

FoxBASE 应用技巧

朱新河 (武汉水运工程学院)

摘要: 本文结合实际应用,介绍了 FoxBASE 应用中的两个技巧:一是介绍一种在 FoxBASE 下打印报表时,不打“零”的方法;二是介绍在 FoxBASE 下画图的新方法。

一、前言

FoxBASE 目前在各行各业的管理事物中,得到了广泛的应用,其中一个重要的内容就是打印报表及进行各种图形曲线的绘制。打印报表时往往要求对零值(即空值)不进行打印;对绘图而言,FoxBASE 本身的绘图功能又比较差。为了解决这两个问题,本文对它们进行了初步的探讨。

二、FoxBASE 下不打“零”的方法

本文介绍的不打“零”方法是基于字段类型转换基础之上的,即将数字型字段转换成字符型字段。其具体方法为:

1.首先将要打印的报表中所涉及的字段及相应的记录都汇总到一个中间库中,设中间库的名称为:ZJK.DBF;

2.将 ZHK.DBF 中的所有数字型字段转换成相应长度的字符型字段,设转换后的库名为:ZJK1.DBF;

3.将 ZJK.DBF 文件拷贝成相应的 TXT 文件;

4.将 TXT 文件中的每一条记录转到 ZJK1.DBF 中,经过这种转变,ZJK.DBF 中的记录就传送到 ZJK1.DBF 中,只是其中个别字段的类型发生了变化,而字段的长度没有变化。

进行这种转换,有两个好处:

(1)它可以实现不打“零”,使打印出的报表美观大方;

(2)在打表程序中,省去了 STR(数据字段名,长度)转换函数命令,简化了打表程序。

其实现程序如下:

```
PROC ZH
SET TALK OFF
SET SAFE OFF
USE ZJK
```

```
COPY TO A__A STRU EXTE
USE A__A
GO TOP
X=1
DO WHILE .NOT.EOF()
    IF FIELD__TYPE="N"
        H=LTRIM(TRIM(STR(X)))
        REPL FIELD__TYPE WITH "C"
        * * * * * 记录数字型字段名
        Z&H=TRIM(FIELD__NAME)
        X=X+1
    ENDI
    SKIP
ENDD
USE
CREA ZJK1 FROM A__A
!DEL A__A.DBF
USE ZJK
COPY TO C__C SDF
USE ZJK1
APPE FROM C__C SDF
G=1
DO WHILE G<=X-1
    GO TOP
    DO WHILE .NOT.EOF()
        H=TRIM(LTRIM(STR(G)))
        Z__D=Z&H
        IF VAL(&Z__D)=0
            REPL &Z__D WITH ""
        ENDI
        SKIP
    ENDD
    G=G+1
ENDD
USE
!DEL C__C.TXT
SET SAFE ON
RETU
```

例如:转换前的数据库(ZJK.DBF)为:

Structure for database: D:\ZJK.DBF

Number of data records:4

Date off last update:11 / 15 / 93

Field	Field Name	Type	Width	Dec
1	YYBH	Character	10	
2	HYSYS	Numeric	6	
3	YW	Numeric	5	1
4	SW	Numeric	5	1
5	QYSJ	Date	8	
6	QYZT	Character	12	
7	SYSJ	Date	8	
** Total **			55	

其中的记录为:

Record#	YYBH	HYSYS	YW	SW	QYSJ	QYZT	SYSJ
1	JC201-001	1	0.0	0.0	06 / 05 / 91	怠速	06 / 05 / 91
2	JC201-002	2	0.0	0.0	08 / 03 / 91	怠速	08 / 03 / 91
3	JC201-003	3	3.0	3.0	09 / 20 / 91	怠速	09 / 20 / 91
4	JC201-004	4	0.0	4.0	04 / 16 / 91	怠速	10 / 16 / 91

经过转换后的数据库(ZJK1.DBF)为:

Structure for database:D:\ZJK1.DBF

Number of data records: 4

Date off last update:11 / 15 / 93

Field	Field Name	Type	Width	Dec
1	YYBH	Character	10	
2	HYSYS	Character	6	
3	YW	Character	5	
4	SW	Character	5	
5	QYSJ	Date	8	
6	QYZT	Character	12	
7	SYSJ	Date	8	
** Total **			55	

其中的记录为:

Record#	YYBH	HYSYS	YW	SW	QYSJ	QYZT	SYSJ
1	JC201-001	1			06 / 05 / 91	怠速	06 / 05 / 91
2	JC201-002	2			08 / 03 / 91	怠速	08 / 03 / 91
3	JC201-003	3	3.0	3.0	09 / 20 / 91	怠速	09 / 20 / 91
4	JC201-004	4		4.0	04 / 16 / 91	怠速	10 / 16 / 91

可见,经过转换后,ZJK.DBF 中数字型字段(HYSYS、YW、SW)为零的记录,经过转换后(ZJK.DBF),其相应的记录位置,零值不存在,因此用该数据库去打印报表时,就可实现零值的不打印。

三、图形曲线的绘制与输出

使用 FoxBASE 进行数据管理时,一方面常用报表进行输出;另一方面,还常用图形曲线进行直观分析。目前大

多数用户是在 FoxBASE 下调用 BASIC 语言或其它画图能力较强的语言进行图形、曲线的绘制与输出,这样做至少有以下几个缺点:

- 1.编程复杂,图形复杂时,编程量很大;
- 2.绘制的图形一般质量不是很好;
- 3.调用其它语言,对计算机的内存要求较高。

不久前推出的与 FoxBASE 可配套使用的 FoxGRAPH 软件,是解决这个问题的很方便的方法。它与 FoxBASE 的接口程序非常简单,对内存的要求也很低。它既可绘制二维图形(15 种),又可绘制三维图形(32 种)以及 16 种不同的视角,图形、曲线的色彩及大小等可任意选择。它不仅可在多种显示设备上输出,而且能在打印机和绘图仪上输出,如果打印机和绘图仪是彩色的,则可输出彩色图形。

在 FoxBASE 下直接调用 FoxGRAPH 的接口程序进行绘图,需要用户选择很多的内容,对用户来讲,显得系统的用户界面不太友好,为了解这个问题,可以在程序中直接调用 FoxGRAPH,这样可以使系统的界面非常的友好,给用户的感觉是数据文件的管理与数据的作图是融为一体的。用户只要将要作图的字段汇总到一个绘图用的数据库中,此后用户不要作任何选择,就可以得到 FoxGRAPH 绘制的三维或二维图形。这个作图过程,需要程序设计人员作以下几项工作:

- 1.建一个画图的数据库(图形涉及的各个字段都包含在这个数据库中),设库名为 HT.DBF;
- 2.在 HT.DBF 库中设计虚拟数据;
- 3.启动 FoxGRAPH 宏功能,记录用上述虚拟数据进行画图的每一个操作步骤(每一次的击键),并保存在一个磁盘文件中(设文件名为 HT.3MC);
- 4.在进行实际工作时,往 HT.DBF 装入实际的画图数据;
- 5.在程序中,用下述命令自动画图:
DO GPLUS WITH "HT"
这个过程无须用户的任何干预,即可绘制出预定的图形。

以上,只是用 FoxGRAPH 绘图的一种方法,还有其它多种方式可以选择。