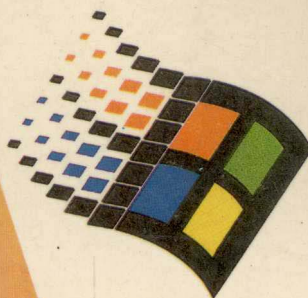


计算机数据库应用技术系列丛书



欧海德 著
亦 鸥 审校

For
Windows™

FOXPRO 2.5

开发指南

学苑出版社

计算机数据库应用技术系列丛书

FoxPro 2.5 for Windows

开 发 指 南

欧海德 著
亦 鸥 审校

学苑出版社

1993

(京)新登字 151 号

内 容 简 介

本书是使用 Microsoft FoxPro 2.5 for Windows 进行程序设计的实用性很强的读物。它详细地介绍了 Microsoft FoxPro 2.5 for Windows 的强大功能和用法。

本书对从事软件设计、开发和应用的技术人员具有重要的参考价值。

需要本书的用户,可与北京 8721 信箱联系,邮码 100080,电话 2562329。

计算机数据库应用技术系列丛书

FoxPro 2.5 for Windows 开发指南

著 者: 欧海德

审 校: 亦 鸥

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

印 刷: 北京东升印刷厂印刷

开 本: 787×1092 1/16

印 张: 12.5 字数: 282 千字

印 数: 1—5000 册

版 次: 1993 年 12 月北京第 1 版第 1 次

I S B N: 7-5077-0874-8/TP·23

本册定价: 19.00 元

学苑版图书印、装错误可随时退换

目 录

第一章 屏幕与菜单	1
1.1 屏幕	2
1.1.1 屏幕建立器的优点	2
1.1.2 本节中使用的术语	3
1.1.3 实用屏幕	3
1.1.4 其他屏幕	4
1.1.5 屏幕集	4
1.1.6 代码段	5
1.1.7 调用屏幕程序	6
1.1.8 READ 的重要性	6
1.1.9 工作环境	7
1.1.10 有关设计的考虑	7
1.1.11 生成的程序	9
1.2 屏幕布局	9
1.2.1 设置代码	9
1.2.2 Cleanup 和过程代码	14
1.2.3 窗口定义	17
1.2.4 READ 级子句	17
1.3 字段对象和控制	22
1.3.1 字段对象	22
1.3.2 按钮	25
1.3.3 单选钮	28
1.3.4 核选框	29
1.3.5 弹出式菜单	30
1.3.6 列表	32
1.4 协调屏幕的浏览窗口	34
1.4.1 激活浏览窗口	34
1.4.2 改变窗口大小和窗口定位	35
1.4.3 在模化 READ 过程中激活菜单	35
1.5 调试应用程序中的屏幕代码	36
1.6 使用屏幕程序的 FoxDoc	36
1.7 菜单	37
1.7.1 菜单建立器的优点	37

1.7.2	常用术语	38
1.7.3	建立代码段	39
1.7.4	调用菜单程序	39
1.7.5	激活菜单	40
1.7.6	使用环境	41
1.7.7	设计时考虑的问题	42
1.7.8	有关生成的程序	43
1.7.9	一般选项	43
1.7.10	生成设置代码例子	44
1.7.11	Cleanup 代码例子	46
1.7.12	菜单条选项	47
1.7.13	菜单选项	47
1.7.14	菜单过程例子	48
1.7.15	选项核选框	49
1.7.16	调试用户菜单	51
1.7.17	隐藏命令窗口	52
1.8	协调屏幕和菜单	52
1.8.1	管理菜单系统	52
1.8.2	通过菜单访问屏幕控制	54
第二章	项目组织工具	55
2.1	项目的优点	55
2.2	项目包括的内容	56
2.3	一个项目与多个项目	56
2.4	可移植式应用程序的根路径	56
2.5	选择主文件	57
2.6	在应用程序中包括可修改的文件	58
2.7	项目中的未知引用	58
2.8	项目中的过程码	59
2.8.1	错误处理	60
2.8.2	保存当前环境	60
2.8.3	建立新环境	61
2.8.4	保存和恢复系统菜单条	62
2.8.5	检测资源	62
2.8.6	实用过程	62
第三章	数组	63
3.1	建立数组	63
3.2	FoxPro 数组函数	64
3.3	数组操作	64
3.3.1	初始化整个数组	64

3.3.2	引用数组元素	65
3.3.3	数组元素赋值	65
3.3.4	重新定义数组维数	66
3.4	公用和私用数组	66
3.4.1	公用数组	66
3.4.2	私用数组	66
3.4.3	数组的局限	67
3.5	把整个数组传给用户定义的函数	67
3.6	在数组和表间传递数据	67
3.7	数组和 SQL SELECT	68
3.8	数组和 FoxPro 控制	69
第四章	使用 SQL SELECT	70
4.1	问题中使用的表	70
4.2	问题	71
4.3	答案	72
第五章	调试应用程序	81
5.1	程序错误	81
5.2	保存编译错误	81
5.2.1	使用程序菜单指定错误文件	82
5.2.2	使用 SET LOGERRORS 命令	82
5.2.3	解释编译错误文件	82
5.2.4	了解产生编译错误的原因	82
5.3	运行错误	82
5.4	调试提示	83
5.5	报表变量所能做与不能做的	84
第六章	低级文件输入/输出	86
6.1	建立文件	86
6.2	打开文件	87
6.2.1	读文件	88
6.2.2	写文件	88
6.2.3	关闭文件	89
6.3	其他另外的低级 I/O 命令和函数	89
6.4	文本合并	89
6.4.1	合并文本和文本合并元素	90
6.4.2	输出到窗口和文件	94
第七章	定制帮助与优化应用程序	96
7.1	选择帮助类型: Wondows 或. DBF	96
7.1.1	检查帮助类型	96
7.1.2	使用 Windows 式帮助	97

7.1.3	使用.DBF 类型帮助	97
7.2	建立.DBF 类型帮助	97
7.2.1	了解 FOXHELP.DBF	97
7.2.2	帮助表的要求	98
7.2.3	FOXHELP 论题	98
7.2.4	FOXHELP 内容	98
7.2.5	FOXHELP 交叉参考	99
7.2.6	查找帮助显示	99
7.2.7	指定帮助表	99
7.2.8	集中显示帮助话题	99
7.2.9	更好的安排	102
7.3	优化应用程序	102
7.3.1	Rushmore 技术	102
7.3.2	一般运行提示	107
第八章	动态数据交换	110
8.1	什么是动态数据交换	110
8.2	客户和服务端	110
8.3	应用程序、论题和项	111
8.3.1	服务名	111
8.3.2	论题	111
8.3.3	项	111
8.4	建立与另一个应用程序的对话	111
8.4.1	初始化和终止会话	112
8.4.2	从其他应用程序请求数据	112
8.4.3	在其他应用程序中执行命令	112
8.4.4	向其他应用程序发送数据	113
8.4.5	处理 DDE 错误	113
8.4.6	建立报告或自动链	114
8.4.7	使用同步命令	114
8.5	将应用程序变为 DDE 服务器	115
8.5.1	建立服务	115
8.5.2	建立论题	116
8.5.3	提供调用返回函数	116
8.5.4	响应该 DDE 动作	116
第九章	对象链接和嵌入	120
9.1	本章中的术语	120
9.2	嵌入与链接	121
9.3	在 Foxpro 界面使用 OLE	122
9.3.1	插入 OLE 对象	122

9.3.2	剪贴和链接 OLE 对象	123
9.3.3	编辑 OLE 对象	124
9.3.4	把 OLE 对象转换成静态图画	125
9.3.5	高级 OLE 概念	125
9.4	与 OLE 对象一起使用 FoxPro 命令	126
9.4.1	MODIFY GENERAL 命令	126
9.4.2	@...SAY 命令	127
9.4.3	APPEND GENERAL 命令	129
第十章	用 FoxDoc 建立应用程序文档	131
10.1	入门	131
10.1.1	启动 FoxDoc	131
10.1.2	获得帮助	131
10.1.3	认识 FoxDoc 文件	131
10.1.4	FoxDoc 的输出	132
10.1.5	FoxDoc 与 Beautify 的比较	132
10.2	试用 FoxDoc	133
10.3	生成文档	133
10.3.1	描述所要建档的内容	133
10.3.2	决定建档选项	135
10.3.3	启动建档过程	135
10.4	选择建档选项	135
10.4.1	格式化源代码选项	136
10.4.2	行为图选项	138
10.4.3	报告表选项	139
10.4.4	打印选项	142
10.4.5	文件扩展名选项	143
10.5	在程序中使用 FoxDoc 命令	143
10.5.1	宏	143
10.5.2	DOCCODE: 程序伪语句	144
10.5.3	PRGPATH 和 DATAPATH	144
10.5.4	其他 FoxDoc 指令	145
10.6	FoxDoc 是如何工作的	145
10.7	解释交叉参考报告	148
10.8	熟悉关键字文件	149
第十一章	用 FoxApp 建立应用程序	151
11.1	使用 FoxApp	151
11.1.1	指定表	151
11.1.2	指定屏幕	152
11.1.3	生成应用程序	152

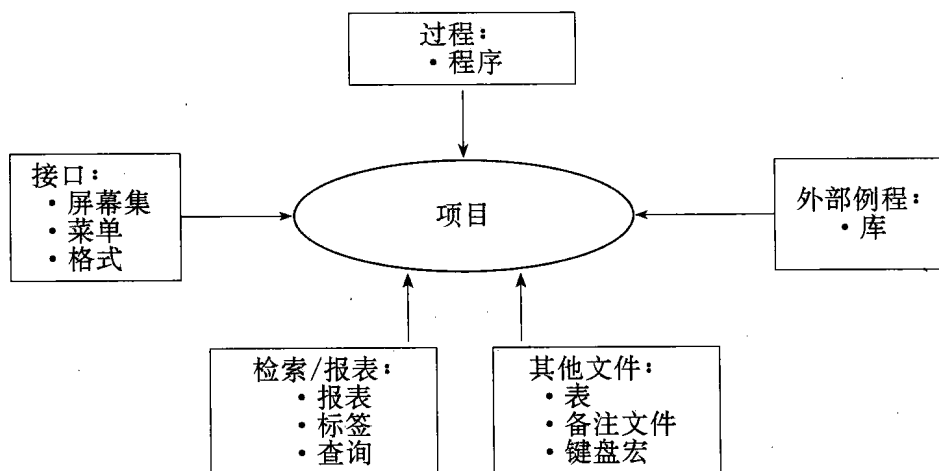
11.2 生成菜单系统.....	153
11.2.1 File 菜单	153
11.2.2 Edit 菜单	153
11.2.3 Application 菜单	153
11.2.4 Utilities 菜单	154
11.2.5 Help 菜单	154
11.2.6 Browse 菜单	154
第十二章 兼容性	155
12.1 FoxBASE+兼容性	155
12.1.1 模拟 FoxBASE+指定击键	155
12.1.2 FoxBASE+模拟的额外 SET 选项	156
12.1.3 无法避免的差异	157
12.2 SET COMPATIBLE 命令	158
12.3 从 FoxBASE+ 2.10 中转换文件	159
12.3.1 .NDX 索引文件	159
12.3.2 .DBT 备注文件.....	160
12.3.3 FOX 程序文件	160
12.3.4 编译程序	160
12.3.5 执行程序	160
12.4 转换 FoxPro 1.XX 文件	162
第十三章 多用户环境下的 FoxPro	164
13.1 系统构造.....	164
13.1.1 临时工作文件	164
13.1.2 CONFIG.FPW	164
13.1.3 特殊 CONFIG 选项	165
13.1.4 FOXUSER 资源文件	166
13.2 多用户环境下的编程.....	166
13.2.1 表的独占使用	166
13.2.2 表的共享使用	167
13.2.3 写访问权与只读访问权	167
13.2.4 记录和文件锁定	168
13.2.5 自动锁定与手动锁定	168
13.2.6 对记录和表解锁	168
13.2.7 执行自动锁定的命令	168
13.2.8 SET REPROCESS	169
13.2.9 手动锁定函数	170
13.2.10 管理冲突	171
13.2.11 错误处理例程.....	171
13.2.12 低级文件函数.....	174

13.3	优化性能	174
13.3.1	将临时文件置于本地驱动器上	174
13.3.2	排序文件与索引文件	175
13.3.3	独占使用文件	175
13.3.4	锁的长度	175
13.4	多用户命令和函数	175
第十四章	跨平台应用	177
14.1	有关术语	177
14.2	在 Windows 下运行 MS-DOS 应用程序	178
14.2.1	方法 1:像运行 MS-DOS 应用程序一样运行	179
14.2.2	方法 2:输送 MS-DOS 应用程序	182
14.2.3	方法 3:充分利用 Windows 特征	184
14.3	维护跨平台文件	184
14.3.1	维护平台屏幕	185
14.3.2	维护跨平台报告	187
14.3.3	维护跨平台标签	187
14.3.4	维护跨平台菜单	187
14.4	在 MS-DOS 下运行 Windows 应用程序	188
14.5	输送机的工作过程	189
14.5.1	Transport 对话框	189
14.5.2	特殊的输送机决定	190
14.6	选择开发平台	190
14.6.1	MS-DOS 字符模式平台	190
14.6.2	Windows 图形平台	190

第一章 屏幕与菜单

本章说明了如何使用 FoxPro 的高效工具来开发应用程序。FoxPro 的高效工具自动建立用户界面结构,取出和显示信息,由不同用途的应用程序组成. APP 或. EXE 文件,并且在这些组成发生变化的时修改应用程序。

所谓项目就是把应用程序的各部分集成在一起的统一机制。如下图所示。



本章的每一节都有例子、解释和推荐的技术,以此帮助用户更深入地学习 FoxPro。为了深入学习本节内容,用户应该知道如何使用高效工具和对 FoxPro 语言有一个基本的了解。如果需要学习该高效工具,请读者参考 Microsoft FoxPro 软件包中的《FoxPro 用户指南》或《FoxPro 入门教程》一书。如果对一些命令或函数的细节感兴趣的话,请参考 Microsoft FoxPro 软件包中的《FoxPro 语言参考手册》。

本章的各节都包含 FoxPro 2.5 版本提供的 ORGANIZER 示例应用程序中的例子。这个应用程序在 FoxPro 下的 \SAMPLE\ORGANIZE 子目录中。

使用 ORGANIZER 时,执行 ORGANIZE. APP。ORGANIZER 包括如下模块:

- | | |
|---------------|-----------------------|
| • Accounts | 组织有关银行会计的信息 |
| • Clients | 组织委托人信息 |
| • Convert | 帮助用户由一个度量单位转换到另一个度量单位 |
| • Credit Card | 组织信用卡信息 |
| • Friends | 组织有关家庭成员和朋友的信息 |
| • Restaurant | 组织餐馆信息 |
| • Transaction | 组织商业事务信息 |

注意:例子中的 ORGANIZER 代码可能与软盘上的 ORGANIZER 代码有一些不同。

ORGANIZER 应用程序包括八个项目: ACCNTS. PJX, TRANS. PJX, CLIENTS. PJX, FAMILY. PJX, CREDIT. PJX, CONVERT. PJX, RESTAURS. PJX 和 ORGANIZE. PJX。

1.1 屏 幕

在用户建立屏幕时,建立了一段源代码。有关屏幕的信息存在.SCX 表中,这个表包含一个具有.SCT 扩展名的相关的备注型字段。屏幕文件包括:

- 定义窗口的信息(如果屏幕被定义为窗口)
- 定义所有字段和控制的大小、位置和外观的信息。
- 有关环境的信息(如果环境被保存在屏幕中)
- 定义屏幕的特性和屏幕中对象的基本代码(用代码段定义)

GERSCRN 是 FoxPro 屏幕生成器,它从.SCX 表中得到信息,然后生成一个扩展名为.SPR 的屏幕程序文件。

注意:不能编辑.SPR 屏幕文件。如果要修改屏幕,就要修改屏幕本身(用 FoxPro Screen Builder 做)另起一段。本节中的例子说明了如何使用有关 READ 和对象级子句定义屏幕的特性和屏幕中的对象。所有例子都来源于 Foxpro 提供的 ORGANIZER 应用程序。

为了更好地了解本节内容,用户应该知道如何建立屏幕。有关建立屏幕的更多信息,请参考 Microsoft FoxPro 软件包中的《FoxPro 用户指南》中的“用屏幕建立器设计屏幕”一章。打开例子中的屏幕(.SCX)文件,看看屏幕上的对象的代码段。比较代码段来看对象是如何与其他对象相互作用的,而且如何与屏幕集中的其他屏幕相互作用,另外再运行 ORGANIZER 应用程序,观察对象和屏幕在应用程序中是如何操作的。

1.1.1 屏幕建立器的优点

1. 节约时间

用 Screen Builder 节约用户建立应用程序的时间,因为代码生成器建立了定义字段的物理安排和控制所需的全部代码。另外生成器还给过程起名,减少应用程序中重名的可能性。

2. 改进组织和清晰度

Screen Builder 提供给用户压缩界面代码和把它与过程代码分开的方法。

用户能够把对象和定义它的动作的过程统一起来。过程是用代码段定义的,这些代码段是和对象存储在一起。

3. 精确显示屏幕的样子

在用 Screen Builder 设计屏幕时,用户能够看到屏幕的样子,以及在生成的屏幕中不同对象是如何相互作用的。用户可以用不同的布局来试验。

4. 提高效率

用户能够使用“实用”屏幕,它可以和屏幕集中的其他屏幕程序组合使用。一个实用屏幕

可以在应用程序中一遍又一遍地使用,而不需要重代码。如果修改了实用屏幕,那么这个变化就反映在使用这个屏幕的每一个应用程序中。

包含在 FoxPro 软件中的一些实用屏幕可以做为应用程序建立块来使用,用户能够使用这些屏幕或建立用户自己的实用屏幕来反映自己的界面风格。

1.1.2 本节中使用的术语

下面列出了本节使用的术语:

代码段——与对象或屏幕有关的一段代码,它与屏幕一起存储。

控制——按钮、单选钮、核选框、菜单、列表和不可见按钮都是屏幕中定义的控制。

生成代码——由 FoxPro GENSCRN 建立的代码。

生成器命名的过程——由屏幕生成器起名的过程,这些过程由生成器起名,防止了在应用程序中重名。

对象——屏幕上的文本、字段、框、线或控制。

对象级子句——屏幕上某一对象的子句。每一子句由代码段定义。对所有对象来说,WHEN,VALID 和 MESSAGE 子句都可用。ERROR 子句只适用于 GET 和 EDIT 字段。

读级子句——某一屏幕的子句。每一子句可用代码段定义。ACTIVATE,VALID,DEACTIVATE,SHOW 和 WHEN 子句对所有对象都是适用的。ERROR 子句适用于 GET 和 EDIT 字段。

屏幕集——屏幕集可以是一个屏幕或几个屏幕的结合。SPR 程序为屏幕集而生成。

.SCX 文件——屏幕表文件。

.SCT 文件——与 .SCX 表有关的备注文件。

.SPR 文件——生成的屏幕程序文件。

.SPX 文件——编译的 .SPR 文件

用户命名的过程——由用户命名的过程,而名字又是由代码段或表达式命名的。与用户命名的过程相对的是生成器命名的过程。

实用屏幕——在屏幕集中与其他屏幕结合的屏幕。实用屏幕在应用程序中提供了一致性。对实用屏幕所做的修改反映在使用该屏幕的应用程序的所有模块中。

1.1.3 实用屏幕

实用屏幕是一个为了能在应用程序中被使用若干次,同时也可以被若干个应用程序使用而设计的。它们常常在屏幕集中与其他屏幕结合,通常被设计成独立于结构和某表内容。

可以用许多方式使用实用屏幕。ORGANIZER 应用程序使用屏幕来遍历数据文件(CONTROL1.SCX)和定位数据文件中的某条记录(BROWSER.SCX)。

当用户使用实用屏幕时:

- 可以在不同的应用程序中共享同一屏幕。
- 在一个应用程序中提供一致性。如果该屏幕在几个地方用,那么用户会熟悉它的样子和功能。
- 对实用屏幕的修改反映在应用程序中。在修改实用屏幕时,Project Manager 保证应用程序中使用的是实用屏幕的最后版本。用户要做的事情就是重建包含修改过的实用

屏幕的项目。

1. 命名实用屏幕的变量

在定义实用屏幕中的变量时,使用前缀“m.”避免了变量名与表中文件字段名冲突的可能性。

2. 实用屏幕中的局部(REGIONAL)变量

因为用户把实用屏幕与屏幕集中的其他屏幕结合起来,所以在设计实用屏幕时一定要用 REGIONAL 命令把变量定义为 REGIONAL。用这个命令,用户可以建立局部的内存变量和内存变量数组。这些内存变量和数组都有同一的名字,是因为与这些名字有关的值保护在“局部(region)”中。

REGIONAL 变量的使用参见本节后面的内容以及 Microsoft FoxPro 软件包中的《FoxPro 语言参考手册》一书。

1.1.4 其他屏幕

许多屏幕是为某表或某一应用程序而专门设计的。这样的屏幕叫做“非实用”屏幕:

- 只能使用一次
- 参考信息与当前应用程序有关。

1.1.5 屏幕集

屏幕集由一个或若干个屏幕组成。生成代码后,为全部屏幕集建立一个.SPR 程序。

1. 界面部分的模式

通常,在屏幕集中使用的屏幕设定义为窗口,这也许有助于一个屏幕一个屏幕的去考虑。例如,控制板就是单独的屏幕,因为它占据自己的窗口。不要把屏幕想成“显示器上的所有东西”,而是占据窗口的实体。

2. 建立屏幕集

用户能够用 Project Manager 建立屏幕集。在向项目增加屏幕时,Edit Screen Set 对话框出现。选择 Add 按钮即可将所需的屏幕增加到屏幕集中。在对话框底部用户可给屏幕集命名。屏幕集中第一个屏幕的名字缺省显示在文本框中。

屏幕集中屏幕放置的顺序影响了执行中访问屏幕的顺序,同时也影响了.SPR 程序中 READ 级代码段生成的须序。

Edit Screen Set 对话中的可选项可帮助用户编辑、删除和安排屏幕。用户也能禁止某代码段的生成。有关在屏幕集中安排屏幕和 Edit Screen Set 对话中的其他选项的信息,请参考 Microsoft FoxPro 软件包中的《FoxPro 用户指南》中的“用屏幕建立器设计屏幕”一章。

注意:在安排屏幕时,通过把屏幕放到项目中和保存项目,来保存指定的坐标。在 Edit Screen Set 对话中指定的所有信息保存在项目文件中。

注意,在定义和生成包含多屏幕的屏幕集时,后继屏幕的设置码与第一个屏幕的设置码

连在一起。写一个屏幕的代码可能会无意地改变了为上一屏所写的代码输出。千万要留心设计和编码,保证能够得到想要的结果。

1.1.6 代码段

用户在生成屏幕时,可以为屏幕中某一个对象的子句定义有关的代码段。当然也可以定义影响全屏的代码段。有关定义代码段的操作,在 Microsoft FoxPro 软件包中的《FoxPro 用户指南》的“用屏幕建立器设计屏幕”一章中说明。

在分配对象或屏幕子句时,可以定义子句的表达式和过程。

本节中的例子来自 ADDUSERS.SCX 中的 USERLAST 字段。如果定义了子句的表达式,那么表达式插入在与子句有关的程序中。

```
@8,41 GET m.userlast;  
...  
ERROR "Blank entries are not allowed";  
DISABLE
```

如果表达式调用其他过程,那么用户可以用 Cleanup & Procedure 代码定义它,该过程也可是位于用户所设路径中的过程。

Cleanup & Procedure 代码在本节后面介绍。

如果是为一个子句定义的过程,那么这个过程以 SYS(2015)函数命名。

过程名插入在程序中。

```
@8,41 GET m.userlast;  
...  
valid _px20n0uaf()
```

该过程在程序文件尾部生成。与子句有关的名字成为 FUNCTION 命令及下面代码段的一部分。

```
.....  
FUNCTION _px20n0uaf                                &&. m.userlast VALID  
#REGION 1  
IF EMPTY(m.userlast)  
    RETURN .F.  
ENDIF  
SHOW GET m.userfirst ENABLE  
_CUROBJ=OBJNUM(m.userfirst)
```

...在代码段中不必要有 RETURN 命令。如果没有 RETURN,那么函数自动返回。
T.。

说明:分配给代码段的名字在每次生成屏幕时都要改变。修改屏幕时,必须修改屏幕本身(.SCX 文件),然后重新生成代码。如果修改生成的程序(.SPR 文件),然后重新生成屏幕,那么所有的修改就丢失了。

在生成的代码中,所有的过程都有唯一的名字和函数名来注释,并描述与过程有关的屏幕、对象和子句。注释也插入在生成的程序中,以便 FoxPro 为用户程序生成文档。

1.1.7 调用屏幕程序

调用屏幕程序时,必须用下面的语法:

DO <filename>.SPR

屏幕程序(.SPR)、菜单程序(.MPR)、程序(.PRG)、查询(.QPR)、项目(.PJX)和应用程序(APP)以不同扩展名区别。这样这些文件可以有相同的基本名,而在软盘上却不会互相覆盖。

编译过的屏幕程序的扩展名是.SPX的文件。调用屏幕程序时一定要用扩展名是.SPR或.SPX的文件。

1.1.8 READ 的重要性

READ 是用来激活和协调屏幕集、菜单和其他窗口,使之成为相互影响的会话层。屏幕程序中的其他语句定义了屏幕中对象的样子和特性。READ 命令激活了对象,下面说明了 READ 级子句和执行相应的动作:

Activate	用来禁止其他窗口中的对象,隐藏窗口,显示消息等。
Deactivate	用来保持当前 READ 窗口的活动(不允许其他窗口成为输出窗口)。也可以用来中断(或不中断)有 RETURN 值的 READ。
Show	用来刷新 SAY 和 GET 字段,允许和禁止 GET 对象。
Valid	用来决定根据逻辑表达式的结果 READ 是否退出。
When	用来决定根据逻辑表达式的结果 READ 是否执行。WHEN 子句也可以用来设置 READ 的环境。例如,如果在执行读操作时,要用菜单,那么就在 READ WHEN 子句中执行菜单程序。
Modal	如果窗口被定义为 Modal 形式模化的,那么它就承担了 FoxPro 对话的操作。就是说屏幕集外的窗口不能提到屏幕集窗口上,并且当前菜单中系统也临时无效了。
With	如果 Associated Window 列表在 Generate 对话中,那么用户就可以包括一个与屏幕集一起激活的窗口列表。这个窗口列表加在 READ 命令上。WITH 子句的语法如下:

READ WITH <window title>

ACTIVATE 子句能够定义 Associated Window 列表中的屏幕的操作。而且只有活动窗口是 READ 窗口时才执行。在屏幕集中的 Browse 窗口,桌面附件或其他非 READ 窗口中不能执行。

WITH 子句自动使屏幕集是 Modal 开式,只有那些在屏幕集中定义,又在 Associated Window 列表中指定的窗口,才能被激活。

下面过程定义了 Modal 屏幕,给屏幕集分配了 Associated Window 列表:

(1)如果屏幕是项目的一部分,打开项目,从显示的列表中选择屏幕。

如果屏幕不是项目的一部分,打开屏幕的.SCX 文件,然后选择 Program 菜单中的 Generate。

(2)选择 More 按钮,标记 Modal Screens 核选框,或选择 Associated Windows 按钮,指定与

屏幕一起激活的有关窗口。

Associated Window 列表的定义自动使屏幕集为 Modal 形式的。

这是有限制的列表。只有那些在列表中指定的窗口才能被激活。如执行屏幕时出现了其他窗口,那么它们只是显示在显示器上,不能被激活和访问。

(3)在 Associated Window 列表中指定屏幕中要包括的窗口。这些窗口能够与 Screen Set 中的窗口一起被激活。在 Associated Window 中没有必要包含 Screen Set 定义的窗口,用户可能要访问的 Browse 或备注窗口,及台式附件都应包括在 Associated Window 列表中。

Rule1——在 READ 中访问备注窗口,在 Associated Window 列表中包括表的别名。

Rule2——在 READ 中访问 Browse 窗口,在 Associated Window 表中包括 Browse 窗口标题(缺省值是表的别名)。

有关 Associated Window 列表的例子,READ WITH 子句和更多信息,请参考本节后面的“Coordinating Browse with Screens”一节。

1.1.9 工作环境

1. 窗口的裁减、拷贝和剪贴

即使窗口的代码不同,用户也能够裁减、拷贝和剪贴一个正在编辑的窗口到另一个中,在几个 Screen Design 窗口间也可以裁减拷贝和剪贴对象。在从一个屏幕向另一个屏幕拷贝和剪贴对象时,与该对象有关的所有信息(包括代码段)也同时拷贝。

2. 操作代码片段编辑窗口

对代码编辑窗口就像对其他文本编辑窗口一样,可以改变大小,缩小和移动,保存屏幕时,窗口的状态也一起保存,打开屏幕时,所有的代码段编辑窗口也出现,就像它们在关闭屏幕时的样子。

从 Screen 菜单选择 Open All Snippets 或 Close All Snippets 可以立即打开或关闭所有代码段编辑窗口。代码段窗口打开时,与它有关的子句或选项在相应的对话中呈暗色显示。

这就向用户提供了窗口状态的直观线索。所有打开的代码段编辑窗口列在 Window 菜单的底部。从菜单中选择一个,就可以激活它。

1.1.10 有关设计的考虑

1. 窗口类型

用户的应用程序应该有一个统一的外观。窗口的样子就向用户提供了该窗口特性的直观线索。例如,所有的输入窗口是一种类型,所有的对话是另外一种类型等等。

有关窗口类型的更多信息,请参考本节后面的“定义窗口”一节。

2. 屏幕控制

屏幕中的控制可减少菜单可选项。如果可以在屏幕控制中使用菜单选项的话,那么就可使用屏幕控制(不要使屏幕很杂乱)。