

# FoxPro 2.0

## 应用技术开发



王振诚 编  
董云

孙德和 审校  
宋阳

北京希望电脑公司

# FoxPro2.0 应用技术开发

王振诚 董云 编  
孙德和 宋阳 审校

北京希望电脑公司

## 内 容 提 要

本书详尽介绍了有关 FoxPro2.0 的应用技术开发问题。全书共分十七章。首先介绍 FoxPro2.0 中新增特征，然后介绍数据库设计，数据字典，设计和定制应用，模板语言处理，处理多种类型 FoxPro 数据的过程和技术，索引和排序，报表和打印技术，结构查询语言，低级文件 I/O，多用户过程和技术，最后介绍了应用程序接口以及如何构造和分配应用。

本书内容详尽，可读性好，可作为广大管理人员和大专院校师生的自学教材和参考书。

## FoxPro2.0 应用技术开发

王振诚 董云 编  
孙德和 宋阳 审校

京准印字：3572—91572

内部成本：15.00 元

## 前 言

随着数据库管理技术在各个行业的普及应用，市场上出现了多种数据库系统，其中 FoxPro 在我国拥有众多的用户。FoxPro2.0 是 FoxPro 的最新版本，新增了许多特征。

本书用十九章的篇幅介绍了开发 FoxPro2.0 应用的技术，并提供了工具和信息。书中的许多资料都是 FoxPro 手册和其它同类书籍中所没有的。书中讨论了数据库设计技术，处理各种类型 FoxPro 数据的过程和技术，多用户过程和技术。介绍了怎样利用 FoxPro 开放结构的最大优点，讨论如何通过应用程序接口(API)设备扩展 FoxPro 语言。本书还介绍了 FoxPro 革命性的 Rushmore 技术，并在最后一章介绍了如何构造和分配 FoxPro 应用。

本书在编写过程中得到王明、张启鹏、杨阳和董婉秋同志的大力支持和帮助，在此向他们表示感谢。由于水平有限，错误或不妥之处在所难免，恳请读者批评指正。

编 者

# 目 录

(14)	.....	.....
(54)	.....	.....
(54)	第一章 FoxPro2.0 新增内容 .....	(1)
(54)	FoxPro2.0 .....	(1)
(54)	开放结构 .....	(1)
(54)	技术 Rushmore .....	(2)
(54)	编译器及项目管理 .....	(2)
(54)	外部例程 API .....	(3)
(54)	类似 dBASE IV 的特征 .....	(4)
(54)	SQL SELECT 和 RQBE .....	(4)
(54)	FoxPro 屏幕和菜单生成器 .....	(4)
(54)	宏及 EVALUATEO .....	(6)
(54)	模板语言 .....	(6)
(54)	新的及增强的 READ / GET 命令及函数 .....	(6)
(54)	菜单命令和函数 .....	(8)
(54)	数据输入和输出 .....	(9)
(54)	关于窗口 .....	(10)
(54)	打印机驱动程序 .....	(10)
(54)	其它增强特征 .....	(11)
(54)	本章小结 .....	(11)
(88)	.....	.....
(88)	第二章 数据库设计 .....	(13)
(88)	数据库设计的八步骤 .....	(13)
(88)	创建 FoxPro 数据库 .....	(17)
(88)	关系完整性规则 .....	(17)
(88)	规范化 .....	(19)
(88)	一个完整的例子 .....	(23)
(88)	本章总结 .....	(30)
(120)	.....	.....
(120)	第三章 数据字典 .....	(32)
(120)	什么是数据字典 .....	(32)
(120)	数据字典的优越性 .....	(33)
(120)	建立一个初始的 FoxPro 字典 .....	(33)
(120)	一个实用 FoxPro 数据字典系统 .....	(35)
(120)	数据字典文件 .....	(36)
(120)	数据元素字典 .....	(39)
(120)	数据源 .....	(40)
(120)	扩充字典概念 .....	(40)

本章总结 .....	(41)
第四章 设计和定作应用程序 .....	(42)
(1) 调用其他程序模块 .....	(42)
(1) 建立一组库例程 .....	(45)
(1) 完成数据确认 .....	(47)
(5) 错误处理 .....	(59)
(5) 定制配置 .....	(66)
(3) 本章总结 .....	(69)
(4) .....	
(4) 第五章 模板语言处理 .....	(71)
(4) FoxPro2.0 的模板语言 .....	(71)
(6) 指定目标文件 .....	(72)
(6) 输入生成行 .....	(73)
(6) 模板语言处理步骤 .....	(73)
(8) 模板文处理的应用 .....	(74)
(2) 本章总结 .....	(82)
(10) .....	
(01) 第六章 处理日期和时间数据 .....	(83)
(11) FoxPro 日期/时间的重要特性小结 .....	(86)
(11) 日期操作 .....	(87)
日期比较 .....	(88)
(81) 日期格式 .....	(88)
(81) 空日期 .....	(89)
(51) 日期存储格式 .....	(90)
(51) 存放日期和时间 .....	(93)
(91) 随机处理和系统时钟 .....	(94)
(85) 从定界文件添加日期数据 .....	(94)
(08) 平均日期 .....	(94)
索引日期 .....	(95)
(58) 是闰年吗 .....	(95)
(32) 一个月或一年中还有多少天 .....	(96)
(83) 计算工作天数 .....	(96)
(33) 本章总结 .....	(98)
(28) .....	
(8) 第七章 备注字段技术 .....	(100)
(39) 为什么使用备注字段 .....	(100)
(04) FoxPro 中备注字段提高能力小结 .....	(100)
(44) 生成和存储备注字段 .....	(106)

(107)	显示备注字段的其他方式 .....	(107)
	备注字段长度 .....	(108)
(108)	管理备注字段空间 .....	(112)
(108)	输入和输出备注字段数据 .....	(112)
(108)	用命令输出备注字段数据 .....	(112)
(108)	APPEND MEMO 和幻象记录 .....	(116)
(108)	备注字段作为填充箱 .....	(117)
(108)	绕过丢失的备注文件 .....	(118)
(108)	备注文件毁坏 .....	(118)
(107)	本章总结 .....	(119)
(107)	.....	
(121)	第八章 处理字串和字符数据 .....	(120)
	数据类型转换 .....	(120)
(121)	搜索字符串 .....	(121)
(121)	在数据入口处的字符字段中的字包围 .....	(123)
(121)	串比较 .....	(124)
(121)	SQL 中的串比较 .....	(125)
(121)	替字符和字串 .....	(125)
(121)	字串加密和解密 .....	(127)
(121)	字串动画 .....	(129)
(121)	REPLICATE()函数 .....	(130)
(121)	本章总结 .....	(134)
(121)	.....	
(136)	第九章 数组技术 .....	(136)
(136)	FoxPro 中的数组 .....	(136)
(136)	数据元素寻址 .....	(136)
(136)	数组初始化 .....	(137)
(136)	FoxPro 数组与 Clipper 数组 .....	(138)
(136)	把数组传递给过程 .....	(139)
(136)	固有的数组函数 .....	(139)
(136)	关于数组的 FoxPro 命令 .....	(142)
	把数据移入 / 移出数组和数据库 .....	(143)
(136)	SCATTER / GATHER 与 COPY / APPEND 命令 .....	(147)
(136)	数据排序 .....	(148)
(136)	数组排序技术 .....	(148)
(136)	气泡排序 .....	(148)
(136)	外壳排序 .....	(149)
(136)	快速排序 .....	(150)
(136)	堆排序 .....	(152)

(50)	本章总结 .....	(154)
(80)	.....	
(51)	第十章 索引和排序 .....	(156)
(52)	排序 .....	(156)
(53)	索引 .....	(156)
(61)	数据库搜索 .....	(161)
(62)	形成索引表达式 .....	(166)
(81)	打开索引及索引次序 .....	(168)
(18)	重索引 .....	(169)
(19)	在多用户环境中索引 .....	(170)
	搜索最近的匹配 .....	(170)
(50)	本章总结 .....	(171)
(51)	.....	
(51)	第十一章 报告和打印技术 .....	(173)
(52)	FoxPro 报告编写系统 .....	(173)
(53)	FoxReport 特征总结 .....	(174)
(54)	在 FoxPro2.0 中加强的报告 .....	(175)
(55)	在报告中包含百分率 .....	(176)
(56)	FoxPro 报表格式文件 .....	(176)
(57)	打印备注字段 .....	(176)
(58)	系统内存变量 .....	(177)
(59)	页处理器 .....	(177)
	邮件合并报告 .....	(178)
(60)	蛇形栏报告 .....	(183)
(61)	直接向打印机发送命令 .....	(184)
(62)	向打印机发送 Null .....	(185)
(63)	重定向打印机输出到文件 .....	(186)
(64)	网络上打印 .....	(187)
(65)	打印任务和结束打印任务 .....	(187)
(66)	打印机驱动程序 .....	(188)
(67)	本章总结 .....	(190)
(68)	.....	
(69)	第十二章 SQL 关联性 .....	(191)
(70)	SQL 引言 .....	(191)
(71)	SELECT 命令 .....	(191)
(72)	SQL 函数 .....	(198)
(73)	RQBE .....	(201)
(74)	CREATE TABLE .....	(201)
(75)	INSERT INTO .....	(202)



(07) 本章总结 .....	(203)
(08) 第十三章 低级文件 I/O .....	(205)
(17) 与 C 文件 I/O 函数比较 .....	(206)
(09) 低级文件 I/O 的使用 .....	(206)
(09) 本章总结 .....	(220)
(80) .....	
第十四章 多用户过程及技术 .....	(222)
共享文件 .....	(222)
FoxPro 隐含锁定 .....	(222)
在网络上删除记录 .....	(226)
控制同时更新 .....	(227)
死锁 .....	(236)
本章总结 .....	(240)
第十五章 附加的多用技术 .....	(241)
报表 .....	(241)
记事处理 .....	(243)
ON ERROR 处理 .....	(246)
维护系统检查跟踪 .....	(247)
其他多用户控制 .....	(249)
本章总结 .....	(252)
第十六章 应用程序界面 .....	(254)
FoxPro API 基础 .....	(254)
编译及连接 API 例程 .....	(255)
从 FoxPro 获取参数 .....	(256)
把结果送回 FoxPro .....	(259)
API 例程总结 .....	(260)
本章总结 .....	(265)
第十七章 建立及分配应用 .....	(267)
项目管理 .....	(267)
使用 ENCRVPT 和 NODEBUG 选项 .....	(269)
优化执行 .....	(269)
编写应用 Conflg.fp 文件 .....	(271)
压缩分配文件 .....	(271)
安装你的应用 .....	(273)
应用物品特别报表及数据库安全性 .....	(273)

(279)	本章总结 .....	(279)
(280)	附录 A .....	(280)
(287)	附录 B .....	(287)
(293)	附录 C .....	(293)
(299)	附录 D .....	(299)
(308)	附录 E .....	(308)
(322)	第十四章 多用户系统 ..... 本章总结 .....	(322)
(322)	共享文件 .....	(322)
(322)	FoxPro 语言概述 .....	(322)
(326)	在网络上安装 .....	(326)
(327)	控制同时连接 .....	(327)
(326)	其他 .....	(326)
(340)	本章总结 .....	(340)
(341)	第十五章 附加的应用技术 .....	(341)
(341)	概述 .....	(341)
(343)	事务处理 .....	(343)
(346)	ON ERROR 处理 .....	(346)
(347)	维护系统检查 .....	(347)
(349)	其他多用户控制 .....	(349)
(352)	本章总结 .....	(352)
(354)	第十六章 应用程序界面 .....	(354)
(354)	FoxPro API 基础 .....	(354)
(355)	编译及连接 API 图标 .....	(355)
(356)	从 FoxPro 获取参数 .....	(356)
(359)	把结果送回 FoxPro .....	(359)
(360)	API 图标总结 .....	(360)
(362)	本章总结 .....	(362)
(367)	第十七章 建立及定制应用 .....	(367)
(367)	项目管理 .....	(367)
(369)	使用 ENCRVPT 和 NODEBUG 选项 .....	(369)
(369)	批处理 .....	(369)
(371)	编译应用 Config.p 文件 .....	(371)
(371)	创建分版文件 .....	(371)
(373)	受保护的数据库 .....	(373)
(373)	应用物品特别报表及数据库安全性 .....	(373)

# 第一章 FoxPro2.0 新增内容

FoxPro2.0 引入了强大的 Dbase 语言的增强功能。本章中，我们简明地讨论 FoxPro 的新特征。在后序章节中，我们将进一步详细讨论。附录 A 总结了 FoxPro2.0 中全部新的以及改变的内容。

## FoxPro2.0

FoxPro2.0 版与 1.x 版一样有单用户或多用户包。可是，Fox 软件现在在每个软件包中都提供二个版本：标准和扩展。因此，单用户软件包由 FoxPro(标准)及 FoxPro(扩展)程序组成，多用户软件包或 LAN 包由 FoxPro2(标准)及 FoxPro2x(扩展)程序组成。

标准版可用于 640K 字节内存的 AT 类机上。扩展版是用于真 32 位机上，并要有扩展内存及 Intel 80386 芯片。扩展版应该在当你处理拥有超过 25,00000 个记录的数据库，或者当你拥有需要 65,000 个内存变量，每个数组有 65,0000 个记录，以及 125 个工作区时使用。扩展版使用全部有效的扩展内存以及内存许可量，对索引、窗口及 Browse(浏览)对话的数目无限制。在扩展版中，最大字符串长度是 2G 字节。

FoxPro2.0 有 280KRAM 的基本内存，比 1.x 版少。在网络上运行有麻烦的 1.x 应用在 2.0 版下会运行得不错。当激活 FoxPro1.x 时，它总是通过产生一个 400K 字节的文件来开始，当软件正常结束时，该文件自动消除。既然要为每一个并行 FoxPro 用户提供足够的磁盘空间，这就给网络管理员带来了相当的麻烦。

2.0 版不需要 400K 起始件，因此即使你注册到 360K 的软驱也可以起动，并且进入磁盘上的 2.0 软件。

2.0 版使用新的内存管理模式——段加载器，替换了 1.x 版的三个小覆盖区。因为 2.0 版避免了由于代码在覆盖区外交换而引起的性能降级，所以 FoxPro2.0 应用在网络上运行会更快。

## 开放结构

FoxPro2.0 采用了开放结构，这反映在许多新特征上。例如，开发者可通过外部例程 API 来访问 FoxPro。本产品还提供了新的命令及函数，允许开发都操作它的系统窗口及菜单，以便建立与 FoxPro 本身类似的应用。1.x 版的文件是模糊的二进制格式，而多数 2.0 版文件是 .dbf 格式。例如，FoxPro1.x 有当作数据库的帮助文件，允许你容易地产生你自己应用的帮助文件。现在，FoxPro2.0 还具有 .dbf 文件中的报表、菜单及屏幕格式，因此可以象操作其它数据库一样来操作它们。

## 技术 Rushmore

FoxPro2.0 使用叫做 Rushmore 的数据存取技术, 允许按与主机速度相似的速度提取数据库记录集。Rushmore 允许 FoxPro 应用处理有百万个记录的数据库。它取决于搜索数据库的 FOR 子句表达式。为利用 Rushmore 技术, 数据库必须被索引并且 FOR 子句必须包含基本优化的表达式(BOE)。BOE 可采用如下形式:

```
<index expr> <Boolean operator> <constart expr>
```

其中 <index expr> 必须与产生索引的表达式匹配。索引表达式不包含别名。索引可以是规则(.idx)索引或压缩索引。

若数据库包含 LastName, Street, City, Zip 及 Comment 字段并且它在头四个字段上索引, 那么下列表达式就是 BOE:

```
Lastname = "Hereford"
```

```
Zip < > "20878"
```

```
UPPER(City) = "POTOMAC"
```

下列表达式不是 BOE:

```
UPPER(COMMENT) = "NOTES" &&
```

没有在注释上索引

```
208 $ Zip
```

&&不是 Boolean 操作符

```
SUBSTRCA->Street,1,4) = "Main"
```

&&包含别名

BOE 可以用 AND、OR 或 NOT 逻辑运算符与其它 BOE 混合来构成复杂的 FOR 子句表达式。BOE 也可与非 BOE 混合结合, 只要用 AND 运算符即可, 以产生部分优化的表达式。

Rushmore 技术可在新的 SQL SELECT 命令中自动使用。在命令中(比如 LIST、SCAN、BROWSE)使用了 WHITE 时, FoxPro 将使 Rushmore 无效。当打开的数据库中的记录总数超过 250 万个时, 标准 FoxPro 不能使用 Rushmore, 在这些情况下, 你应该使用扩展版。

可在使用 FOR 表达式的命令中加入 NOOPTIMIZE 项或用 SET OPTIMIZE OFF 命令来使 Rushmore 无效。SET OPTIMIZE ON 将重新使 Rushmore 生效。当部分优化命令修改了在 FOR 中的索引关键字时, 应该使 Rushmore 失效。在这种情况下, 当你不使 Rushmore 失效, 就不会得到有关数据库的大部分当前信息。

## 编译器及项目管理

FoxPro2.0 有一个 .exc 编译器, 它使用户可以分配应用, 而不要将它们与 FoxPro 运行时模块包在一起。编译器仅仅是 FoxPro 项目管理工作的一个方面, 它与 C 语言的 Make 工具相类似。一个 FoxPro 项目文件是特殊的文件, 它跟踪应用需要的全部程序及数据文件, 包括程序、屏幕、库、报告、标签、查询、格式文件, 以及文件之间连接和独立性。

FoxPro 在编译应用时, 项目文件保证应用中的每个编译模块都是基于最新源代码。从项目文件产生的应用包括以屏幕及菜单生成的代码。项目文件本身是一个带有一个相关备注文件(.pj1)的 FoxPro 数据库(.pjx)。

通过使用 CREATE / MODIFY PROJECT 命令激活一个对话, 你可以交互式地产生项目。另外, 也可以从使用下列命令的命令行或程序中产生项目:

```
BUILD PROJECT <prjfile> [FROM <mainpgm> [, <pgm> | <lib> [, ...]]
```

FoxPro 通过处理一个以上的程序和库文件, 并且搜索其它相关程序及库来产生项目。

可以用新的 EXTERNAL 命令来包容文件并由项目管理程序产生的项目中的未定义参数。EXTERNAL 命令也可用于把包含在表达式或宏中的文件名通知该项管理程序, 或者把在另一个过程或 UDF 中生成的数组名通知给项目管理程序。

使用 FoxPro 的 API, 你可以写自己的函数以及对语言的扩展, 并将它们装入外部库。然后新 SET LIBRARY TO <lib> 命令使外部库中的函数对 FoxPro 及应用有效。外部函数可用 C 或汇编语言编写并编译成目标文件。

在建立 FoxPro 项目后, 就可以选择建立一个运行时分配的应用(.fxp), 与 FoxPro1.x 中的类似, 或者可以建立一个独立.exe 文件, 它不需要 FoxPro 运行时模块。为了建立.exe 文本, 要求 BUILD EXE <exefile> FROM <prjfile> 命令。补充的 BUILD APP 命令将生成.fxp 格式的应用。

## 外部例程 API

FoxPro 的应用程序界面(API)允许你扩展语言的功能以及用户界面。使用它, 可以执行高级操作, 包括处理 FoxPro 事件; 处理 FoxPro 语句以及操作内存, 备注文件; 数据库和 FoxPro 窗口。

API 库包含如下例程扩展集:

- 生成、释放、改变内存变量的值。
- 抽取 FoxPro 的文件能力来优化文件 I/O。
- 存取 FoxPro 的数据库, 包括返回记录到上一个版本。
- 允许事件处理截取 FoxPro 事件。
- 控制窗口和流输出。
- 产生或释放菜单基和菜单, 或改变系统菜单, 进而产生新的磁盘附件集。
- 截取及执行错误情况。

外部例程接受 FoxPro 的一个参数——指向包含参数数目以及参数本身的数块的指针。程序可以使用引用或值来传递参数。程序可以用各种 API 例程返回结果, 给 FoxPro 包括叫做 \_RetVal() 的例程, 它用于返回带 null 的字符串或返回任何数据类型(备注类型除外)。

对于 FoxPro 用户, 外部例程就象固有的 FoxPro 函数或命令, 只是它不能缩写成四个字符。万一外部例程与 FoxPro 固有函数, 内存函数组, 或一个 UDF 的名字一样怎么办? 外部例程优先于类似名字的 UDF 的定义, 但它不优先于固有的函数或数组。

外部例程仅用了用 8086 汇编语言或 Watcom C 编码并且以严格条件考虑内存保护。它们可以被清理或锁定, 这取决于它们是放在覆盖层上还是放在根段上。

## 类似 dBASE IV 的特征

用户熟悉的 dBASE IV 的特征之一是多重索引工具，它允许在 .mdx 文件中最多可以保持 47 个索引标志。在 dBASE IV 中，.mdx 文件可以是产品或非产品 .mdx。产品 .mdx 有一个数据库一样的名字，它的所有标志都自动保护，以便它们总与数据同步。FoxPro 具有混合索引(.odx)文件，它与 dBASE IV 的 .mdx 文件等价，只是 .mdx 最多有 47 个标志而无索引标志数的限制。“结构”.cdx 与 dBASE IV 产品 .mdx 在功能上等价。.cdx 文件自动使用 FoxPro 2.0 的压缩类型索引。

如下面 INDEX 命令：

```
INDEX ON <idx_expr> TAG <tagname> [OF <.cdx file>]
```

在 .cdx 文件中产生标志入口。索引表达式可以包含用户定义的函数。在 dBASE IV 中，仅仅在 .mdx 中的索引标志可具有 DESCENDING 项。在 FoxPro 2.0 中，不必建立索引标志或新索引文件来获得降序数据。SET ORDER TO <field or index> DESCENDING 命令可以满足你的要求。为了支持新的 .cdx 系统，2.0 版增加了新的函数及命令，如 CDX()、MDX()、TAG()、COPY TAG，以及 DELETE TAG。

另一个 dBASE IV 特征(FoxPro 2.0 新添特征)是 SET SKIP TO 命令。与 SET RELATION 命令连接起来，当在父文件和子文件中的记录之间有一对多或多对多关系时，它就可以连接文件，允许存取与关联词匹配的子文件中的记录。

## SQL SELECT 和 RQBE

FoxPro 2.0 的另一个新特征是 SQL SELECT 命令(不要与 SELECT <workarea> 命令混淆)，可用于从一个或多个数据库中提取数据。SQL SELECT 是一个功能强大的命令，允许你提取数据，而不必考虑工作区中包含的数据库。一个 SQL SELECT 命令代替了一系列的 FoxPro 命令或过程，以便获得排序和显示希望的信息。

与 dBASE IV 不一样，在使用 SQL SELECT 之前你不必要先使 SQL SELECT 为 ON。而且，若你不想在命令窗口或程序中使你自己的整个 Select 语句格式化，就可用 FoxPro 的新 RQBE(关系例子查询)系统来交互地提取数据集。使用 RQBE 可以定义数据。FoxPro 然后形成适当的 Select 查询并执行它。你可以将查询存取到一个 .qpr 文件中，并且可以将查询直接定向到文件、打印机或表中。

## FoxPro 屏幕和菜单生成器

作为开放结构的一部分，FoxPro 2.0 为用户提供了所有用户界面设计及生成工具，可用于生成与 FoxPro 本身的类似的应用屏幕及菜单。工具之一是屏幕生成器，可以用它来设计屏幕对象和用户屏幕。FoxPro 屏幕对象列在表 1-1 中。屏幕对象可以配备一些属性，包括颜色、范围及有效规则。图 1-1 显示一个屏幕生成器的实例。

Screen Builder 可以用 CREATE / MODIFY SCREEN 命令激活。有关屏幕的全部信息(包括关于每个对象，屏幕布局和数据的数据)存放在 FoxPro 的带有相应备注文件的

数据库文件中。这些.scx.sct 文件可以当作.dbf/.fpt 文件一样使用。

表 1-1 FoxPro 屏幕绘图对象

对象	描述
实例	定义数据及命令怎样进入FoxPro。包括检查框、菜单弹出、可翻译表、文本按钮和无线按钮等。
字段对象	包括数据库字段、内存变量，来自UDF和表达式的输出数据。
图象对象	包括线及框
文本对象	包括任何文本。

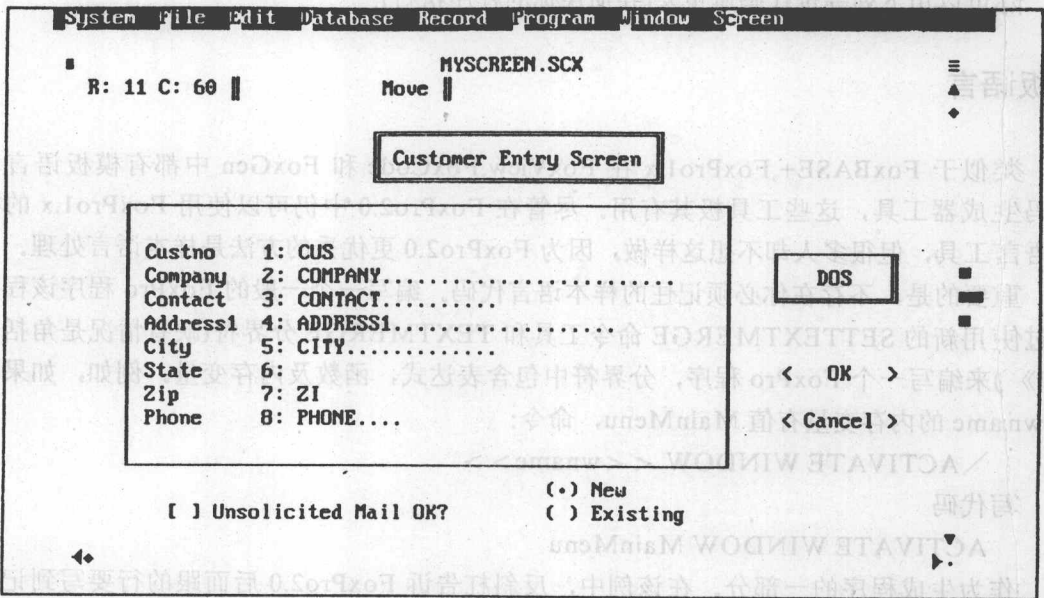


图 1-1 FoxPro2.0 的样本屏幕生成器屏幕

当设计完屏幕后，可以使用存放在.scx 数据来生成重新构造屏幕的代码。由屏幕生成且产生的代码包括你定义的所有对象定义和代码片。

FoxPro2.0 有一个菜单生成器，可用于处理菜单，其行为类似于屏幕生成器。使用屏幕生成器，可以为菜单选项分配命令或生成过程。用菜单生成器生成的数据存放在带有.mnx 扩展名的菜单文件中，它实际上是一个.dbf 数据库，类似于屏幕文件(.scx)。

菜单文件实际上是应用的源代码的一种形式，.mnx 文件存放你生成的菜单定义和代码片。菜单生成且使用存放在菜单文件中的数据来生成适当的应用代码。

## 宏及 EVALUATE()

宏虽然很方便，但会减慢程序的速度。这样，若内存变量 Mfile 在 dBASE IV 和 FoxPro1.x 中，当你指向一个文件时，可以用间接文件引用代替宏函数。包含了数据库文件名，USE(Mfile)命令将比 USE & Mfile 更有效。FoxPro2.0 突破了文件名的概念。使用需要一个名字如数据库字段、窗口和菜单框名的命令时，你就可以用间接名字引用代替宏替换。当 Wname 间接名字引用在少数几个命令中不支持，如？，LIST,DISPLAY(带有一个字段名)，以及 CALL。是一个内存变量时，类似于 DEFINE WINDOW(Wname)的命令将完全有效。

FoxPro2.0 很少使用宏，这是由于新的多功能的 EVALUATE()函数的存在。象 TYPE()，该函数给字符表达式做评价，但是它并不返回结果的数据类型，而是返回表达式的结果。因此，用下列代码编码表达式的应用中：

```
LIST FOR & criteria
```

你可以用下列等价代码避免宏(并加快你的程序执行)：

### 模板语言

类似于 FoxBASE+,FoxPro1.x 在 FoxView,FoxCode 和 FoxGen 中都有模板语言和代码生成器工具，这些工具极其有用。尽管在 FoxPro2.0 中仍可以使用 FoxPro1.x 的模板语言工具，但很多人却不想这样做。因为 FoxPro2.0 更优秀的方法是样本语言处理。

重要的是，不存在你必须记住的样本语言代码。编写一个一般的 FoxPro 程序该程序通过使用新的 SETTEXTMERGE 命令工具和 TEXTMERGE 分界符(缺省情况是角括号《和》)来编写一个 FoxPro 程序，分界符中包含表达式，函数及内存变量。例如，如果你称做 wname 的内存变量有值 MainMenu，命令：

```
\ACTIVATE WINDOW <<wname>>
```

写代码

```
ACTIVATE WINDOW MainMenu
```

作为生成程序的一部分。在该例中，反斜杠告诉 FoxPro2.0 后面跟的行要写到记录生成的应用的文件中。

你只需记住少数几个命令来产生生成的代码。例如，如果你想用一个数据入口屏幕生成一个完整的应用，就可以简单地用屏幕生成器设置屏幕，并且用使用 SETTEXTMERGET 和相关命令的程序读存放在 .scx 文件中的数据。第五章将详细论述模板语言。

### 新的及增强的 READ / GET 命令及函数

为支持如检查框等设计对象的产生及显示，FoxPro2.0 具有新的、增强的与 READ 及 @...GET 有关的命令。例如，CLEAR READ 是新命令，可终止并退出活动的 READ，在 FoxPro 中，READ 最多可以嵌套 4 级。当前 READ 级可以用新的



RDLEVEL()函数终止。

大家熟悉的 READ 命令现在有几个选择项, 包括 CYCLE 及 OBJECT。使用 CYCLE 选项, READ 不终止最近的 GET。相反, 它循环返回到第一个 GET。循环可以用 Ctrl-N 或 ESC 终止。OBJECT 项指定 READ 期定最初选择了哪个 GET 字段或对象。

@...GET 命令用于产生 GET 对象, 包括检查框、文本、按钮、无线按钮、不可见按钮、弹出式菜单和滚翻表。这些项在表 1-2 中做了简明定义。@...GET 命令的 PICTURE 或 FUNCTION 项指示要生成哪一种对象。例如, 子句 PICTURE"@ \* C"或 FUNCTION"\* C"生成一个检查框。表 1.3 列出了表出了不同对象的 FUNCTION 格式。

表 1-2 GET 对象及意义

GET 对象	说明
检查框	屏幕或窗口上的一个框或小区域可以包含或不包含检查标志(X)指示内存变量或数据库字段的状态。
不可见按钮	是窗口或屏幕上的一个可选的矩区域, 在它的底下可放置显示文本或字符。
无线按钮	屏幕或窗口上的一个框或小区域, 指示当前选择的按钮。
弹出式菜单	包含从中做选择的选择项表的矩形区域。
翻滚表	包含来自内存变量及弹出定义的一组选择项的表。
文本按钮	包含由两对角括号包围的文本的窗口及屏幕区域。

表 1-3 GET 对象的 FUNCTION 格式

GET 对象	FUNCTION 格式
检查框	* C
不可见按钮	* I
无线按钮	* R
弹出式菜单	^
翻滚表	无
文本按钮	*

如果在前面加上@符号, 就可以在 PICTURE 子句中使用同样的格式。

@...GET 命令具有附加的 SIZE 和 ENABLE / DISABLE 子句。例如, 在检查框中, SIZE 指示矩形区域的宽度。ENABLE 子句激活选择的对象, DISABLE 的对象用暗