

FoxBASE 数据库通用表格输出处理的方法

阮学武 王毓彬 (云南省会泽县统计委员会)

在 FoxBASE 的使用过程中,经常需要将数据库的数据内容在表格上输出,由于每一种输出表格或输出数据的处理不尽相同,均需要编写单独的输出程序,尤其是对某些只使用一次的表格也是如此,既费时又费力。这样显然大大加重了程序编制的工作量,同时也减少了磁盘可利用空间,对于系统的维护和使用就显得冗余,给实际工作带来了许多不便。为此,我们在运用 FoxBASE 的实际工作中,探索编制了一套 FoxBASE 数据库通用处理(打印及文本输出)程序,以满足各种不同形式或格式的 FoxBASE 数据库的表格输出。

用户首先用外部文本编辑软件编制一个所需表头文本文件,程序根据用户所建立的表头文本文件,将其读入 biaotou. \$\$\$ 的数据库中(其中 biaotou. \$\$\$ 数据库是一个只有一个字符型字段 bt 的空库,字段 bt 的宽度为 254 字节。它用来临时存放表头文件信息,以供 FoxBASE 访问),然后从 biaotou. \$\$\$ 数据库中根据标识将表标题、表框、中画线、字段、底行等读入相应的数组变量中,打开相应的数据库名,将该库的结构部分拷贝到 jie gou. \$\$\$ 数据库中,再将要输出的名字段名称及小数位赋值给数组保存,通过中画线来判断各栏宽度,最后再调用实际的数据库输出。为了简化程序,对于输出的各字段通过函数 pstr 来判断其类型,并将之转换为字符型。本程序能自动识别开式表或闭式表,能自动将表标题以 B 字体居中输出,其余表框及数据部分以 A 字体输出。对于数值型字段若为 0 则输出空栏,否则右对齐输出,其余字段左对齐输出。若字段超过表格的宽度程序自动将其截尾,本程序只能识别细线条所画的表框,对于逻辑型字段程序没有考虑其输出。另外,对于表头文件中的行宽不能超过 254 个字节,不过这对于一般的打印机所能打印的宽度已经足够了。

举例及程序清单附后。(见下页)

例如数据库 jl.dbf 其结构及内容如下:

Field	Field Name	Type	Width	Dec
1	BH	Character	6	
2	XM	Character	8	
3	RQ	Date	8	
4	GZ	Numeric	6	2
5	DW	Character	12	
6	DH	Character	7	

** Total ** 248

需要输出的表头格式文本文件为:

人事档案
====
单位:

姓名	出生 年月	工 资	联系地址	
			工作单位	电话
#2#3#4#5#6				

* bt
* bt
* bk
* bk
* bk
* bk
* zx
* zd
* dh

1994.4.27

[说明:上面表框中标识 * bt 为表标题; * bk 为表框; * zx 为中画线; * zd 为表字段(其中: #2 #3 #4 #5 #6 分别表示所要输出的数据库中的字段 2 XM; 字段 3 RQ; 字段 4 GZ; 字段 5 DW; 字段 6 DH); * dh 为表底行]

调用 MAIN 程序后输出结果为:

人事档案

====

单位:

姓名	出生 年月	工 资	联系地址	
			工作单位	电话
张三	05 / 12 / 60	350.01	一车间	2364592
李四	01 / 05 / 65	200.10	二车间	3624798
王五	06 / 10 / 70	100.12	三车间	2210469
赵六	12 / 07 / 69	230.05	四车间	2221465

1994.4.27

主程序 MAIN 清单如下:

```
* (打印 FoxBASE 数据库程序清单)
SET TALK OFF
SET SCOR OFF
SET STAT OFF
CLEAR
* 接收用户输入数据库名及表头文件名并判断其是否存在
DO WHILE T.
  ACCE "请输入数据库名:" TO M->MAIN_DF
  IF AT(".",M->MAIN_DF)=0
    M->MAIN_DF=M->MAIN_DF+".DBF"
  ENDI
  IF .NOT.FILE(M->MAIN_DF)
    ?CHR(7)+"数据库不存在,请重输!"
  LOOP
  ENDI
  EXIT
ENDD
DO WHILE T.
  ACCE "表头文件名:" TO M->MAIN_F
  IF .NOT.FILE(M->MAIN_F)
    ?CHR(7)+"表头文件不存在,请重输!"
  LOOP
  ENDI
  EXIT
ENDD
* 选择输出到打印机或磁盘文件
DO WHILE T.
  WAIT "输出到(1.打印机;2.磁盘)=>" TO M->MAIN_OD
  IF M->MAIN_OD="1"
    IF M->MAIN_OD="2"
      ACCE "输出文件名:" TO M->MAIN_OF
    ENDI
  EXIT
ENDI
ENDD
* 接收每页输出行数及起止记录号
INPUT "请输入每页行数:" TO M->MAIN_H
INPUT "从第几条记录开始:" TO M->MAIN_Q
INPUT "到第几条记录结束:" TO M->MAIN_Z
SELE 1
* 将表头文件追加到 BIAOTOU. $$$ 数据库中
```

```
USE BIAOTOU. $$$
APPE FROM &MAIN_F SDF
* 定义数组 MAIN_BK, MAIN_DH, MAIN_BT 分别用于保存表框、底行、表标题
DIME MAIN_BK(40),MAIN_DH(10),MAIN_BT(10)
M->MAIN_ZX=""
M->MAIN_JBK=1
M->MAIN_JDH=1
M->MAIN_JBT=1
GO TOP
* 从 BIAOTOU. $$$ 库中读出内容并根据标示符分别送入数组和变量中
DO WHILE .NOT.EOF()
  DO CASE
    CASE AT(" * bt",BT)>0
      MAIN_BT(M->MAIN_JBT)=LTRIM(TRIM(LEFT(BT,AT(" * bt",BT)-1)))
      M->MAIN_JBT=M->MAIN_JBT+1
    CAST AT(" * bk",BT)>0
      MAIN_BK(M->MAIN_JBK)=TRIM(LEFT(BT,AT(" * bk",BT)-1))
      M->MAIN_JBK=M->MAIN_JBK+1
    CASE AT(" * zk",BT)>0
      M->MAIN_ZX=TRIM(LEFT(BT,AT(" * zk",BT)-1))
  )
  CASE AT(" * zd",BT)>0
    M->MAIN_ZD=TRIM(LEFT(BT,AT(" * zd",BT)-1))
  )
  CASE AT(" * dh",BT)>0
    MAIN_DH(M->MAIN_JDH)=TRIM(LEFT(BT,AT(" * dh",BT)-1))
    M->MAIN_JDH=M->MAIN_JDH+1
  ENDC
  SKIP
ENDD
* 删除 BIAOTOU. $$$ 库内容并将其关闭
DELE ALL
PACK
USE
* 判断输出字段个数
M->MAIN_S=1
M->MAIN_GS=0
DO WHILE M->MAIN_S <= LEN(M->MAIN_ZD)
  IF SUBS(M->MAIN_ZD,M->main_s,1)="#"
```

```

M->MAIN_GS=M->MAIN_GS+1
ENDI
M->MAIN_S=M->MAIN_S+1
ENDD
* 定义二级数组 MAIN_PZD 用于保存各输出字段的字段
名、宽度、小数位
DIME MAIN_PZD(M->MAIN_GS,3)
SET SAFE OFF
USE &MAIN_DF
COPY TO JIEGOU. $$$ STRU EXTE
SET SAFE ON
COUNT TO M->MAIN_ZI
IF M->MAIN_Z=M->MAIN_ZI
  M->MAIN_Z=M->MAIN_ZI
ENDI
USE JIEGOU. $$$
M->MAIN_S=1
M->MAIN_GS=1
DO WHIL M->MAIN_S<=LEN(M->MAIN_ZD)
  IF SUBS(M->MAIN_ZD,M->MAIN_S,1)="#"
    M->MAIN_JL=VAL(SUBS(M->MAIN_ZD,(M->
MAIN_S+1)))
    IF M->MAIN_JL=0
      MAIN_PZD(M->MAIN_GS,1)="RECNO"
      MAIN_ZI
      MAIN_PZD(M->MAIN_GS,3)=0
      M->MAIN_GS=M->MAIN_GS+1
    ELSE
      GO M->MAIN_JL
      MAIN_PZD(M->MAIN_GS,1)=FIELD_NAME
      MAIN_PZD(M->MAIN_GS,3)=FIELD_DEC
      M->MAIN_GS=M->MAIN_GS+1
    ENDI
  ENDI
  M->MAIN_S=M->MAIN_S+1
ENDD
* 判断表框为闭式表或开式表并作相应处理
M->MAIN_GS=1
IF SUBS(M->MAIN_ZX,1,2)="|"
  M->MAIN_S=3
  M->MAIN_BX="|"
ELSE
  M->MAIN_S=1
  M->MAIN_BX=""
  M->MAIN_ZX=M->MAIN_ZX+" "
  ENDI
  * 判断各输出栏宽度
  M->MAIN_SS=0
  DO WHIL M->MAIN_S<=LEN(M->MAIN_ZX)-1
    IF SUBS(M->MAIN_ZX,M->MAIN_S,2)◇"—"
      MAIN_PZD(M->MAIN_GS,2)=M->MAIN_SS
      M->MAIN_SS=0
      M->MAIN_S=M->MAIN_S+2
      M->MAIN_GS=M->MAIN_GS+1
    ENDI
    M->MAIN_S=M->MAIN_GS-1
    IF M->MAIN_OD="1"
      CLEAR
      @3,10 SAY "请准备好打印机,按任意键继续..."
      WAIT ""
      @5,10 SAY "开始打印"
    ENDI
    M->MAIN_SS=1
    * 打开要输出的数据库
    USE &MAIN_DF
    IF M->MAIN_OD="2"
      SET ALTE TO &MAIN_OF
      SET ALTE ON
    ENDI
    * 制表输出
    DO WHIL M->MAIN_Q<=M->MAIN_Z
      SET CONS OFF
      IF M->MAIN_OD="1"
        SET PRIN ON
      ELSE
        SET ALTE ON
      ENDI
    * 按 B 字体中对齐输出表标题
    M->MAIN_S=1
    DO WHIL M->MAIN_S<M->MAIN_JBT
      M->MAIN_QQ1=MAIN_BT(M->MAIN_S)
      IF LEN(M->MAIN_QQ1)*2>=LEN(M->MAIN_ZX)
        ?CHR(27)+"IB"+M->MAIN_QQ1+CHR(27)+"IA"
      ELSE
        M->MAIN_X=INT(LEN(M->MAIN_ZX)-LEN(
M-MAIN_QQ1)*2)/2)

```

```

CHR
(27)+"IA"+SPAC(M-AIN__X)+CHR(27)+"IB"+M->MAIN__QQ1+CHR(27)+"IA"
ENDD
M->MAIN__S=M->MAIN__S+1
ENDD
M->MAIN__S=1
DO WHIL M->MAIN__S<M->MAIN__JBK
? MAIN__BK(M->MAIN__S)
M->MAIN__S=M->MAIN__S+1
ENDD
M->MAIN__SSS=1
DO
WHIL
M->MAIN__Q<=M->MAIN__Z.AND.M->MAIN__SSS<=M->MAIN__H
?M->MAIN__ZX
GO M->MAIN__Q
?M->MAIN__BX
M->MAIN__QQ1=MAIN__PZD(1,1)
M->MAIN__QQ2=MAIN__PZD(1,2)
M->MAIN__QQ3=MAIN__PZD(1,3)
? ?
PSTR(M->MAIN__QQ1,M->MAIN__QQ2,M->MAIN__QQ3)
M->MAIN__S=24 DO
WHIL
M->MAIN__S<=M->MAIN__GS
??"|"
M->MAIN__QQ1=MAIN__PZD(M->MAIN__S,1)
M->MAIN__QQ2=MAIN__PZD(M->MAIN__S,2)
M->MAIN__QQ3=MAIN__PZD(M->MAIN__S,3)
? ?
PSTR(M->MAIN__QQ1,M->MAIN__QQ2,M->MAIN__QQ3)
M->MAIN__S=M->MAIN__S+1
ENDD
?? M->MAIN__BX
M->MAIN__SSS=M->MAIN__SSS+1
M->MAIN__Q=M->MAIN__Q+1
ENDD
M-MAIN__S=1
DO WHIL M->MAIN__S<M->MAIN__JDH
? MAIN__DH(M->MAIN__S)
M->MAIN__S=M->MAIN__S+1

```

```

ENDD
?""4M-MAIN__SS=M->MAIN__SS+1
SET ALTE OFF
SET CONS ON
SET PRIN OFF
WAIT "按任意键继续..."
ENDD
IF M->MAIN__OD="2"
CLOSE ALTE
CLEA ALL
ENDI4RETU
函数 PSTR 清单如下:
*(函数 PSTR.PRG 程序清单)
PARA M->PSTR__N1,M->PSTR__N2,M->PSTR__N3
M->PSTR__NN=M->PSTR__N1
*判断类型并作相应处理
M->PSTR__TP=TYPE("&PSTR__NN")
DO CASE
CASE M->PSTR__TP="D"
M->PSTR__NN=DTOC(&PSTR__NN)
IF LEN(M->PSTR__NN)<M->PSTR__N2
M->PSTR__NN=M->PSTR__NN+SPAC(M->PSTR__N2-LEN(M->PSTR__NN))
ENDD
CASE M->PSTR__TP="N"
IF ROUND(&pstr__nn,m->pstr__n3)=0
M->PSTR__NN=SPAC(M->PSTR__N2)
ELSE
M->PSTR__NN=STR(&PSTR__NN,M->PSTR__N2,M->PSTR__N3)
ENDD
CASE M->PSTR__TP="C"
IF LEN(&PSTR__NN)>M->PSTR__N2
M->PSTR__NN=LEFT(&PSTR__NN,M->PSTR__N2)
ELSE
M->PSTR__NN=&PSTR__NN+SPAC(M->PSTR__N2-LEN(&PSTR__NN))
ENDD
ENDD
*按表框宽度进行截尾
M->PSTR__NN=SUBS(M->PSTR__NN,1,M->PSTR__N2)4RETU M->PSTR__NN

```