

计算机数据库应用技术系列丛书

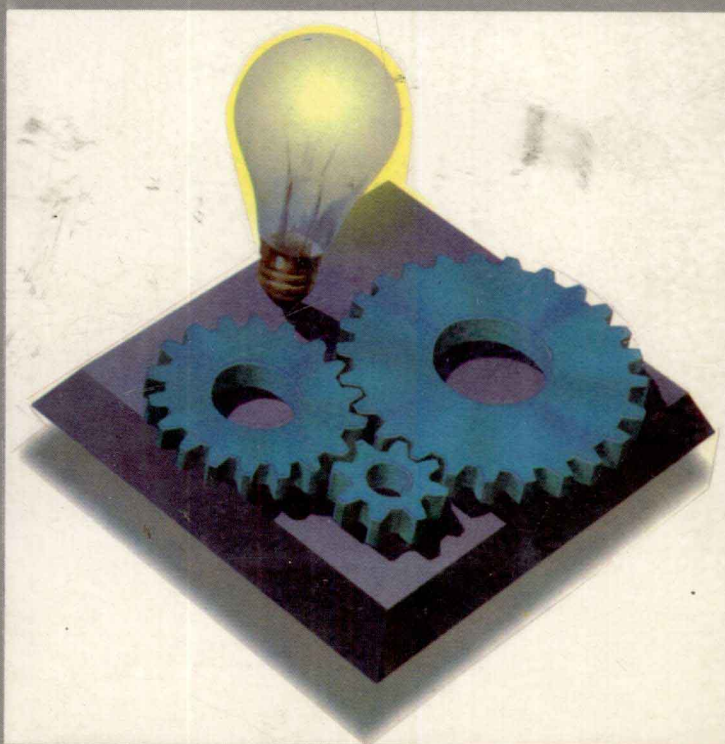
附磁盘

CA—Clipper 5.2

实 用 教 程

杨世莹 编著

学苑出版社



计算机数据库应用技术系列丛书

CA-Clipper 5.2 实用教程

——附范例磁盘

杨世莹 编著
天 奥 改编
熊可宜 审校

学苑出版社

1994

(京)新登字 151 号

内 容 提 要

本书是继 1990 年与 1991 年相继推出的 Clipper 5.0 与 Clipper 5.01 之后所推出的一本有关 Clipper 实用性很强的书,同时也是一本初学者的入门书籍,或可为自修和授课的教材。本书根据指令的功能加以分类,附有丰富实例与菜单,并且配合国内的实际使用情况,实例中均以中文数据及菜单进行设计,还加有详尽的注释说明,能使读者循序渐进地了解 Clipper 5.2 的指令及功能,是 Clipper 数据库开发与使用人员的必备参考资料。

欲购本书的用户,请直接与北京 8721 信箱联系,电话:2562329,邮码:100080。

版 权 声 明

本书繁体字中文版原书名为《CA-Clipper 5.2 程式设计入门实作》,由松岗电脑图书资料股份有限公司出版,版权归松岗公司所有。本书简体字的中文版版权由松岗公司授予北京希望电脑公司和学苑出版社独家出版、发行。未经出版者书面许可,本书的任何部分不得以任何手段复制或传播。

计算机数据库应用技术系列丛书

CA-Clipper 5.2 实用教程

编 著:杨世莹
改 编:天 奥
审 校:熊可宜
责任编辑:徐建军
出版发行:学苑出版社 邮政编码:100032
社 址:北京市西城区成方街 33 号
排 版:北京天奥科技公司激光照排中心
印 刷:双青印刷厂
开 本:787×1092 1/16
印 张:24 字 数:559 千字
印 数:1~5000 册
版 次:1994 年 1 月北京第 1 版第 1 次
ISBN7-5077-0874-8/TP·23
本册定价:42.00 元(含盘)

学苑版图书印、装错误可随时退换

目 录

第一章 概述	1
1.1 Clipper 简介	1
1.2 解释程序与编译程序	2
1.3 操作 Clipper 程序的步骤	3
1.4 Clipper 的优缺点	4
第二章 安装 Clipper 5.2	6
2.1 安装过程概述	6
2.2 Clipper 5.2 的系统需求设备	13
2.3 进行安装 Clipper 5.2	14
2.4 测试各实用程序与重建	19
2.5 建立可模拟 dBASE 点提示的 DOT.exe	21
2.6 编译与连接的批处理文件(CL.bat)	22
第三章 程序的编写、编译与连接	24
3.1 Clipper 5.2 的程序编辑器 PE.exe	24
3.2 Clipper 5.2 的程序编译器 CLIPPER.exe	26
3.3 Clipper 5.2 的连接程序 RTLINK.exe	33
3.4 一种简单的建立自定义程序库的方法	45
3.5 建立编译与连接程序的批处理文件	46
第四章 DBU.exe、RL.exe 实用程序与 REPORT、LABEL 指令	49
4.1 DBU.exe 实用程序	49
4.2 RL.exe 实用程序	54
4.3 以 REPORT FORM 指令产生标准报表	66
4.4 以 LABEL FORM 指令打印标签	68
第五章 程序语法、数据类型与运算式	70
5.1 Clipper 的程序语法	70
5.2 Clipper 的数据类型与运算式	71
5.3 宏运算符号——&	83
第六章 数据库文件管理相关指令	86
6.1 数据库文件的建立、打开与关闭	86
6.2 查询文件名	91
6.3 更改文件名	93
6.4 拷贝数据库文件	94
6.5 合并与压缩数据库文件	97
6.6 删除文件	99
6.7 修改数据库文件的结构定义	100

第七章 显示数据的指令与 OUTSTD()在中文系统下的应用 101

7.1 显示数据的指令 101

7.2 OUTSTD()在中文系统下的应用 113

第八章 记录增删与修改..... 118

8.1 增添新记录内容 118

8.2 删除记录内容 121

8.3 修改记录内容 124

第九章 排序、索引与移动记录指针 141

9.1 排序与索引 141

9.2 有关索引的几个函数 148

9.3 移动记录指针 150

第十章 查询与设置关联..... 165

10.1 查询数据内容..... 165

10.2 设置过滤条件(SET FILTER TO 与 DBSETFILTER())..... 167

10.3 建立关联..... 170

第十一章 子程序、子过程与自定义函数 178

11.1 子程序..... 179

11.2 子过程..... 186

11.3 自定义函数..... 192

第十二章 处理内存变量的指令..... 198

12.1 定义变量类别的语句..... 198

12.2 处理变量的指令..... 208

第十三章 控制程序流程的指令..... 220

13.1 控制重复执行的指令..... 220

13.2 控制执行分支的指令..... 232

第十四章 数组..... 245

14.1 定义产生数组..... 245

14.2 处理数组内容..... 247

14.3 拷贝数组内容..... 252

14.4 数组内容的排序与搜寻..... 254

第十五章 处理屏幕的指令..... 262

15.1 安排菜单内容..... 262

15.2 安排菜单..... 277

15.3 设置菜单颜色与光标..... 295

第十六章 处理键盘的指令..... 299

16.1 捕捉按键内容..... 299

16.2 管理键盘暂存器..... 312

16.3 定义按键内容(SET KEY 指令) 315

第十七章 函数	322
17.1 数值函数.....	322
17.2 字符处理函数.....	326
17.3 日期函数.....	341
17.4 转换函数.....	345
17.5 特殊测试函数.....	346
17.6 辨认函数.....	355
17.7 执行 Cold Block 的函数.....	359
第十八章 设置系统配置状态	367
18.1 以 SET...ON/OFF 或 SET...TO... 设置系统参数	367
18.2 以 SET()函数设置系统参数	372
附录 A Clipper 5.2 中无法使用的 dBASE 指令	375
附录 B Clipper 5.2 的保留字	377
附录 C 利用 Clipper 程序指令控制倚天打印控制码	378

第一章 概 述

本章内容:

- 1.1 Clipper 简介
- 1.2 解释程序与编译程序
- 1.3 操作 Clipper 程序的步骤
- 1.4 Clipper 的优缺点

1.1 Clipper 简介

在个人计算机的数据库应用上,dBASE III PLUS 算是一个相当普遍而且受欢迎的数据库管理系统。用户仅需以极简单的指令,即可轻易完成数据库的创建、增添、修改、查询、索引以及产生报表或标签的工作,甚至利用其程序语言开发应用系统程序。但其最大的缺点即是采用解释式(interpreter)语言,这种语言以交互方式与用户双向沟通,极具可观性,易学易懂,但就是执行速度较慢!

由于 dBASE 的用户众多,还有多家软件公司投入于将其语言转换成编译式(compiler)的市场,以增快其速度来吸引用户购买。Nantucket 公司即为较具有代表性的软件公司之一,其产品 Clipper 为目前较广泛被采用作为 dBASE 的编译器,并已获得极高的评价。也因这些软件公司使用 Clipper 有数年的经验,也自行开发了许多非常实用的自定义函数或过程以方便其开发应用系统,故即使 dBASE Compiler 已在 1993 年推出,但各软件公司对它好像还是没有兴趣!

在 Clipper 中,我们除可使用绝大多数的 dBASE 指令(仍有少部分的指令无法兼容,参见附录 A)与各类型文件外;还可利用 Clipper 新增的指令或函数来强化执行效果与速度(Clipper 的执行速度约为 dBASE 的 5~20 倍)。我们甚至可以说,dBASE 可做到的 Clipper 也大都能完成;但 Clipper 能做的 dBASE 就未必能做到!到此,我们可能会有个疑问:Clipper 既然拥有如此众多的优点,为何无法取代 dBASE 呢?其最大的原因应在其采用编译式语言,用户界面不友好。故至今仍只限于专门替代别人开发应用系统的专业程序员中较被广泛使用!

通常,以 dBASE 所编写的程序(.prg)文件,只能在 dBASE 下执行。但若将其以 Clipper 编译及连接(link)成.exe 文件后,此程序即可退出 dBASE 而可直接在 DOS 下执行。对某些不自行开发程序的用户来说,即可省下购买 Clipper 及 dBASE 软件的经费。所以,国内多数软件公司均采用 Clipper 来为客户设计应用系统。如此除了程序较易开发且执行速度较快外;还具有一重大优点——源程序经过编译及连接成.exe 文件后已成为机器码,客户无法得知(或者篡改)源程序的内容,较 dBASE 的.prg 文件更具安全性与保密性。此项优点应为软件公司原来放弃 dBASE 而采用 Clipper 的主要原因。

随着 dBASE 的不断改进,Clipper 也不断推陈出新(除随着 dBASE 的变更,也修正本身既有的缺点),Nantucket 公司先后推出的 Clipper 版本共有:85 年版、86 年版、87 年夏季版、88 年春季版、90 年 Clipper 5.0 以及 91 年推出的 Clipper 5.01 版等各种版本。93 年推出的 Clipper

5.2 版,而在 Nantucket 公司为 Computer Associates International, Inc. 收并之后的事,故又称之为 CA—Clipper 5.2。

1.2 解释程序与编译程序

前面所讲到的 dBASE 采用解释,而 Clipper 采用编译方式,将我们所写的程序转换成可被计算机执行的机器语言。

采用解释方式翻译 dBASE 指令的操作如图 1.1 所示,每翻译完一行指令码直接执行该指令。执行后用户即可看到执行结果或查知程序的错误;接着再进行下一行指令的翻译与执行操作。如此,依序翻译、执行……,直到所有指令均处理完毕为止。简而言之,解释程序一次只翻译并执行一行指令,任一行指令执行前均得重新翻译,因此耗费在翻译的时间较长,故执行速度较慢。特别在执行循环(loop)时,翻译指令所花的时间将随循环的执行次数而明显增加。不过,这种执行方式的优点为:易学易懂,极具可观性,程序的编写和调试较为容易,非常适合初学者使用。

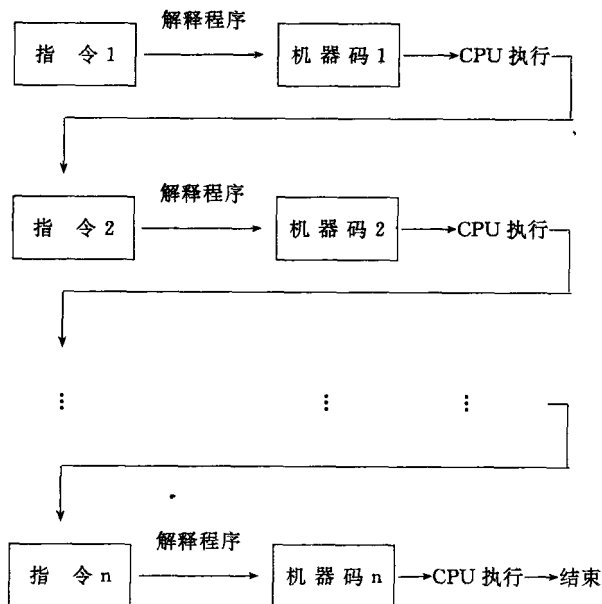


图 1.1 解释程序翻译、执行 dBASE 程序的概念图

Clipper 编译程序翻译 Clipper 程序指令的操作则如图 1.2 一样,一次即将整个源程序翻译成机器语言,以后执行即不用再重新翻译,所以其执行效率较高、速度较快。但其缺点为:必须每行指令均完全正确才能执行。对初学者来说,这不是一件容易的事!可能一个程序经过数十次的编译与修改后仍无法顺利执行,对初学者的信心打击相当严重,甚至令其望而却步,路途放弃学习!此外,每一个程序在执行前均得经过编译和连接的过程,任一行指令更改,又得重新再编译和连接后才能执行,而这段工作所耗的时间可能得花 1~10 分钟(看计算机速度而异)。假如用户没有经验,可能耗上一整天,仅看到比原程序还长的错误信息而已,根本没看到任何执行结果(无法通过其成就来体验编写程序的兴趣,怎样能提高学习兴趣呢?);而不像

dBASE 那样,一行可执行的指令即可导致某种程度的执行效果(就算指令错了,它还很热心地问你——Do you want some help?)。

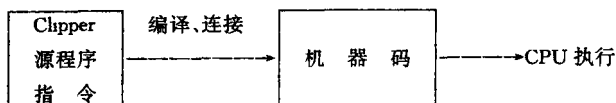


图 1.2 编译程序翻译、执行 Clipper 程序的概念图

综合前文说明可知,dBASE 程序的特性为:程序编写与调试较为容易,但执行速度较慢。而 Clipper 程序的特性为:程序编写与调试较为困难,编译及连接程序耗时较长,但执行的速度则较快。

1.3 操作 Clipper 程序的步骤

事实上,将 Clipper 的源程序编译后仍无法执行,还得经过一连接的过程。图 1.3 标示出我们操作 Clipper 程序的基本步骤,依序为:

1. 以字处理程序编写源程序文件(.prg)

此一步骤可用任何字处理程序来完成,如:PE2、PE3、DW3 或 Clipper 所提供的程序编译器——PE(Program Editor)。若无特别理由,请赋予.prg 扩展名,如:TEST.prg。

2. 将源程序编译成目标程序文件(.OBJ)

此一步骤须以 Clipper 5.2 的编译程序 CLIPPER.EXE 来完成,若程序无语法错误即可将源程序编译成目标程序(object file),缺省的扩展名为.OBJ。例如:将 TEST.PRG 编译成 TEST.OBJ。

3. 将目标程序与程序库连接成可执行程序(.EXE)

此一步骤须以 Clipper 5.2 的连接程序 RTLINK.EXE 来完成。由于,每一指令应有的操作是由一连串(module)集合而成,而这些模块则存入在 Clipper 的程序库文件内(library file 扩展名为.LIB,如:CLIPPER.LIB、EXTEND.LIB、DBFNTX.LIB 与 TERMINAL.LIB)。此外,目标程序内仍有许多调用库函数的指令未处理,例如:处理数学运算的数值函数,或进行日期转换的日期函数……等,这些函数所需的模块也仍存入在程序库中。故得将相关模块一起连接到可执行文件,才可直接在 DOS 下执行。由程序库中取出目标程序指令与函数所需的模块,加以连接后所产生的可执行文件的缺省扩展名为.EXE。例如:将 TEST.OBJ 连接成 TEST.EXE。

有时,也可能将数个目标程序汇集在一起而连接成单一的可执行程序,例如:将 MAIN.OBJ、UPDA.OBJ、PRINT.OBJ 等目标程序连接成 MAIN.EXE。

4. 执行程序

目标程序连接成可执行文件后,可在 DOS 下直接以其文件名来调用执行。如:

C>TEST

若发觉执行结果仍有不妥,则仍得再重复上述步骤 1~步骤 4 的工作,重新修改源程序,通过编译、连接后才能再执行。

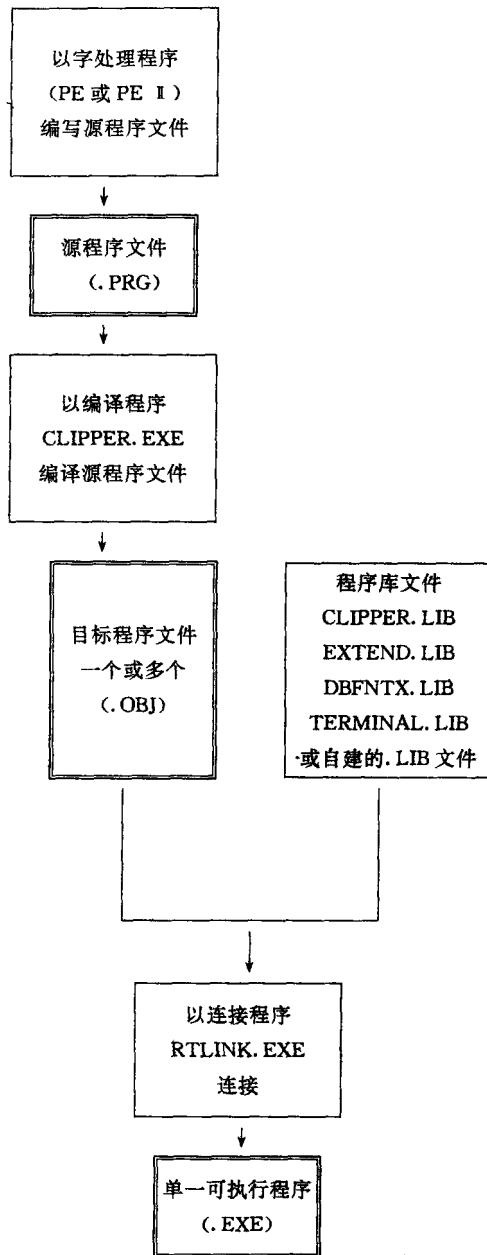


图 1.3 操作 Clipper 程序的基本步骤

1.4 Clipper 的优缺点

Clipper 程序相对于 dBASE 程序,至少有下列几个优缺点:

- 1) 执行速度快。
- 2) 源程序较具有保密性与安全性。
- 3) 在 DOS 下即可执行,不必进入 dBASE。对程序的用户来说,可省下购买 dBASE 的成

本。

- 4) 可使用更多的文件,如:工作区可达 250 个,同时打开的索引文件可达 15 个。
- 5) 可处理一维数组(array)与多维数组,其元素可达 4,096 个。
- 6) 备注数据(memo)的处理能力较强,如:REPLACE 指令已可将字符存入备注数据栏、MEMOEDIT()可以全屏幕编辑菜单对备注数据进行修改。
- 7) 字符或备注数据的长度均可达 65,519。
- 8) 用户可自定义函数或以预处理自建指令。
- 9) 提供用户更多函数及指令。无论在字符、数值或日期数据类型中,均有很多新增函数为 dBASE III PLUS 所没有。
- 10) 与 C 语言或汇编语言的沟通更为容易。
- 11) 可使用更多的内存变量(2,048 个)。
- 12) 屏幕、键盘的控制与规划较为简便。如:在 dBASE III PLUS 中,要安排一可利用 ↑、↓、←、→键移动选择指针,并配合以回车键进行选择的下拉或上托功能菜单,得全靠我们自行设计,其过程很繁琐。但在 Clipper 中此一工作就显得格外简单。仅须先以 @... PROMPT... MESSAGE... 安排好各选项的内容及应有的提示信息,再以 MENU TO... 启动即可。

相对地,Clipper 也有其缺点,与 dBASE 相比较它至少有下列几个缺点:

- 1) 较不具有可观性,不易学习。
- 2) 程序在达到可执行的境界前,所耗的编译及连接时间较长。
- 3) 程序的编写与调试较为困难。
- 4) 过份强调指令的精简性,使得其程序的可阅读性较差。
- 5) 拥有庞大的程序库、函数库,非 386 占空间。无硬盘的系统几乎无法操作执行,这也是某些学校或补习班无法以这作为教材的主要原因。
- 6) 连接后的可执行程序变得相当大,可能只是几行简单指令(未满 50 个字符)的源程序,编译连接后的 .exe 文件竟要占用约 200K 的空间。

第二章 安装 Clipper 5.2

本章内容：

- 2.1 安装过程概述
- 2.2 Clipper 5.2 的系统需求设备
- 2.3 进行安装 Clipper 5.2
- 2.4 测试各实用程序与重建
- 2.5 建立可模拟 dBASE 点提示的 DOT.exe
- 2.6 编译与连接的批处理文件 (CL.bat)

Clipper 5.2 所提供的程序均经过压缩后存于磁盘, 这些程序并无法立即执行, 必须通过其 INSTALL.exe 将相关程序解码(解压缩), 安装到硬盘后才可执行。执行安装的主要目的在将软件中通过压缩过的所有系统程序, 以反压缩解码拷贝到硬盘的适当子目录。故其安装程序会先在硬盘中建立几个子目录, 然后再将相关程序分别拷贝。此外许多实用程序是以源文本形式拷贝存放, 故在其安装过程中还得将这几个文件进行编译与连接, 以产生可执行文件。

2.1 安装过程概述

Clipper 5.2 的安装过程可分为下列几个部分：

一、建立相关子目录

将在指定的磁盘根目录下(如:C:\), 建立表 2.1 所示的几个子目录, 各子目录的作用详细说明见下表。

表 2.1 安装过程自动建立的子目录及其作用

子 目 录	作 用
C:\CLIPPER5	
\BIN	存放编译及连接所需的 .EXE 文件
\INCLUDE	存放编译及连接所需包含的头文件(.CH 或 .H 文件, Clipper Header File)
\LIB	存放连接所需的程序库文件(.LIB, Library File)
\PLL	存放前置连接所需包含的程序库(.PLL, Pre-Linked Library file)
\OBJ	当前空白可存要进行连接的 .OBJ 文件
\SOURCE	
\DBU	存放编译及连接 DBU .EXE 实用程序所需的源程序, 可用以建立及修改数据库文件
\PE	存放编译及连接文字编辑程序——PE .EXE, 所需的源程序

\RL	存放编译及连接制作报表文件(.FRM)及标签文件(.LBL)的实用程序 ——RL.EXE,所需的源程序
\SAMPLE	存放范例程序,如:模拟点提示状态的 DOT.PRG
\SYS	存放部分与系统设置有关的源程序
C:\NG	存放联机帮助说明,执行过 NG 后以 Shift—F1 启动

二、自软盘将系统程序及相关文件反压缩解码拷贝到适当子目录

依序将系统磁盘上经过压缩的系统程序及相关文件,以反压缩解码方式分别拷贝到先前建立的适当子目录下(Clipper 5.2 有两片 3.5 寸磁盘)。各子目录下所拷贝的文件及其作用,请参见表 2.2~表 2.12。

表 2.2 C:\CLIPPER5 子目录下所存的文本文件及作用

文 件 名 称	作 用
README	有关安装 Clipper 5.2 的注意事项及操作手册的勘误表的文本文件
NATION.DOC	有关欧洲各国用户适应其文字不同所需注意的事项

注:均可用字处理程序查询或以 TYPE 打印。

表 2.3 C:\CLIPPER5\BIN 子目录下所存程序及作用

文 件 名 称	作 用
CLIPPER.EXE	编译程序
RTLINK.EXE	连接程序
RTLINK.HLP	连接程序的联机帮助说明
RTLINKST.COM	RTLINK.exe 的启动程序(Startup code)
TRLINK.DAT	连接程序的数据文件
CLD.EXE	查错程序(debugger)
RO.COM	将文件转换成只读(read only)状态
DBT52.EXE	将.dbt 文件转换成 Clipper 5.2 可用的文件
RMAKE.EXE	同时执行编译与连接的工作
CL.BAT	执行编译与连接的批处理文件
RL.EXE	建立报表与标签的实用程序
PE.EXE	文字编辑实用程序
DBU.EXE	建立及修改数据库的实用程序
DBU.HLP	DBU.exe 实用程序的联机帮助说明

注:CL.BAT、RL.EXE、PE.EXE、DBU.EXE 与 DBU.HLP 是在安装后自行拷贝。

表 2.4 C:\CLIPPER5\INCLUDE 子目录下所存头文件及作用

文 件 名 称	作 用
ACHOICE.CH	ACHOICE()函数引用
ASSERT.CH	调试程序所引用
BOX.CH	定义表框符号
CLIPDEFS.H	与 C 语言沟通时所引用
COLOR.CH	COLORSELECT()函数引用
COMMON.CH	一般通用的定义用指令转换
DBEDIT.CH	DBEDIT()函数引用
DBSTRUCT.CH	DBSTRUCT()取得数据结构时引用
DIRECTRY.CH	DIRECTORY()函数引用的头文件
ERROR.CH	定义错误码的头文件
EXTASM.INC	与汇编语言沟通时引用的头文件(Clipper 5.x)
EXTEND.API	有关 EXTEND 系统在 C 语言中的定义
EXTEND.H	与 Micro soft 5.x 沟通时引用的头文件
EXTENDA.MAC	与汇编语言沟通时引用的头文件(Clipper '86)
EXTENDA.INC	与汇编语言沟通时引用的头文件(Clipper '87)
FILEIO.CH	定义文件输入/输出的头文件
FILESYS.API	有关低级文件输入/输出的 C 语言定义
GETEXIT.CH	定义有关@...GET 的头文件
GT.API	有关一般终端机(general terminal)在 C 语言中的定义
INKEY.CH	INKEY()函数引用的头文件
ITEM.API	与 C 语言沟通时所引用的头文件
MEMOEDIT.CH	MEMOEDIT()函数引用的头文件
RESERVED.CH	有关保留字的头文件
RDD.API	RDD(replaceable database drivers,使各不同软件的数据库可以互用)所需使用有关 C 语言的定义
SET.CH	如 SET()函数或指令引用的头文件
SETCURS.CH	光标设置引用的头文件
SIMPLEIO.CH	标准输入/输出函数引用的头文件
STD.CH	标准指令引用的头文件
VM.API	控制 VM(Virtual Memory,虚拟存储器)所需使用有关 C 语言的定义

注:头文件属文本性质的,可以任何字处理程序查询其内容,将有助于了解其作用。

表 2.5 C:\CLIPPER5\LIB 子目录下所存程序库及作用

文 件 名 称	作 用
ANSITERM.LIB	设备符合美国国家标准局所定标准
CLIPPER.LIB	主要程序指令与函数
CLD.LIB	查错程序指令与函数
DBFCDX.LIB	数据库文件与索引文件的相关程序库(FoxPro)
DBFMDX.LIB	数据库文件与索引文件的相关程序库(dBASE IV)
DBFNDX.LIB	数据库文件与索引文件的相关程序库(dBASE III)
DBFNTX.LIB	数据库文件与索引文件的相关程序库(Clipper)
DBPX.LIB	数据库文件与索引文件的相关程序库(Paradox)
EXTEND.LIB	扩充指令与函数
NOVTERM.LIB	设备非标准型所需的程序库
PCBIOS.LIB	掌握一般基本输入/输出
RTLUTILS.LIB	制作覆盖文件(overlay file)时使用的程序库
SAMPLES.LIB	存放范例程序的.OBJ 文件
TERMINAL.LIB	掌握终端机键盘或屏幕输入/输出

表 2.6 C:\CLIPPER5\PLL 子目录下所存程序及作用

文 件 名 称	作 用
BASE52.PLL	前置连接的程序库(Pre-Linked Library)
BASE52.PLT	前置连接的转换文件(Pre-Linked Transfer file)
BASE52.LNK	连接 BASE52.PLL 与 BASE52.PLT 的文本性质配置文件

注, BASE52.PLL 与 BASE52.PLT 在现阶段还未产生。

表 2.7 C:\CLIPPER5\SOURCE\DBU 子目录下所存程序及作用

文 件 名 称	作 用
DBU.RMK	以 RMAKE.exe 制作 DBU.exe 实用程序时的文本性质配置文件
DBU.HLP	DBU.exe 实用程序的帮助说明文件
DBU.PRG	DBU.exe 实用程序的主程序, 可用以建立、修改、拷贝数据库文件或对其建立索引及产生关联
DBUSTRU.PRG	DBU.exe 实用程序中执行建立数据库文件的子程序
DBUCOPY.PRG	DBU.exe 实用程序中执行文件拷贝的子程序
DBUEDIT.PRG	DBU.exe 实用程序中执行数据修改的子程序
DBUHELP.PRG	DBU.exe 实用程序中执行调用帮助说明的子程序
DBUINDX.PRG	DBU.exe 实用程序中执行建立索引的子程序

DBUUTIL. PRG	DBU. exe 实用程序中各实用程序的模块
DBUVIEW. PRG	DBU. exe 实用程序中执行建立. vue 文件的子程序

表 2.8 C:\CLIPPERS\SOURCE\PE 子目录下所存程序及作用

文 件 名 称	作 用
PE. PRG	Clipper 提供的文字编辑程序——PE. exe 的源程序,可用以编写程序或编辑文件
PE. RMK	以 RMAKE. exe 制作 PE. exe 文字编辑程序时的文本性质配置文件

表 2.9 C:\CLIPPERS\SOURCE\RL 子目录下所存程序及作用

文件名称	作 用
RL. RMK	以 RMAKE. exe 制作 RL. exe 功能程序时的配置文件,用以建立报表格式文件(. frm)与标签文件(. lbl)
RLBACK. PRG	建立标签文件与报表格式文件的子程序
RLDIALG. PRG	控制建立标签文件与报表格式文件时交互式菜单的子程序
RLFRONT. PRG	产生标签或报表的子程序

表 2.10 C:\CLIPPERS\SOURCE\SAMPLE 子目录下所存范例程序

文 件 名 称	文件名称	文件名称	文件名称
ARRAY. CH	ARRAY. PRG	ASRTDEMO. PRG	BOX. PRG
BROWSE. PRG	DATE. PRG	DAYS. PRG	DBF. PRG
DICT. CH	DICTT. PRG	DOT. PRG	ELAPTIME. PRG
ENVIRON. PRG	EXAMPLEA. ASM	FILEDEMO. PRG	FILEDEMO. RMX
FILEIO. PRG	AMPM. PRG	FKLABEL. PRG	FKMAX. PRG
GAUGDEMO. PRG	GAUGE. PRG	ITERATOR. PRG	KEYBOARD. PRG
LENNUM. PRG	LOCKS. PRG	LOCKS87. CH	MOD. PRG
NUM. PRG	PRINT. PRG	READKEY. PRG	SCROLBAR. PRG
SECS. PRG	SOUNDEX. C	STACK. PRG	STATUS. PRG
STRING. PRG	STRZERO. PRG	STUFF. C	FILEMAN. CH
FILEMAN. PRG	TIME. PRG	TIME87. CH	TSTRING. PRG
VAEDIT. PRG	TBEDMO. PRG	TEMPLATE. ASM	

表 2.11 C:\CLIPPER5\SOURCE\SYS 子目录下所存程序及作用

文 件 名 称	作 用
FRMDEF.CH	与产生报表指令(REPORT FORM)有关定义的头文件
FRMBACK.PRG	将.lbl 标签文件的定义存入数组
FRMARUN.PRG	执行 REPORT FORM 产生报表
LBLDEF.CH	与产生标签指令(LABEL FORM)有关的定义的头文件
LBLBACK.PRG	将.lbl 标签文件的定义存入数组
LBLRUN.PRG	执行 LABEL FORM 产生标签
ERRORSYS.PRG	控制错误状况的标准程序
GETSYS.PRG	掌握@...GET/READ 相关操作的程序
RDDSYS.PRG	控制 RDD 的切换,缺省为 DBFNTX。
NTXERR.PRG	掌握 DBFNTX 的错误信息。

注 此处如 prg 文件均已另编译成 obj 文件,存放在 CLIPPER.lib 程序库。

表 2.12 C:\NG 子目录下所存程序及作用

文 件 名 称	作 用
NG.EXE	联机帮助说明的启动程序,执行后以 Shift-F1 启动
C52G01B.NG	联机帮助说明的数据文件(指令部分)
C52G02B.NG	联机帮助说明的数据文件(错误信息部分)
C52G03B.NG	联机帮助说明的数据文件(扩充函数部分)
C52G04B.NG	联机帮助说明的数据文件(实用程序的操作说明部分)
C52G05B.NG	联机帮助说明的数据文件(5.2 版的特殊说明)
C52G06B.NG	联机帮助说明的数据文件(范例)
C52G07B.NG	联机帮助说明的数据文件(RDD 驱动程序的指引)

三、更改 DOS 工作环境

Clipper 所使用的工作环境 与 DOS 缺省状况有点不同,故须在安装过程中加以调整。Clipper 5.2 的安装程序会依用户意愿自动决定是否更改计算机系统 中的 CONFIG.SYS 与 AUTOEXEC.BAT 两个文件的内容,以重定 DOS 的工作环境。在修改此两文件的内容前,会先将其原内容存入 AUTOEXEC.BAK 与 CONFIG.SYS 两备份文件中

对 CONFIG.SYS 文件将加入的内容及意义为:

FILES=30 (定义可同时打开的文件个数的上限)

BUFFERS=15 (定义存储器可使用的缓冲器个数)

而对 AUTOEXEC.BAT 文件将加入的内容及意义为:

PATH=C:\CLIPPER5\BIN;C:\NG;%path%

(设置文件搜寻路径为 C:\CLIPPER5\BIN;C:\NG 后再接原 PATH 的设置)