了告诉 AutoCAD 图元要加入到哪个空间，需要先获取指向该空间的指针。

1.3.8 模块化代码和创建工程

Visual LISP 处理大文件并没有困难，但是如果将文件中定义函数按功能大致分类，然后将文件分拆成几个小文件，将同类函数放置于一个文件中，这样代码维护将变得更容易一些，而且调试也更方便。

将所有程序代码分成多个小文件，要管理这几个文件就要使用 Visual LISP 的工程功能。Visual LISP 的工程功能为管理组成应用程序的多个文件提供了一个方便的途径。使用工程功能，可以不用打开应用程序中的所有文件，只要打开单个工程文件即可。

创建 Visual LISP 工程，选择“工程”菜单上的“新建工程”命令，将工程保存。

将选定的文件添加到工程中，并且通过“底部”、“顶部” 来移动文件位置，Visual LISP 加载工程文件的顺序和文件在列表中的顺序相同，如图 1-3 所示。

选择确定后桌面上会显示一个小的工程窗口，如图 1-4 所示，该窗口列出所有文件，双击任意文件就可以在 Visual LISP 文本编辑其中打开它，并使之成为活动的编辑窗口。

1.3.9 对话框的定义

给应用程序添加对话框界面，还需要使用另一种语言，即对话框控制语言（DCL）。

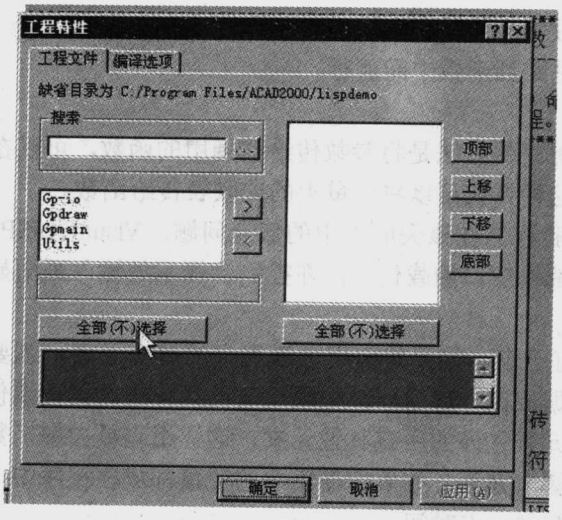


图 1-3 工程特性

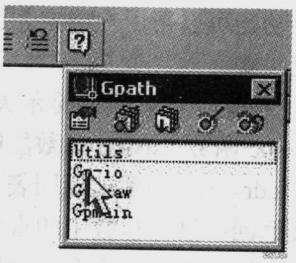


图 1-4 工程窗口

对话框界面的创建可以分为两个步骤：①定义对话框的外观和内容；②添加程序代码，控制对话框的行为。对话框的外观和所包括内容是用 DCL 文件定义的。还需要编写程序代码来初始化对话框的缺省设置，以及定义和用户交互的方法。

在 DCL 中对话框部分间被称为控件，写出一个对话框完整 DCL 文件可能会比较难以理解，可以先规划出想要设计的对话框，将其分成几小块，然后分别写出一小块。

请注意在保存 DCL 文件之前要确保 AutoCAD 运行是能够找到所保存的 DCL 文件，所以文件必须放在 AutoCAD 支持的搜索路径中。如果无法确定这些路径，请选择 AutoCAD “工具”菜单上的“选项”命令。检查“文件”选项卡中的“支持文件搜索路径”参数，如图 1-5 所示。确认文件保存在 AutoCAD 支持搜索路径下，或者添加一个路径作为支持文件搜索路径。

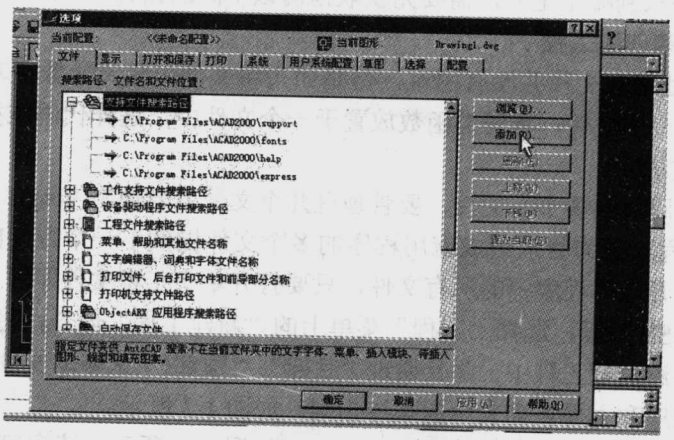


图 1-5 确认文件保存在 AutoCAD 支持搜索路径下

确定工作路径是支持文件搜索路径后，将文件存为“DCL 源文件”保存到 AutoCAD 支持的文件搜索路径。

在 LISP 程序中加载对话框文件用 load_dialog 函数来加载 DCL 文件，对代码中每个

load_dialog 函数，必须有一个 unload_dialog 函数和它相对应。当用户关闭对话框时，还要卸载对话框，并给适当的数据返回给应用程序。

1.3.10 程序集成

最后介绍整体运行和调试工程，然后再认识几种有利于程序和调试的 Visual LISP 工具，最后将已经调试好的工程编译为应用程序，如图 1-6 所示。

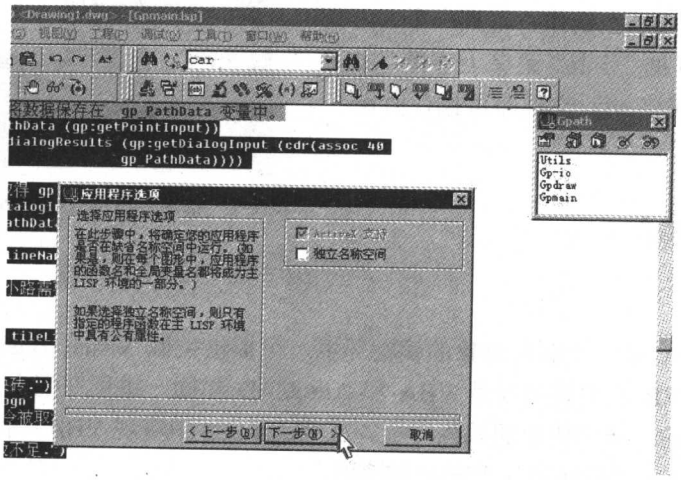


图 1-6 应用程序的调试

在调试过程中，最恼人的事情是程序报告有不匹配的括号却又找不到，这使得程序的调试变得十分不容易，因此 Visual LISP 中提供了括号匹配功能，可以帮助查找与开括号对应的闭括号。Visual LISP 将找到所选的开括号相匹配的闭括号，并选中两者间的代码，这不仅能检查输入的括号数目是否正确，而且还简化了复制或剪切选定文本的工作。

另外 Ctrl+Shift+空格可以直接调用 Visual LISP 的自动匹配能，在［符号服务］对话框（图 1-7）中单击［帮助］按钮，Visual LISP 将提供有关函数的帮助，如图 1-8 所示。

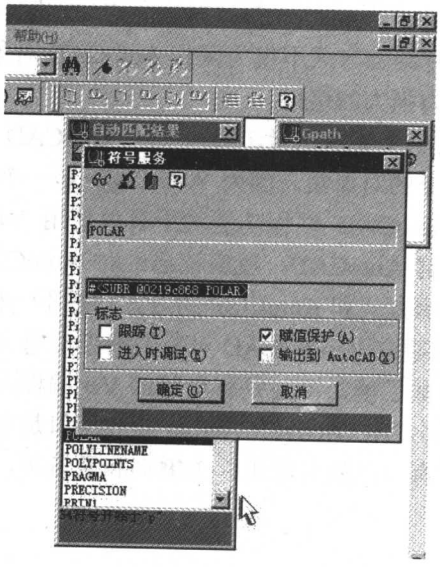


图 1-7 ［符号服务］对话框

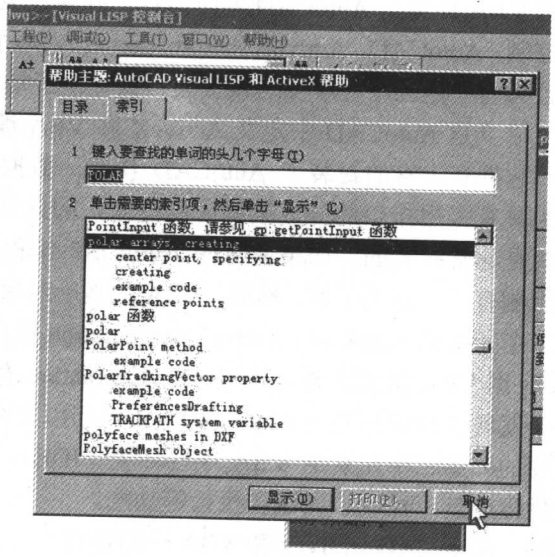


图 1-8 ［帮助］对话框

为了创建独立的应用程序，Visual LISP 提供了［生成应用程序］向导，选择“专家”模式，向导提示指定创建应用程序所需的源文件所在的目录，并提示为应用程序命名。一般只需接受缺省值，如图 1-9 所示，单击下一步，最后检查所作的选择，选择“完成”。Visual LISP 开始生成过程，并在［编译输出］窗口显示编译结果。完成以上操作将得到一个 vlx 文件，在 AutoCAD 的“工具”菜单上选择“加载应用程序”，加载该 vlx 文件即可执行所设计的应用程序。

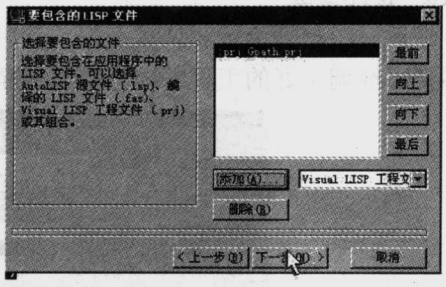


图 1-9 创建应用程序的 LISP 文件添加对话框

1.4 VBA 编程简介

1.4.1 VBA 简介

Microsoft VBA 是一个面向对象的编程环境，可提供类似 Visual Basic（VB）的丰富开发功能。VBA 和 VB 的主要差别是 VBA 和 AutoCAD 在同一进程空间中运行，提供的是具有 AutoCAD 智能的、非常快速的编程环境。在 AutoCAD 中实现 VBA。有以下优点：

（1）Visual Basic 编程环境易于学习和使用。

（2）VBA 可与 AutoCAD 在同一进程空间中运行。这使程序执行得非常快。

（3）对话框的构造快速而有效。这使开发人员可以构造原型应用程序并迅速收到设计的反馈。

（4）工程可以是独立的，也可以嵌入到图形中。这样就为开发人员提供了非常灵活的方式来发布他们的应用程序。

1.4.2 VBA 在 AutoCAD 中的实现方式

VBA 通过 AutoCAD ActiveX Automation 接口将消息发送到 AutoCAD。AutoCAD VBA 允许 VBA 环境与 AutoCAD 同时运行，并通过 ActiveX Automation 接口对 AutoCAD 进行编程控制。AutoCAD、ActiveX Automation 和 VBA 的这种结合方式不仅为操作 AutoCAD 对象，而且为向其他应用程序发送或检索数据提供了功能极为强大的接口。

以下是 AutoCAD 中定义 ActiveX 和 VBA 编程的三个基本元素。第一个是 AutoCAD 本身，它拥有丰富的封装了 AutoCAD 图元、数据和命令的对象集。因为 AutoCAD 是一个设计为具有多层接口的开放架构应用程序，因此熟悉 AutoCAD 编程功能对于有效使用 VBA 来说是非常必要的。如果使用过 AutoLISP 编程来控制 AutoCAD，就应该已经对 AutoCAD 的机制有了一定的了解。然而，VBA 的基于对象的处理方式和 AutoLISP 的方式却很不一样。第二个元素是 AutoCAD ActiveX Automation 接口，它建立与 AutoCAD 对象的消息传递（通信）。用 VBA 编程需要对 ActiveX Automation 有基本的了解。即使是有经验的 VB 编程人员也会发现要理解和开发 AutoCAD VBA 应用程序，AutoCAD ActiveX Automation 接口是非常重要的。第三个元素是 VBA 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。

1.4.3 AutoCAD Activex 和 VBA 一起运行

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。一旦这些对象被显示，许多不同的编程语言和环境以及其他应用程序就可以访问它们。

AutoCAD ActiveX/VBA 接口与其他 AutoCAD API 环境相比，有许多优点：

（1）速度快。当与 VBA 在同一进程空间中运行时，ActiveX 应用程序比 AutoLISP 和 ADS 应用程序运行速度快。

（2）易于使用。其编程语言和开发环境易于使用，而且随 AutoCAD 安装。

（3）具有与 Windows 的互操作性。ActiveX 和 VBA 设计为与其他 Windows 应用程序共同使用，并为应用程序之间的信息交流提供了绝佳的途径。

（4）快速原型。VBA 的快速界面开发为原型应用程序开发提供了一个优良的环境，即使那些应用程序最终将用其他语言开发。

（5）有程序员基础。世界上有数百万的 Visual Basic 程序员。AutoCAD ActiveX 和 VBA 技术为这些程序员以及将来更多的学习 Visual Basic 的人员打开了自定义 AutoCAD 和开发 AutoCAD 应用程序的途径。

对象是所有 ActiveX 应用程序的主要构造块。每一个显示的对象均精确代表一个 AutoCAD 组件。AutoCAD ActiveX 接口中有许多不同类型的对象。例如①直线、圆弧、文字和标注等图形；②线型与标注样式等样式设置；③图层、编组和块等组织结构；④视图与视口等图形显示；⑤图形、AutoCAD 应用程序本身。

从上可以看出 AutoCAD 定制开发的必要性和优越性，用户可以根据自己的情况，选择适合的开发工具，通过对 AutoCAD 的定制开发，开发出属于自己的 AutoCAD 系统，从而使系统更高效运转，发挥更大的经济效益。随着计算机技术的不断发展，技术人员素质的不断提高，AutoCAD 定制开发技术在工程制图中的应用将会得到更快更好的发展。