AutoCAD ADS 程序的开发与应用

焦宗夏　方永宏　编著

机械工业出版社

AutoCAD ADS 程序的开发与应用

焦宗夏　　方永宏　　编著

机械工业出版社

AutoCAD的ADS开发环境是AUTODESK公司推出的与C语言编程的接口环境。由于C语言在编程方面提供了极其丰富的函数库及良好的保护功能，通过C语言调用ADS库函数，不但可以简化程序的书写、增加软件的保护功能，同时还可提高函数的运行速度。ADS开发环境为AutoCAD软件的第三开发方开发出高质量的应用软件提供了方便。

本书介绍了ADS变量、实体、变换等基本概念及初学者如何根据不同版本的C语言设置自己的编译连接环境，同时详细介绍了ADS的库函数，并对所有库函数都给出了使用范例。最后，本书给出了作者近来开发的部分ADS完整的应用程序，包括三维图形装配及图样管理系统等。

本书文字通俗易懂，深入浅出。

图书在版编目(CIP)数据

AutoCAD ADS程序的开发与应用/焦宗夏，方永宏编著.
—北京：机械工业出版社，1996
ISBN 7-111-04887-3

Ⅰ.A… Ⅱ.①焦…②方… Ⅲ.图形软件—软件开发
Ⅳ.TP31
中国版本图书馆CIP数据核字（95）第14751号

出 版 人：马九荣 （北京市百万庄南街1号 邮政编码100037）
责任编辑：温莉芳 版式设计：张世琴 责任校对：韩晶
封面设计：方 芬 责任印制：卢子祥
北京交通印务实业公司印刷·新华书店北京发行所发行
1996年2月第1版第1次印刷
787mm×1092mm 1/16·14.5印张·345千字
0001—3 000册
定价：28.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

前　言

ADS（AutoCAD Development System）是 AutoCAD R11.0 以上版本推出的 C 语言与 AutoCAD 接口开发环境，是 AutoCAD 的一个开放的可扩展的开发系统。ADS 的推出改变了过去的封闭、单一解释式的程序接口形式，给 AutoCAD 应用程序第三开发方带来了新的希望。

自从 ADS 开发系统推出以后，在建筑、机械、电气、地理信息系统、仿真分析等方面均有较高水平的通用应用软件推出，但由于各种原因，ADS 在国内的应用还不是很普遍，没有一本很好的通俗易懂、深入浅出的培训教材。针对这一状况，作者根据多年的编程经验并参考了一些国外有关资料编写了本书，以期对我国 CAD 应用水平的提高尽一份绵薄之力。

本书是为那些熟习 C 语言和 AutoCAD 绘图环境的人员编写的，它包含有 AutoCAD 环境中 ADS 基本概念的介绍及随后的一些 ADS 练习程序，练习从在 AutoCAD Command 命令下打印"Hello，AutoCAD"的简单程序开始，直至介绍一些复杂的操作，如选择集的交换和扩展实体数据的管理，每个练习事先都对其概念和练习中用到的 ADS 函数进行详细的解释。最后给出了大量的由本书作者在工作中所编制的实用程序，如二维图形的布尔运算，三维图形的装配及图样管理系统等，读者从中可以有所借鉴和启发。

作　者

1995 年 5 月

目　录

此为试读，需要完整PDF请访问：www.ertongbook.com

1 概　述

1.1 ADS、EED的基本概念、发展及演变

AutoCAD R11 和R12 的新特点是 ADS 和 EED(Extended Entity Data)，ADS 允许外部可执行程序以最紧密的形式在 AutoCAD 中运行；EED 则几乎可以将任何数据加入到 AutoCAD 的实体中去。

ADS 最早是在 OS/2 上的 AutoCAD R10 上引入的，到 AutoCAD R12，ADS 已经成为所有工作平台上的一个标准属性。传统的 AutoCAD 编程语言是 LISP 语言，它有优点也有缺点。缺点是从 LISP 程序语言中继承下来的，在相同的规则下需要很多的括号，所以 LISP 语言通常被称为很愚蠢的括号语言；但其优点是具有很多方便使用的地方，它能非常好地嵌入到 AutoCAD 中。在 AutoCAD Command 下敲入任何 AutoLISP 语句，马上就可以看到其执行结果，它是一种 read-eval-prompt 的循环过程。ADS 是 AutoLISP 的一种补充和加强。

事实上，所有的 ADS 应用函数都是由 AutoLISP 内部处理，并称之为"外部 AutoLISP 函数"。那么 ADS 到底是什么呢？

1) ADS 是一种接口。ADS 是一种应用程序接口，允许执行由 C 和 C++开发并通过各种标准编译来实现的二进制文件，并在执行期间与 AutoCAD 绘图环境通讯。

2) ADS 是一种语言工具库。ADS 是发展强大的、丰富的 AutoCAD 二次开发系统的强有力的工具，AutoCAD 的高级扩展模型(AutoCAD Advanced Modeling Extension，简称 AME)、AutoCAD 渲染模型(Render)、AutoCAD 数据库查询扩展模型(AutoCAD SQL Extension，简称 ASE)均是 ADS 的应用程序。

3) ADS 是 C 的代码。ADS 是由一些 C 库和头文件及相应的文档构成的。

1.2 ADS 与 AutoLISP 之间的关系

ADS 不是 AutoLISP 的替代品，AutoLISP 是一种易学易用的语言，它增强非编程人员即普通用户在 AutoCAD 下开发的勇气；而 ADS 则是职业编程人员开发高级应用程序的强有力的工具。ADS 应用能扩展 AutoLISP 接口，并定义新的 AutoLISP 函数给用户使用。

(1) AutoLISP 优点及缺点　AutoLISP 易学、易用，程序文件比较小，不需要额外的开发工具，源代码在同一种版本下所有工作平台上是通用的，并可广泛得到第三方资料的支持。

另一方面，由于 AutoCAD 没有二进制及随机的 I/O，AutoLISP 只支持顺序的 ASCII 文件访问；又因为它是一个解释型语言，与 ADS 相比表现出较慢的数值计算功能，并且，该语言封闭于 AutoCAD，开发环境包含有 AutoCAD 提供的函数，但目前可用 ADS 来扩展。归纳起来，AutoLISP 对于如下几点很难实现：

1) 二进制文件读写；
2) 内存访问；
3) 联机数据库访问；
4) 直接写屏 I/O；
5) 高水平用户接口；
6) 并发进程的访问；
7) 操作系统的访问；
8) 计算机硬件的访问；
9) 最大限度地数据压缩；
10) 运行速度；
11) 拷贝保护；
12) 数据保护。

有一些缺点是可以通过引入新的 AutoLISP 函数来克服，如二进制读写文件，其它则不然。限制 AutoLISP 使用的根本原因是它整个包容在 AutoCAD 中，为解决这些问题，便推出了 ADS。ADS 程序是外部可执行的，它可以做任何您愿意做的事。

(2) ADS 的优点和缺点　ADS 使用广泛接受的商业通用语言开发环境，使用符合工业标准的编译器、连接器和调试器，使用通用的 C 和 C++库及类。ADS 表现出相当优越的数值计算功能，其应用程序通过 C 和 C++语言库得到多种操作系统功能的支持，其大小及复杂程度可以是无限制的。

ADS 应用程序是编译型的，源代码是受到保护的。

另一方面，ADS 应用系统需要额外的第三方编译器和连接器的支持，应用程序源代码是可移植的，但二进制码则不能移植。此外，C 和 C++语言比 LISP 语言难写、难用。

ADS 的开发是基本的编译—连接—调试过程，而不是交互的解释环境。典型的二进制文件要比 AutoLISP 程序大，并且每个不同的二进制文件都需要 ADS 库的支持。

那什么情况下使用 ADS，什么情况下使用 AutoLISP 呢？

AutoLISP 是最终确定 ADS 程序框架的一个优秀环境，对于那些不需要二进制随机存取或大量文件 I/O 及复杂的运算功能，而且不需要频繁调用 AutoCAD 命令的应用，最好采用 AutoLISP 来写。

ADS 是优秀的数值计算环境，如 AME 就是要用到大量的数值处理。对于那些需要用到大量系统资源或必须用二进制随机存取方面，以及大型文件的 I/O 和现存 C 语言应用代码的使用等方面需要用 ADS 编程。

(3) ADS 的工作原理　AutoCAD 通过专用的通讯缓冲将信息传到 AutoLISP 或从那里接收信息。一个 ADS 应用程序是定义了许多外部函数的 C 或 C++可执行程序，可以从 AutoLISP 装入并执行。程序包含有 ADS 与 AutoCAD 通讯的初始化函数的调用，完成初始化后，ADS 应用程序就好象是 AutoLISP 和 AutoCAD 的奴隶一样，它进入一个无限的调度循环之中，等待 AutoCAD 的请求码，一个开关语句通过接收到的 AutoCAD 结果码决定合适的执行通路。这种调度控制有些类似于 Windows 编程中的消息循环控制。

1.3 ADS的构成

建立一个ADS应用，一个开发者需要ADS库和相应的头文件，这些都包含在AutoCAD的ADS目录中；另外还需要支持AutoCAD上ADS开发的C或C++工作平台，这些信息可以从ACAD的readme.ads中查得。更详细的编译连接信息可以从ADS/DOCS中的*.TXT中找到。有关目录和信息如图1-1所示。

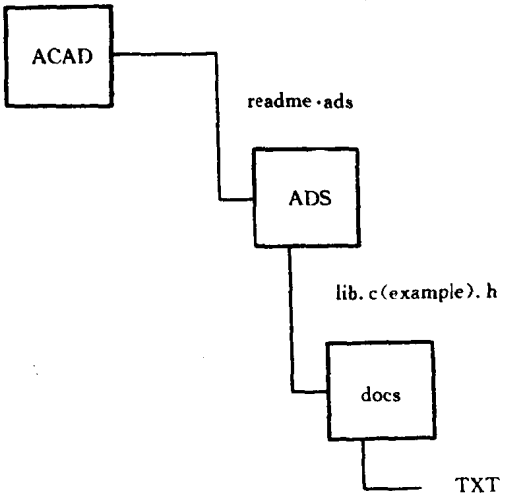

图1-1 AutoCAD中与ADS有关的目录文件信息

1.4 如何在AutoCAD中运行ADS应用程序

在AutoCAD上，装入或卸下ADS应用程序的方式共有五种：

(1) 菜单 从File菜单的Applications项可以装入和卸掉ads应用程序。

(2) AutoCAD命令appload 与第一项类似，出现对话框。

(3) LISP函数xload 如要装入test应用程序可以键入如下命令Command:(xload"test")，同样可以通过如下命令Command:(xunload"test")卸掉test。

(4) ads函数ads_xload(app) 通过ads函数可以装入ads应用程序，也可以用ads_xunload(char *app)卸掉一个应用。

(5) acad.ads文件 列在acad.ads中的ADS应用程序在AutoCAD起动时自动装入。

ADS函数的执行有两种方式，其一作为"LISP外部函数来运行"，即在用ads_defun定义时，其函数名前未加"C:"；反之则以AutoCAD命令形式来运行，如示例fact程序中的fact函数，属"LISP外部函数"运行方式为Command:(fact n)。这里值得一提的是LISP函数形式允许带参量，而命令形式则不能带参量。

此为试读，需要完整PDF请访问：www.ertongbook.com

1.5 哪些C编译支持ADS编程

开发ADS应用程序需要附加的工具，这与AutoLISP是不同的。程序源代码必须通过编译器译成机器码并用连接程序装配成可执行程序，对于AutoCAD R12，可以使用如下编译器。

1) Watcom C386 9.0或更高版本(不需要Pharlap连接器)；

2) Metaware High C386 1.X或2.X需要Pharlap386 link4.0或更高，调试则需要MDB1.3或更高；

3) Zortch386 C++3.X及Pharlap连接器link4.0或更高。

另外，AutoCAD R12 for DOS386还直接支持实模式ADS、普通的16位DOS C编译器和如下连接器：

1) Borland C++2.0或更高；

2) Turbo C++2.0 (C++ 版)；

3) Microsoft C6.0或更高版。

AutoCAD R11版不支持实模式(原来只作为SDK2.0中的一部分提供)，现在AutoCAD R12版中已支持实模式和保护模式两种方式，均可以用xload装入。

由于实模式程序运行效率比较低，所以一般建议用实模式编译器进行编程调试，而真正推出应用程序则最好采用保护模式编译器进行最后的编译连接。

2　ADS 程序结构及接口原理

2.1　ADS 应用程序的一般结构

下面是一个最简单的 ADS 应用程序 hello.c：

```
#include  <stdio.h>
#include  "adslib.h"                    /*  一定要包含此头文件 */

int user_func  _(());                   /* 用户建立的 ADS 函数原型 */

/* 下面是主函数 */
void main(int argc, char *argv[])
{
    short scode = RSRSLT;               /* 正常的 ADS 程序结果码(缺省值) */
    int stat;
    ads_init(argc, argv);               /* 建立与 AutoLISP 之间的通讯,此句
                                           必须做为第一句 ADS 可执行语句 */

    for ( ;; ) {                        /* 进入调遣循环 */
        if ((stat = ads_link(scode)) < 0) {
            printf("bad status from ads_link() = %d\n", stat);
            fflush(stdout);
            exit(1);
        }           /* 用 ads_link()函数取得 AutoLISP 请求码(参见表 2-1) */

        scode = RSRSLT;  /* 设置 ADS 程序结果码为正常值
                            (非正常值为 RSERR) */

        switch (stat) {                 /* 根据请求码选择运行通路 */
            case RQXLOAD:               /* 得到 AutoLISP 装入请求(xload),
                                           一般用 ads_defun()定义 AutoLISP 外部函数 */
                if (ads_defun("user_func", 0) != RTNORM) {
                                        /* "0" 也可为其它正整数,但必须唯一
```

```
                                    RTNORM 的含义参见表 2-4 * /
                    scode = RSERR;
                } else {
                    scode = RSRSLT;
                }
                break;
            case RQSUBR:        /* 运行外部函数请求 */
                switch(ads_getfuncode()) {        /* 取得函数码,已在前面定义,
                                并将外部 AutoLISP 函数与 ADS 函数对应起来 */
                    case 0:
                        scode = user_func() == RTNORM ? RSRSLT : RSERR;
                                /* 执行用户函数 */
                        break;
                    default:
                        ads_printf("\nError — no such function defined.");
                        scode = RSERR;
                        break;
                }
                break;
            default:
                break;
        }
    }
}

/* 用户函数 */
static int user_func()
{
    ads_printf("Hello AutoCAD R12! \n");
    return RTNORM;
}
```

此程序是一个最简单的 AutoCAD 的 ADS 程序,经编译连接后是可以运行的,在 AutoCAD 的 Command 命令下用格式(xload"hello")将其调入,然后运行所定义的外部 AutoLISP 函数格式(user_func),将打印出"Hello AutoCAD R12!"。编译连接方法可参见附录 A。

AutoLISP 与 ADS 应用程序的通讯是按照以下步骤进行的:

1) AutoLISP 通过 xload 函数将 ADS 应用程序中的函数装入,作为其外部函数供 AutoLISP 调用。

2) ADS 应用程序在装入过程中,首先通过调用 ads_init( )函数与 AutoLISP 建立初始化通

讯，它是主程序中第一个 ADS 可执行函数。

3）当初始化通讯完成后，应用程序进入了一个 for( )无条件循环（调度循环）状态，直至 ADS 应用程序退出循环状态，在此循环之前，应用程序先将结果码 RSRSLT 赋给状态变量 scode，以便调用 ads_link( )时得到正确的 AutoLISP 请求码。

4）ads_link( )函数运行后，将返回结果请求码（首先总是返回 RQXLOAD）。

5）按照结果请求码 RQXLOAD 的要求，应用程序调用 ads_defun( )函数，定义 AutoLISP 外部函数。

6）应用程序再一次将结果码 RSRSLT（当调用正确时）或 RSERR（当调用失败时）赋给状态变量 scode，再次调用 ads_link( )函数，确定整个通讯过程是否成功。

7）当用户或 AutoLISP 函数调用外部函数时，ads_link( )函数返回结果请求码 RQSUBR。

8）按照结果请求码 RQSUBR 的要求，ads_getfunccode( )函数获取函数代码，根据代码执行对应的 ADS 函数（可由用户指定）。

9）当用户所定义的 ADS 函数运行完后，再次将结果码 RSRSLT 赋给状态变量 scode，返回应用程序的 AutoCAD 的 command 状态，等待用户输入新的已定义的外部函数，新的函数名输入后，程序将重复步骤 6）至 9）的工作过程。

10）若用户输入 xunload，save，end，quit 命令，ads_link( )函数分别返回结果请求码 RQXUNLOAD（卸掉 ADS 应用程序的外部函数）、RQSAVE、RQEND、RQQUIT，应用程序将根据结果请求码的不同分别执行相应的请求。

从以上的工作步骤可以看出，当 ADS 应用程序装入内存后，程序就一直处在交互式等待的工作状态，在这种无限环循的工作状态下，用户可以输入 AutoCAD 命令、AutoLISP 命令及已定义的外部函数命令。这个过程可以用图 2-1 所示的框图表示。

表 2-1 至表 2-3 列出 ADS 应用程序常用代码表。

表 2-1　AutoLISP 的请求码

请求码	含　义
RQXLOAD	装入应用程序；定义外部函数
RQXUNLD	退出应用程序
RQSUBR	执行外部函数
RQSAVE	AutoCAD 存储图形
RQEND	AutoCAD 结束绘图
RQQUIT	AutoCAD 放弃绘图

表 2-2　ADS 应用程序的结果码

结果码	含　义
RSRSLT	结果有效
RSERR	运行产生错误；无计算结果

表 2-3　ADS 内部函数的返回码

返回码	含　义
RTNORM	结果正确
RTERROR	运行产生错误；无计算结果

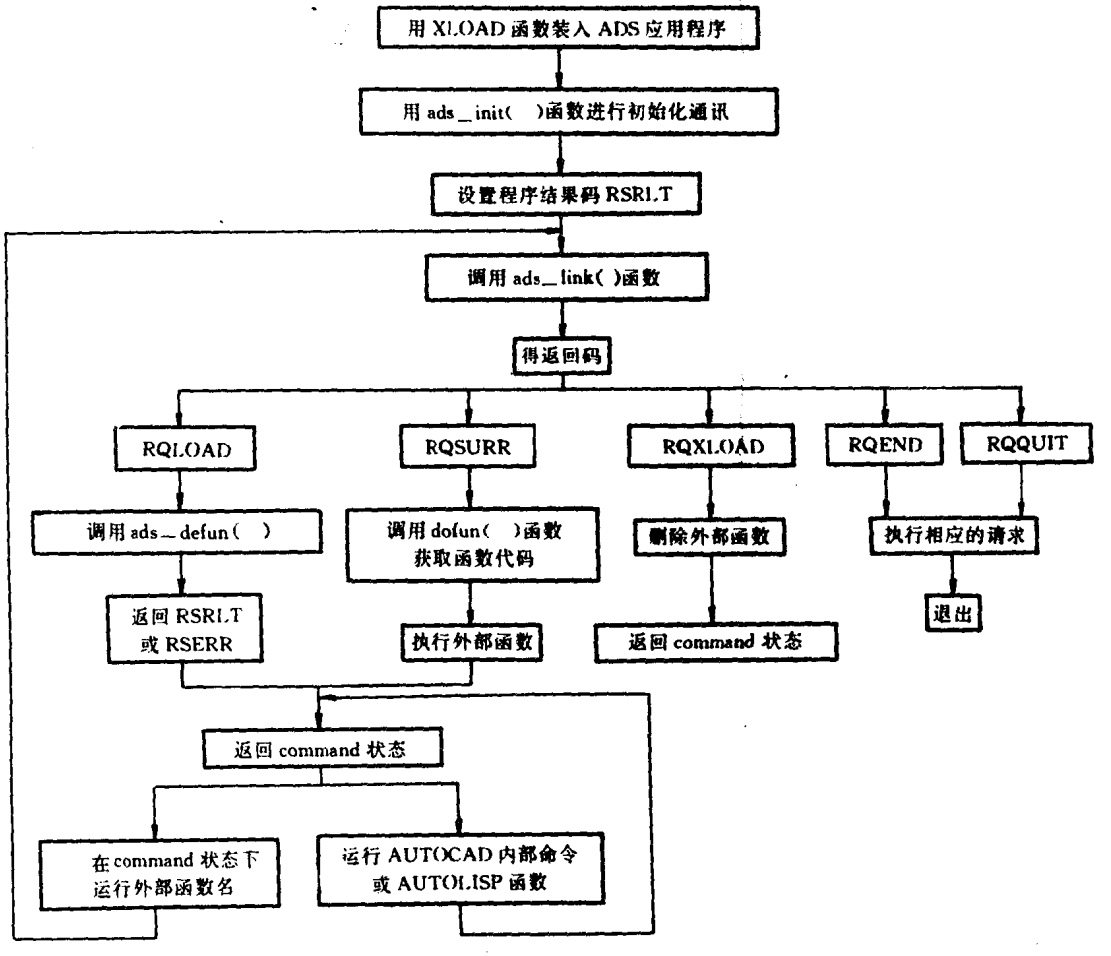

图 2-1　ADS 应用程序工作框图

2.2　ADS 应用程序的典型范例

ADS 应用程序 fact.c 是 AutoCAD R11.0 与 R12.0 中提供的一个典型范例，通过对此范例各部分的详细说明使用户对程序有一个总体的了解，在总体结构的基础上，应掌握：

1）加深对程序总体结构的理解。

2）ADS 应用程序的接口部分可作为标准样板，被其他 ADS 应用程序所采用。

3）在 ADS 应用程序的接口部分如何添加自己的应用程序。

下面是 AutoCAD R11 及 R12 中提供的 ADS 应用程序 fact.c，请用户详细阅读本程序的注释。

```
/* Next available MSG number is 8 */
/* FACT.C
```

Copyright (C) 1990 by Autodesk, Inc.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted.
THIS SOFTWARE IS PROVIDED "AS IS" WITHOUT EXPRESS OR IMPLIED WARRANTY.
ALL IMPLIED WARRANTIES OF FITNESS FOR ANY PARTICULAR PURPOSE AND OF MERCHANTABILITY ARE HEREBY DISCLAIMED.

```
DESCRIPTION:

ADS FACTORIAL application — by Mark Barrow — 3/14/90
Comments and revisions — Randy Clark, 4/20/90
Fancier function tables, and add sqr for illustration — Dan Drake, 7/21/90
*/

#include <stdio.h>
#include "adslib.h"

/* 定义一个宏,用来计算一个数组的维数,如:

        int arr[] = {1,2,3,4,5};
        ...
        printf("%d", ELEMENTS(arr));
    运行结果为5。*/

#define ELEMENTS(array) (sizeof(array)/sizeof((array)[0]))

/* 我们将定义的所有函数列在一个表(一个结构数组)中,里面包含有所要定义的外部函数和与之相关联的ADS用户函数,所有的函数只包含有一个变参(是一结果缓冲器(下一章中要介绍),为一单链表,可以包含多个参量),返回结果为一整型量(RTNORM 或 RTERROR) */

/* 首先,定义一个结构:字符串型变量给出了函数的AutoCAD名(AutoCAD命令名或AutoLISP外部函数名),函数指针返回整型量,将与所要调用的函数一致 */
```

此为试读,需要完整PDF请访问: www.ertongbook.com