

计算机数据库丛书

学苑出版社

MICROSOFT FOXPRO 2.5 PROGRAMMING

程序设计技术



Foreword by
Walter Kennamer,
Software Design
Engineer, Microsoft
Corporation

Les Pinter



希望

TP311.13
8

计算机数据库丛书

Microsoft FoxPro 2.5 Programming

FoxPro 2.5 程序设计技术

Les Pinter 著

周南木杉等译

任天 审校



学苑出版社

(京)新登字 151 号

内 容 提 要

本书对 FoxPro 工具、技巧与实例作了系统介绍,并对事件驱动编程和菜单驱动编程作了重点阐述。本书内容丰富,集技巧性与实用性于一体,是初学者的良师,有经验编程人员的益友。并可作为大中专院校学生、科研人习使用 FoxPro 的教材和参考用书。

欲购本书的读者可直接与北京海淀路 8721 信箱书刊部联系。邮政编码:100080,电话:2562329。

版 权 声 明

本书英文版名为《Microsoft FoxPro 2.5 Programming》,由 McGraw-Hill 出版公司出版。版权归 McGraw-Hill 有。本书中文版由 McGraw-Hill 授权出版。未经出版者书面许可,本书的任何部分均不得以任何形式或任何手制或传播。

计算机数据库丛书

FoxPro 2.5 程序设计技术

著 者:Les Pinter
译 者:周 南 木 杉 等
审 校:任 天
责任编辑:甄国宪
出版发行:学苑出版社 邮政编码:100036
社 址:北京市海淀区万寿路西街 11 号
印 刷:北京宏伟胶印厂印刷
开 本:787×1092 1/16
印 张:20 字数:461 千字
印 数:1~5000 册
版 次:1994 年 10 月第 1 版第 1 次
ISBN 7-5077-0886-1/TP·28
本册定价:34.00元

学苑版图书印、装错误可随时退换

用户请注意

欲购本书配套软盘的朋友,请按下列方法汇款:

单价:50.00元(含邮费)

注:从银行电汇款的朋友请按下列帐号和收款单位汇款:

收款单位:北京希望电脑公司	
开户银行及帐号:北京海淀工商行中关村城市信用社	帐号:05079—08
注:要增值税发票的朋友,请仔细填下表。	

购 货 单 位	名称	纳税人登记号
	地址电话	开户银行及帐号

注:本次共订盘 张 应收款为¥(小写):

注:用户填好此单后请连同此单、信汇单一并传真 01—2561057 或 01—2579874,收到传真后即发货。或邮寄 100080 北京海淀 8721 信箱资料部朱红收
联系电话:01—2562329,2541992

声 明

本书所介绍的技术已发展成熟了。书中引用的菜单驱动例子取自 Luis Castro 的 ViewGen、FoxView 和 Stage 中的范例。事件驱动编程范例的来源有两个：一是由 Dave Fulton 博士、Walt Kennamer 及其他 MicroSoft\Fox 小组成员所编写的应用程序，另一是 Alan Griver 和 Tom Rettig 所开发的软件。这些应用软件所涉及的概念比本书所介绍的要深刻得多，读者可以对其作进一步的研究。

如果读者不想将各范例程序的代码逐一输入，则可以订购本书附加的代码盘，也可以订阅《Pinter FoxPro Letter》，里面介绍了本书有关出版物的摘要。工作通讯有助于读者跟上不断更新的软件概念，特别适合于 FoxPro 这类变化很快的环境。另外，在 Moscow 还出版有俄文版的工作通讯。

译 序

随着计算机数据库管理软件技术的不断发展, FoxPro 已成为微机上数据库管理软件中的首选对象。由于 FoxPro 和 FoxBASE 与 C 完全兼容, 并在它们的基础上增添了许多新的功能, 因此 FoxPro 逐渐在微机数据库管理领域中拥有了越来越多的用户, 并越来越受到广大用户的重视。这是我们译这本书的目的。

通常, 学习编程语言的最好办法是研究已有的程序代码并修改已有代码以适应自己的需要。本书作者把大量的篇幅放在源程序上, 这是相当明智的。采用生动浅显的方式讲解如何用 FoxPro 语言进行程序设计是本书的一大特色。作者给出了菜单驱动编程和事件驱动编程的区别, 并详细介绍了每一种编程技术的长处和不足。事实上, 本书的主题之一就是讲述这种区别, 大量例子中绝大多数是围绕这些区别给出的, 这些例子对初学者极有帮助。

本书由周南和木杉翻译, 任天负责审校, 参与本书翻译工作的还有张东灵、刘彬、章强、付国勇、刘小东、林得利和张小莉。

译者

1994 年 10 月

目 录

引 言	(1)
0.1 FoxPro 历史	(1)
0.2 FoxPro 如何参与竞争	(1)
0.3 事件驱动软件与菜单驱动软件	(2)
0.4 如何使用本书	(2)
0.5 一种不同的方法	(3)
第一章 编程范例	(5)
1.1 文件结构	(5)
1.2 数据输入 I/O	(8)
1.3 窗口 (Windows)	(10)
1.4 弹出菜单 (Popups)	(10)
1.5 运行程序和函数	(11)
1.6 控制结构	(12)
1.7 强有力的工具	(14)
1.8 色彩	(23)
1.9 出错抑制	(24)
1.10 多用户编程	(24)
1.11 小结	(25)
第二章 编写一个应用软件	(27)
2.1 原始资料	(27)
2.2 放在何处	(27)
2.3 一个简单的应用软件	(28)
2.4 屏幕设计	(30)
2.5 FoxPro 的生成代码	(36)
2.6 是字段还是记录变量	(36)
2.7 重新排列 GET 字段	(36)
2.8 给屏幕加开关	(38)
2.9 关于获取记录变量的 GET 字段	(41)
2.10 关闭编辑功能	(42)
2.11 数据输入确认	(44)
2.12 建立项目文件	(45)
2.13 小结	(48)
第三章 事件驱动的多屏幕范例	(49)
3.1 增加一个屏幕	(49)
3.2 统一的开关	(55)
3.3 应用屏幕的 CLEANUP 代码	(63)

3.4	FoxPro 屏幕集的用途	(65)
3.5	应用屏幕软件的控制	(66)
3.6	窗口函数	(68)
3.7	READ ACTIVATE 语句	(70)
3.8	修改 MAIN.PRG 和 MENU.MPR 程序	(70)
3.9	MAIN.PRG 新的全局变量	(71)
3.10	修改事件句柄	(71)
3.11	控制开关	(72)
3.12	项目文件	(72)
3.13	小结	(73)
第四章	事件驱动软件的加强	(75)
4.1	再加入一个屏幕	(75)
4.2	增加一个弹出菜单	(76)
4.3	给事件驱动系统增加浏览 (BROWSE) 功能	(83)
4.4	小结	(99)
第五章	菜单驱动举例	(101)
5.1	操作原理	(101)
5.2	哪些做法会导致错误	(101)
5.3	恰当的例子	(101)
5.4	一个菜单驱动应用软件	(102)
5.5	折衷方案	(102)
5.6	菜单文件的修改	(103)
5.7	主程序的修改	(105)
5.8	屏幕的修改	(109)
5.9	综合的观察结果	(130)
5.10	小结	(130)
第六章	屏幕设计中的列表管理	(133)
6.1	设计自己的屏幕显示函数	(133)
6.2	滚动列表	(133)
6.3	口令系统	(133)
6.4	存取函数	(134)
6.5	多项选择列表	(152)
6.6	小结	(165)
第七章	编写自己的工具	(167)
7.1	设计自己的 BROWSE	(167)
7.2	保存历史数据	(179)
7.3	带有温度计的报表	(182)
7.4	小结	(190)
第八章	特别的结算应用软件	(191)
8.1	应用软件的设计	(191)

8.2	应用软件代码	(196)
8.3	小结	(230)
第九章	专门的数据转换应用软件	(231)
9.1	基本方法	(231)
9.2	转换表格	(231)
9.3	项目文件	(233)
9.4	FoxPro 的 FPATH. PLB 库	(234)
9.5	源程序代码	(235)
9.6	HELP 文件	(263)
9.7	使用转换文件	(264)
9.8	非标准输入文件	(264)
9.9	把报表作为输入文件	(265)
9.10	小结	(265)
第十章	其它	(267)
10.1	使用 INKEY () 的 FoxPro 电子表格	(267)
10.2	标签打印.....	(278)
10.3	使用单一的确认表格.....	(283)
10.4	代码/姓名查找	(285)
10.5	销售额图形.....	(291)
10.6	滚屏控制.....	(295)
10.7	小结	(311)

引言

1979年,我受命主持小型商务应用软件(Small Business Applications)的管理。它由两个人发起,并且开发出了最早的用于微机的文字处理程序之一的 Magic Wand。两个出色的波士顿人, Bill Radding 和 Mike Griffin,用了一年多时间攻克了这个课题,并准备将其推向市场。

Jerry Pournelle 在《Byte》杂志上发表了一篇评论,说“Magic Wand 胜过 Wordstar”。我们大获成功,在八个月内卖出了超过了一百万美元的软件,而且增加了八名雇员。

一天, Microsoft 公司二十三岁的董事长, Bill Gates 告知我,他想购买我们的源程序,第二天,交易就做成了。Bill Gates 到了休士顿,给了我们一张支票和一份不参与竞争的合同,就回了西雅图。去年,在西雅图的一次聚会上,比尔告诉我,我的合同已经到期,这就是读者现在能看见这本书的原因。

我曾经尝试过做主体程序设计,但是那个小程序的程序令人难忘。于是我投入到 Lotus 1.2-3 中,写了第一个 Lotus 添加软件(房地产指导)。最后,我开始用 dBASE 做软件,其后的进展便如乘坐火箭一般飞快。

0.1 FoxPro 历史

FoxPro 是七十年代初, Wayne Ratliff 在喷气动力实验室工作时创立的。这种语言的独特之处是,它把文件的描述作为一个标题放在文件本身中。这样,当程序打开一个文件时,可以读出文件描述:名字、类型、字段长度,以及记录的数目等。因为这一创新,在程序中就不必对每个文件逐一描述。

这一创新产生了一种非程序员也可以很快理解并使用的语言,第一次使其它领域的专业人员能够建立自己的数据库。当然,这种开放有时会导致程序与课本上的技术规则不完全相符,但不管怎么说,它向非程序员敞开了编程的大门。

dBASE 语言受到了普遍的好评和广泛的应用,并且象细胞繁殖一样,一代更比一代强,最后发展成为功能更好的 FoxBASE。我是在为拍摄《星球大战》三部曲的露斯卡电影公司制做一份合同时发现它的。我的 dBASE 程序在 FoxBASE 下运行时,速度居然提高了 7 倍,我立刻改用了 FoxBASE。

FoxBASE 变成 FoxPro,于是引起了一场“战争”。最初,是在法律上的战争,最终以 XBASE 不具版权结束。从而把战争转向技术领域。后来, Microsoft 买下了 Fox 软件公司,技术竞争变成 Microsoft 公司的内部事务。竞争的结果是推出了 FoxPro 2.5—Microsoft 公司的第一代 XBASE 产品。于是我又遇上了 Bill Gates,这是一次对双方都有益的合作。

0.2 FoxPro 如何参与竞争

FoxPro 仍有少数几个竞争者。最大的两个竞争者是 Clipper 和 dBASE IV 除此之外,还有其它几个也一直占有一定的市场。Clipper 更适合于训练有素的程序员,它能生成 EXE 文件,

这种文件一度显得太长,但现在跟 FoxPro 生成的 1M 多的巨型文件比起来,就显得微乎其微了。Clipper 程序看起来就象亚美尼亚文。尽管我从 1984 年起就编写 XBASE 程序,但还是几乎不懂 Clipper 程序。所以 Clipper 的路线,使初级和中级程序员不免有些望而生畏。

dBASE IV 的情况怎样呢?推出 dBASE IV 的 Borland 公司曾推出 Turbo Pascal,这是第一个不需退出 Dos 环境就能进行编辑、编译和检验的集成开发环境,Borland 公司多次以高质量和低价格占领市场。

如果 Borland 公司仍然想延续过去的成功之路,他们必须双管齐下:既要保持技术上的领先,又要有低于 FoxPro 一半的价格,否则,就只有退出市场了,只依靠商业广告是无法与 Microsoft 公司竞争的。

我个人希望 Borland 公司能坚持以往的原则。与推出 Clipper 的 Aston—Tate 公司竞争,使 Fox 产品处于领先地位,虽然没有竞争,我们也许能从垄断中获得更多的利益。事实上,Dave Fulton 和他的集体可以随时取代 Borland 公司。Microsoft 公司会更更多地了解用户的需求,并且提供给用户远比其它公司更多的服务。

0.3 事件驱动软件与菜单驱动软件

FoxPro 要面对的一个重要的问题就是:采用传统的菜单驱动技术,还是采用在 Windows 环境下占优势的事件驱动技术。对于 FoxPro2.5 仍旧存在这个问题。事件驱动不仅仅是 Windows 独有的技术。事实上,DOS 下的 FoxPro 和 Windows 下的 FoxPro 只有一点小小的差别,而编写在两种环境下事件驱动软件的方法几乎是相同的。那么如何作出选择呢?本书中同时给出了用两种技术编写的应用程序,读者只需确定用户的要求,并且实现他们的要求。小型的事件驱动软件很容易编写。大型的商业软件本身就很难编写,而大型的事件驱动软件的编写就更困难了。事件数目的增加,会使编程的困难程度以指数形式增加。

0.4 如何使用本书

认识所有的英语词汇并不一定能写出象莎士比亚一样的文章,词语只有在上下文的句子中和段落间才能表达意义。同样 FoxPro 的命令和函数只有放在用户定义的函数、过程和应用软件中才能有意义。

本书是面对想查找工具、技巧和编程实例的程序员。如果是 FoxPro 的初学者,还需要先浏览一下 FoxPro 的命令和函数。这本书里跳过了介绍性的章节。直接将命令和函数用于建立模块和例程,所以对于初学者有些超前。这里推荐 Tom Rettig, John Hawkins, Miram Liskin 以及其他作者的初级书目。当然,这些作者同样对 FoxPro 的高级应用很在行,但他们的书中介绍了本书没有提供的基础知识。

读者可以把这本书作为参考书,但除非深入下去,不然难免有遗漏之处,甚至连我也记不清所有的程序了,读者可以先找出最感兴趣的部分,然后把它拿出来,仔细研究。

有关专家指出。单独记忆理论要比记忆大型结构中的一部分要难得多。在中世纪,学习就通常像在一个“记忆体育馆”中进行。通常最下面的一层是想象的千屋大会堂。在这里可以把理论与实际相联系。即使是虚构框架,或是中世纪的学者,都能让人联想起一些理论。所以学习 FoxPro 的命令和符号的最好的办法是结合程序进行,从运行程序开始,将感兴趣的部

分拿出来运行,看它是如何工作的,这样就能在有意义的环境中学会 FoxPro 的命令和函数了。

0.5 一种不同的方法

每个月我都能读到关于面向对象的程序设计方法和事件驱动技术的文章。这是一个使它的作者处于尖端的课题。我的朋友 Alan Schwartz 说这是开发 FoxPro 的“流血的刀刃”。我接到许多读者的电话,说他们对此一窍不通,希望给予解释。

在雷斯大学学习经济课程时,我曾是班上的最后几名,也正是因为这段经历,使我对于普通程序员遇到这类困难感觉有很深刻地体会。通过与我的工作通讯的日常的订户联系,我清楚地知道大多数程序员想要什么。他们想要好的、实在的、有用的、完整的应用软件,我将努力满足这些要求。

关于事件驱动技术的章节,先从简单的情况讲起,然后逐渐增加技巧。先说明为什么有些东西丢失了,然后增加代码。希望读者能够满意地看完每个 DEACTIVATE、VALID 和 WHEN 子句。

第五章给出了菜单驱动模式。与第四章的事件驱动模式相比较,它们的不同之处一部分是外表的,一部分是形式上的,最重要的不同在于需要的劳动量。我强烈地感到,许多时候在事件驱动软件上花费太多,是不划算的。读者可以自己判断。

第一章 编程范例

每种计算机语言都有其特有的规定,如赋值、条件分支、迭代、输入/输出等等。但是与 C 或 COBOL 不同, FoxPro 在怎样编写程序方面的特别规定很少。XBASE 正是因为这个原因成为最受欢迎的语言。许许多多想自己编程的用户,往往因为看不下去 BASIC 或 COBOL 冗长的介绍,而以失败告终。后来,他们遇到了 FoxBASE,发现使用起来毫无障碍。

有些程序员刻意地象编写 C 程序那样使用 FoxPro 语言,这完全没有必要, XBASE 系列正是以其简单取胜的。

1.1 文件结构

FoxPro 应用软件中有几种文件,程序代码文件、数据文件(表格)、索引和其它组成部分。

1.1.1 程序文件

FoxPro 的程序通常是具有 .PRG 扩展名的 ASC II 文件。程序以赋值和打开文件开始,然后就是完成各项功能的语句体,被调用的函数附在程序的后面,示例如下:

```
* Main program
* program code
FUNCTION first
* program code
FUNCTION second
* program code
```

调用一个函数时, FoxPro 会到当前程序的末尾查找函数代码。在 FoxPro 中,许多地方不是把代码按顺序写出,而是将其插入叫做代码片段的文件中。这一章后面介绍屏幕生成工具和菜单生成工具时,将介绍如何插入这类代码片段,然后将代码放入其间适当的位置。

这些代码片码中最重要的大概要算 CLEANUP 片段了。放在 CLEANUP 片段中的内容被安置在合成代码的最后。若想把所有的函数都放在片段中,就需要在 CLEANUP 片段中加入一段入口参数程序,这段程序将被放在最后,即 FoxPro 查找函数的地方。

渐渐地,程序中会有大量的能实现各种功能的函数,把它们都装进 CLEANUP 片段当然可以,不过还有更好的办法。可以这样:把一些函数和过程装入一个叫 MYPROCS 的文件中,并在软件中用," SET PROCEDURE TO MYPROCS" 来声明这个函数库的存在。如果调用一个不包含在文件尾部的函数时, FoxPro 就会到用" SET PROCEDURE TO MYPROCS" 创建的函数列表中寻找该函数并执行。必要的时候,可以用" DO functionname IN file name", 来告诉 FoxPro, 让其快速到文件中查找函数,并执行。不过,这种办法通常是不采用的,因为执行速度慢,但是当调用菜单中包含的函数时,就有必要这样做了。比如,对于 DO MENUHIT IN MYMENU.MPR 来说,无论正在执行什么程序或正在打开哪个文件, FoxPro 都会执行菜单选项。

1.1.2 数据文件

和其它 xBASE 语言相同, FoxPro 的数据库文件的扩展名是 .DBF。如果一个数据库文件中包含一条以上的索引, 通常会派生出与 .DBF 文件名相同, 但扩展名为 .CDX 的辅助结构索引文件。如果一个数据库文件中包含一个以上的备注字段, 通常会有一个与 .DBF 文件名相同的, 但扩展名为 .FPT 的辅助备注文件。

1.1.3 标题

数据文件有一个 32 字节的标题, 其后的 32 字节是对文件各字段的描述, 最后用一个结束字结束, 标题一旦被破坏, 即使数据仍完好, 也没法读该文件了。万一文件被破坏, FoxPro 会把文件字节复位为 0, 但仍保持数据不变。如果 FoxPro 找不到 .CDX 文件, 就会在第 29 字节置一个 "1", 程序将返回出错信息。

这类事件在编程过程中, 都有可能发生。不用担心, 以后的章节里, 将介绍一些文件修复的方法。

1.1.4 字段和记录

FoxPro 有五种类型的字段:

- 字符型 (Character)
- 数字或浮点型 (Numeric or floating point)
- 逻辑型 (Logical)
- 日期型 (Date)
- 备注、图像或通用型 (Memo, picture 或 general)

后三种类型往往存在 .FPT 文件中, 在 .DBF 中最多占有 10 个字节。逻辑型几乎不使用, 因此文件中通常只有四种类型的数据。FoxPro 在 CREATE TABLE 语句中将数据列成八种类型, 虽然每种类型都有差别, 但通常不用去理会。

1.1.5 删除标志位

一条记录的长度, 是它的所有字段的长度和, 再加上一位删除标志。除非记录被再填写, 否则 FoxPro 并不真正删除一条记录, 只是在删除标志位中写一个 * 星号。函数 DELETED() 会从这些记录中返回一个真值。语句 SET DELETED ON 会使这些记录消隐。

编写应用程序遇到的一个重要问题, 就是如何处理这些被删除的记录, 在复合索引出现之前, 记录再生技术已经完全解决了删除和填充这个令人头痛的问题。复合索引和 Rushmore 技术给出了一个更好的解决办法, 此书的后面将介绍这种方法。

1.1.6 索引

FoxPro 可以使文件指针按文件的物理顺序移动: 记录 1, 记录 2, ……。也可以在索引特征位加任意数字, 使其按照任何方式移动。

在 FoxPro V1.0 时, 每条索引都放在数据库文件中。由于一条索引被移去后, 记录会增加或删除, 这就给在软件中反复索引带来很多问题, 良好的编程经验也解决不了这个问题。记住 Murphy 法则!

在 V2.0 和 V2.5 中, 索引放在单独的 .CDX 文件里, 因此任何索引不会被遗漏。 .CDX 文件中可以保存比用户所需要的更多的索引。用命令: INDEX ON expression TAG name 能够增加一条索引。而用 SET ORDER TO TAG name 可以切换主索引, 主索引将告诉 FoxPro, 下一条记录是哪一个。例如, 对于下面这样的存有人名的文件:

```
Record # Name
1      BAKER
2      ALLEN
3      CHARLES
```

并且在 Name 字段有一个按字母顺序的索引, 那么命令: SET ORDER TO TAG NAME 会将文件的顺序变成:

```
Record # Name
2      ALLEN
1      BAKER
3      CHARLES
```

不能在一条索引中, 同时使用不同类型的表达式, 即不能够用 NAME + DATE + AMOUNT 作为索引, 这里 NAME 是字符型, DATE 是日期型, 而 AMOUNT 是数字型的表达式。不过, 可以把索引中不同类型的字段化为字符型的等价值, 用法如下: UPPER (NAME) + DTOC (DATE, 1) + STR (Amount, 10, 2)。

FoxPro 能够有效地提取数据子集, 这给许多应用软件带来了非常大的好处, 只要有可能, FoxPro 就提取索引标签里的信息, 因为索引可以存在内存中, 这就能大大地提高速度。快速查找的窍门就在于使用了与索引标签相同的方式来查阅数据库。

1.1.7 删除记录再生

上文提及过删除标志位。删除 (DELETE), 彻底清除 (Pack) 和恢复一条记录的操作在网络上是非常非常慢的。因此, 有创意的记录再生技术得到了发展, 并且越来越受欢迎。在可修改文件的后面加上: INDEX ON DELETED () TAG DELETED。如果让 DELETED () 成为一条索引, 那么管理删除记录的再生就变得又快又简单了, 特别是在网络应用中, 这一点就显得非常重要。

1.1.8 FoxPro 内部文件

文件中带有一个标题的想法似乎在日渐消失, 其它语言都不采用了, 但 COBOL 程序员使用它, xBASE 系列也使用它, 可见这并不是个坏注意。FoxPro 的所有内部文件都采用相同的标题结构。FoxPro 把所有的信息: 屏幕、项目、菜单、报告和报表, 甚至最后一次浏览 (BROWSE) 的彩色图解、尺寸和位置, 都放在一个与 .DBF 文件结构相似的内部文件中。

用户可以象操作数据文件一样对内部文件进行操作: 阅读、返回报告、全局查找和替换参数, 或只是浏览。事实上, 这是编程的一个特殊技巧, 可以以最快速度找到有效代码, 从而在屏幕上移动和插入内容到屏幕文件中。

1.2 数据输入 I/O

FoxPro 有三种从键盘获得数据的方式：`@...GET` 和 `READ`, `BROWSE`, 以及 `INKEY` () (菜单和弹出菜单是第四种获得信息的方式, 但是很少用于数据输入)。

1.2.1 @...GET 和 READ

这是 FoxPro 最基本的输入途径, 包括各种形式的语句:

```
@ row, col GET memvar [field
  [FUNCTION expC1]
  [PICTURE expC2]
  [FONT expC3 [, expN1]]
  [STYLE expC4]
  [DEFAULT expr1]
  [ENABLE |DISABLE]
  MESSAGE expC5]
  [ [OPEN] WINDOW window name]
  [RANGE [expr2] [, expr3]]
  [SIZE expN2, expN3]
  [VALID expL1 |expN4]
  [ERROR expC6]]
  [WHEN expL2]
  [COLOR SCHEME expN5 |COLOR color pair list]
```

`GET` 字段按照发出的顺序被堆栈起来, 通过 `READ` 命令可以将其激活。`READ` 按照入栈的先后顺序处理 `GET` 字段。如果 `GET` 字段是在不只一个窗口输入的, 那么处理第一个信息时将选择一个适当的窗口作为最优先的。

第十四条命令变量处理上大不相同, 特别是当 `VALID` 语句或函数使用时, 语句或函数一个执行, 光标就会离开相关变量的 `GET` 区域。这是 FoxPro 应用软件和其它语言应用软件的差别。

FoxPro 在读取数据信息和无效的开关信息时是有区别的。传递给 `PICTURE` 或函数的参数会使 FoxPro 显示出不同类型的 `GET`。特别是 `CHECK BOX`、`LIST` 和 `Radio Buillon` 这类控制会使界面增添一些趣味。

1.2.2 BROWSE

`BROWSE` 这个命令的语法形式如下:

```
BROWSE
  [FIELDS field list]
  [FONT expC1 [, expN1]]
  [STYLE expC2]
  [FOR expL1]
  [FORMAT]
  [FREEZE field]
```