

第一章 FoxPro产品简介

§ 1.1 FoxPro产品的演变过程

FoxPro最早是由Fox Software公司在FoxBASE数据库软件之后,于80年代末推出的一个功能较强的数据库管理系统软件。它应用在微机环境下,继承了xBASE微机数据库的主要特征,成为继xBASE之后最受用户欢迎的又一个大众型数据库。

Microsoft公司于3年前兼并了Fox Software公司,Microsoft凭借其雄厚的软件技术实力,继续保持对FoxPro产品的深入开发和广泛的市场宣传,从而使FoxPro数据库产品在步入市场的短短数年时间内,历经了数次版本升级,从原先的FoxPro V1.0和V2.0 for DOS版,发展到今天的FoxPro V2.5 for DOS、WINDOWS、UNIX和MACINTOSH版,产品臻于完美,用户迅速增长。

目前,FoxPro V2.5的汉化版已正式推出,伴随由DOS操作系统向WINDOWS操作系统升级的发展潮流,FoxPro for Windows 2.5也以成为国内广大xBASE用户应用升级的最理想选择。

FoxPro for Windows 2.5产品包括:

- FoxPro基本版(FoxPro for Windows V2.5)
- FoxPro扩展工具(FoxPro Distribution Kit V2.5)
- FoxPro移植工具(FoxPro Migration Kit V2.0)
- FoxPro连接工具(FoxPro Connectivity Kit V2.5)
- FoxPro函数构造工具(FoxPro Library Construction Kit V2.5)

§ 1.2 FoxPro产品的功能特征

§ 1.2.1 命令和函数丰富

FoxPro提供了约400多条的命令和函数,数倍于xBASE,这给用户提供了一个相当丰富的可编命令语言环境。这些命令和函数一方面继承了xBASE的命令语言的特征,使广大微机用户自然地进入到新的FoxPro的系统环境中,而另一方面它又对许多命令进行了改进和扩充,使得更具有新的特色,在使用FoxPro编写程序或操作时,使用户感到更加方便,确实好用。

§ 1.2.2 多窗口用户界面

FoxPro完全采用了最新的多窗口技术,为用户提供了一个较好的应用界面。应用多窗口技术给计算机用户界面增加了新的特色,多个窗口在屏幕上显示很具有立体感,且窗口的大小和位置都可任意调整和移动,提高了屏幕显示的有效使用面积,操作直观、简单,且又快又方便,因此深受用户欢迎。FoxPro采用这一技术,不仅使用户在编程或运行操作时与系统进行会话,有更多交换信息的窗口,提高了会话和编程的工作效率,而且给用户提供了一个舒适美观的工作环境。

§ 1.2.3 菜单功能完备

FoxPro仍然提供了在原先一些微机数据库所使用的菜单驱动的操作方式,并进一步把它们的功能组织增强。FoxPro能够象使用xBASE支持的菜单操作方式一样,方便地使用弹出多种菜单操作方式,且能简单、灵活地读取数据库中的数据,同样给用户的编程和应用带来了许多方便。

§ 1.2.4 系统运行速度快

FoxPro在微机环境下以运行速度较快著称,因此受到用户的关注,这是FoxPro系统在性能方面的一个重要特征。FoxPro能充分利用微机系统的硬件资源,优化系统性能,提高了系统的运行速度。FoxPro为提高应用程序执行的速度大量地使用内存。它支持使用扩展的内存规范(EMS),可以使用超过640K的扩展内存。

另外,除了内存资源的利用之外,在用户的计算机装有协处理器(如30387)的情况下, FoxPro的运行速度会更快。FoxPro还可以利用外存的超高速磁盘缓冲区(disk cache)提高速度,以实现运行的高效率,总之, FoxPro充分利用了微机环境丰富而低廉的硬件资源,从各方面优化系统性能,在产品的运行速度方面,具有绝对的竞争优势。

§ 1.2.5 网络功能强大

FoxPro和xBASE一样,除了支持单用户使用之外,还可以在网络环境中运行,作为网上的文件服务器,为多个用户提供网上数据共享。FoxPro在网上还提供了多种保护数据库和控制数据共享的手段。在网络环境下,多个用户都可能随时访问由FoxPro管理的文件服务器中的数据, FoxPro提供了支持多个用户使用的系统命令和函数,实现对事务的并发控制和管理,从而保证数据的正确性和一致性,支持用户的分布应用处理。

§ 1.2.6 与xBASE完全兼容

FoxPro继承了DOS环境下运行的许多好的微机数据库特征,有不少的功能模块与xBASE 完全兼容。只要通过FoxPro的配置文件CONFIG.FP进行设置,或者在命令方式下通过兼容命令(SET COMPATIBLE)进行设置,就可在FoxPro下运行原

先用上述系统环境下编写的应用程序。这对于充分利用软件资源,减少应用系统开发的人力消耗方面,都是十分有意义的。

§ 1.2.7 开放的应用接口API

API是FoxPro支持系统开放性、提供系统扩展处理能力的重要组成部分,它允许用户扩展命令语言的功能,执行一些高级操作。API可以用Watcom C语言编写,用(SET LIBRARY)设置后,直接在用户程序中调用。

§ 1.2.8 先进的Rushmore技术

FoxPro使用了一种称之为Rushmore的数据存取技术,大大提高了数据存取的速度。使用这种技术允许FoxPro在索引数据库中快速地完成数据的提取,支持建立复合索引文件中的多个标志,用压缩索引格式生成小的索引文件,以加快对数据的搜索,从而允许用户处理数百万个记录的数据库。

§ 1.2.9 丰富的工具软件

FoxPro有许多附加的工具库供开发人员使用,包括项目管理器、屏幕生成器、菜单生成器、查询生成器、报表生成器、统计图形生成器,外加编译工具、移植工具、连接工具和库构造工具等,确保了应用开发的效率和质量。

第二章 xBASE向FoxPro移植

§ 2.1 xBASE编制的人事管理系统

为了更好地阐述从xBASE向FoxPro的升级, 我们用一个简单的人事管理系统程序配合讲解。该系统由人员管理主程序, 人员增加、修改、删除和查看子程序以及人事库组成。

其中用xBASE编制的系统程序为:

DBRYXG. PRG 人员修改
DBRYSC. PRG 人员删除
DBRYZJ. PRG 人员增加
DBRYGL. PRG 人员管理
DBRYCK. PRG 人员查看

数据库文件为: RSK. DBF。

在xBASE环境下运行DBRYGL. PRG可进入如下画面:

```
人事管理系统功能选择
*****
1. 增加人员          2. 删除人员
3. 修改数据          4. 查看数据
5. 列表查看          0. 退    出
*****
输入功能选择键 0---5
```

也可在FoxPro环境下运行DBRYGL. PRG, 操作与在xBASE环境下完全相同。

用FoxPro手工编制的系统程序为(执行RYGL. PRG):

RYXG. PRG 人员修改
RYSC. PRG 人员删除
RYZJ. PRG 人员增加
RYGL. PRG 人员管理
RYCK. PRG 人员查看

用FoxPro自动编程器编制的系统程序为(执行RYGL. SPR):

RYXG. SPR 人员修改
RYSC. SPR 人员删除
RYZJ. SPR 人员增加

RYGL.SPR 人员管理

RYCK.SPR 人员查看

§ 2.2 FoxPro与xBASE的兼容性

FoxPro for Windows与xBASE相比,增加了许多功能,最大的区别是,FoxPro支持许多Windows的功能,如窗口、菜单、对话框、DDE和OLE等.但FoxPro与xBASE毕竟一脉相承,FoxPro的许多命令功能与xBASE相兼容.相兼容性主要表现在数据库的处理,各种变量的处理,基本的输入输出功能等方面.

1. 相似的数据库处理

在有关数据库的记录、字段、索引、查询中,FoxPro与xBASE有相似的命令,有些命令增加了其功能.

2. 相似的变量处理

FoxPro有四种变量:日期、数值、字符、逻辑.有关这些变量的处理,FoxPro与xBASE基本相兼容.

3. 结构化编程语句

在FoxPro结构编程中,常用到DO...ENDDO,IF...ENDIF命令.还有多种选择语句等,与xBASE都是兼容的.

4. 基本的输入输出语句

FoxPro与xBASE拥有相似的输入输出语句,如?,@...SAY等.但FoxPro与xBASE的输入输出语句也有许多不同,如显示坐标、显示字模.在FoxPro编程中,输入输出一般是通过对话框进行的,这也是Windows的风格,它主要由@...GET和@...SAY命令来生成对话框,而xBASE没有此功能.

FoxPro本身为保证与xBASE的完全兼容,还附带了一些特殊功能.如设定语言兼容性的SET COMPATIBLE命令;模拟xBASE控制键的键盘宏设置功能;模拟xBASE的额外SET选项;以及各种扩展名相同,但结构不同的文件自动转换(如*.DBF);同类文件,但扩展名不同的文件自动转换(如.DBF和.FPT、.NDX和.IDX)等.

当然,FoxPro与xBASE之间仍有细微差别必须注意:

1. 错误报告

FoxPro检测一些被xBASE忽略的错误条件,如一些编辑器的控制符.

2. 新添的函数和关键词

FoxPro有许多新函数和新关键词,如果这些FoxPro函数名在xBASE中被用作UDF(用户定义函数),FoxPro会将该名字解释成内部函数,而不会执行UDF.

3. 个别命令的结果差异

xBASE中SET COLOR TO命令不带参数将重置默认颜色。

FoxPro中SET COLOR TO命令不带参数将重置Windows默认颜色。

xBASE中SET DEFAULT TO命令带驱动器和目录为参数时只有默认驱动器被设置。

FoxPro中SET DEFAULT TO命令带驱动器和目录为参数时精确设置默认驱动器和目录。

xBASE中可用空字符串, 而在FoxPro中不行。

§ 2.3 从xBASE向FoxPro升级的捷径

FoxPro虽然与xBASE有许多相似之处, 但仍有较大差别。这主要是由于FoxPro采用Windows统一的编程风格造成的。xBASE都是在DOS下运行的程序。所以从整体上来讲造成FoxPro与xBASE不同的原因, 是由于DOS和Windows的巨大差别造成的。

对国内众多软件开发人员来讲, xBASE一般都是用过的, 有了这个基础, 就对FoxPro有了一半的了解, 所以说, 从xBASE向FoxPro的升级并不难, 原来用xBASE编制的程序, 可以直接在FoxPro for Windows上运行。不过, 要想真正用FoxPro编制出Windows风格的程序, 就要突破几个难点。

用好FoxPro的最大难点, 在于FoxPro用到了Windows的统一编程规范。Windows与DOS相比是一个庞大的系统, 它为应用程序的开发制订了许多编程规范, FoxPro也是按照这个规范进行设计的。但FoxPro毕竟只用到Windows少量的规范, 所以对xBASE用户来讲, 很容易突破这些难点。

对广大的国内xBASE用户来讲, 完成向FoxPro的升级须突破以下三个难点:

1. 窗口管理

FoxPro的程序都是运行于Windows的一个窗口, 窗口是FoxPro编程中首先碰到的问题。FoxPro提供了一系列的命令函数来管理窗口。§ 3中将专门说明如何编写窗口程序。

2. 菜单系统

一个FoxPro管理程序的所有功能, 都是列在菜单中, 用户通过菜单使程序完成所有功能。§ 4中将详细说明菜单系统的编程。

3. 对话框

Windows的应用程序中, 一般所有的输入操作都是由对话框完成的。而在DOS下, 程序员常自己设计一个输入格式, 而FoxPro一般采用Windows的对话框来进行输入操作。§ 5中将详细说明对话框的编程。

FoxPro的其它功能, 如数据库操作、变量操作, 与xBASE基本相似。

总之, 对于有过xBASE编程经验的用户, 突破了窗口、菜单、对话框编程, 则会熟练地运用FoxPro来编写理想的Windows应用程序, 轻松地完成从xBASE到FoxPro的升级。

§ 2.4 Windows编程概念及风格

FoxPro for Windows是运行于Windows数据库管理系统, Windows为应用程序开发提供了许多编程规范, 故要想用FoxPro编写好的Windows应用程序, 有必要了解一下FoxPro用到的一些Windows编程规范。这将有助于您从xBASE升级到FoxPro前, 对在Windows下使用FoxPro有一个整体的了解。

§ 2.4.1 Windows的特点

Windows是一个功能强大的图形操作系统, 它与DOS相比, 使PC机用户使用界面焕然一新, 提供了不同于DOS的命令操作方式, 是通过“窗口”、“对话框”、“图标”和“菜单”等图形符号来操作。它与DOS系统相比有以下优点:

· 图形用户界面

Windows的最大特点是图形用户界面(GUI), 用形象的图形代替了DOS的命令, 使用户通过图标、对话框、菜单等实现操作。用户只需轻轻用鼠标键按一下, 就可以完成DOS一串命令的作用。

Windows的字体要比DOS丰富, 采用了可作各种缩放、修饰的TrueType字体。FoxPro利用此字体可以在屏幕上显示(或打印)各种大小的漂亮汉字和西文。

所有Windows下的应用程序, 都采用统一的Windows界面和操作风格, 用户只需学会一个Windows的应用程序, 就可以触类旁通, 方便地使用其它Windows应用程序。用户也可用Foxpro来开发出与Windows界面相似的管理程序, 而在DOS下是无法达到这种效果的。

· 与设备无关的图形界面

众所周知, 计算机的外设种类繁多, 有各种显示器, 不同型号的打印机等, 以前在DOS下编写一个管理程序, 常为不同的外设搞得焦头烂额, 而在Windows下就没有这些烦恼了。

Windows提供了一种与设备无关的图形界面, 使应用程序不再考虑众多的外设, 只需在一种外设上能运行, 在其它外形上同样可以。与各种外设打交道, 是由Windows内部完成的, 编程人员不再考虑。因此, 用FoxPro编写的管理软件, 可以在任何显示器, 打印机上运行。

· 完善的内存管理

在DOS下编程, 如程序较大, 常为内存不够而头痛, 而在Windows下几乎不用再费心了, 因为Windows已突破了DOS 640k的内存限制, 可以使用计算机内所有内存, 并且可以把硬盘作为虚拟内存使用。对FoxPro编程人员来讲, 很少再考虑内存问题, Windows在内部已对内存做了完善的管理。

· 支持多媒体

Windows支持多媒体技术,可以处理声音,图象,文字等各种媒体。FoxPro本身没有多媒体处理功能的,但在Windows下可以用FoxPro编写一个多媒体数据库,因为Windows提供了两个卓越的功能:动态数据交换(DDE)、对象链接与嵌入(OLE)。利用DDE和OLE技术,FoxPro可采用Windows其它应用程序的多媒体数据,并进行管理,这是在DOS下所无法想象的。

· 多任务操作

Windows是一种多任务的操作系统,可同时执行多个程序。这为用FoxPro编写的应用程序提供了极大的方便。

§ 2.5 FoxPro基本编程概念

§ 2.5.1 FoxPro窗口处理

窗口是Windows应用程序最基本的输入输出对象,一切操作都是在窗口中完成的。Windows的窗口有统一的管理方法,FoxPro也是按照Windows的窗口管理机制来编程。

在用FoxPro编写窗口程序时,首先要用DEFINE WINDOW命令定义窗口属性,类似的窗口编程语句包括: SHOW WINDOW、ACTIVATE WINDOW等。

在FoxPro中,窗口有两类,一是用户自定义窗口,这些窗口由用户自己定义;二是FoxPro系统的窗口,其处理与自定义窗口相似。

刚接触FoxPro窗口编程时应该了解窗口的基本属性,如窗口名字、窗口位置和大小、父子窗口、窗口字模和字体、窗口标题、窗口边框形状、窗口可关闭性等。

§ 2.5.2 FoxPro菜单处理

菜单是一个命令列表,用户可以用鼠标和键盘来选择。每当用户选择了菜单的一个选项时,程序就执行指定的功能。FoxPro生成的菜单与Windows的菜单相似,遵循Windows统一的菜单机制。FoxPro生成的菜单有两类:菜单条、弹出式菜单。在FoxPro中,不但能自定义菜单,还可使用FoxPro的系统菜单。

FoxPro的菜单系统包括菜单条(MENU)、菜单笈(PAD)、弹出式菜单(POPUP)和弹出式菜单的选择项(BAR)等。

§ 2.5.3 FoxPro对话框处理

对话框是一个临时窗口,在Windows中常用做用户输入、选择,使用之后立即删除。对话框由复选框、列表框、弹出式菜单、按钮、单选钮、无形按钮、微调器、文本编辑区等组成。

在FoxPro中用@...GET命令定义对话框中的各种控制项,用SHOW GET或SHOW GETS命令可显示控制项,但不激活。用READ命令可激活并显示控制项。

§ 2.5.4 FoxPro的坐标系统

Windows的坐标是很复杂的,这是由于Windows采用图形界面,其坐标的灵活性很大。而DOS下程序一般是字符方式,其坐标是固定的。

Windows的程序一般都运行在一窗口中。对FoxPro来讲每个窗口都有自己的坐标系统。一个窗口的左上角是坐标的原点(0, 0),从左到右,列坐标增加,从上到下,行坐标增加。

窗口的字模大小决定了坐标的单位,窗口中的一行等于窗口字模的高度。由于FoxPro采用各种字模,每个字模的字母宽度是不一样的,如'i'字通常要比'W'字母窄,所以窗口一列等于当前字模中各字母的平均宽度,我们可从FoxPro中的联机帮助中按FONT主题,查询各字模的详细情况,如果选用汉字字模,坐标单位就简单了,因为汉字的宽度都是一样的,所以窗口中的一行等于汉字字模高度,窗口一列等于汉字字模的宽度,在FoxPro中为了精确定位坐标,坐标可以使用小数,这与DOS相比是一大进步。

注意,对打印机来讲,第一行是从1开始,而窗口坐标则是从0开始。

§ 2.5.5 FoxPro的颜色

FoxPro可用两种方式指定颜色,一是颜色代码,如N代表黑色,二是用红、绿、蓝三基色来合成任意颜色,这是Windows独有的功能,使用更为方便。

我们知道大自然所有颜色都是由红、绿、蓝三基色组成的,不同比例的红、绿、蓝组成了五彩缤纷的世界。在FoxPro中也可以用三基色的原理生成任意颜色。

在FoxPro中红(R)、绿(G)、蓝(B)各为0—255的数值表示,如红色R,从0到255,由浅红逐渐过度到深红。不同颜色的三基色合成了一种颜色,如RGB(0, 255, 0)是绿色,RGB(255, 0, 0)是红色。

· 颜色对

FoxPro中的颜色总是成对出现,代表前景色和背景色。FoxPro中能使用以下的颜色代码:

颜色	代码	颜色	代码
黑色	N	蓝色	B
无色	X	棕色	GR
青色	BG	绿色	G
洋色	RB	红色	R
白色	W	黄色	GR+
反向显示	I	下划线	U
背景色变亮	*	前景色变亮	+

一个颜色对(color)的前一个颜色指前景色,后一个颜色指背景色。

例:用颜色代码表示红色前景白色背景:

R/W

例:用RGB三基色来表示红色前景白色背景:

RGB(255, 0, 0, 255, 255, 255)

· 颜色对列表

在FoxPro中, 一个颜色对列表由1—10个用逗号分开的颜色对组成。

例: 用颜色代码表示一个颜色对列表, 包含4个颜色对:

W+/R, R/W, W+/RB, N+/N

例: RGB表示一个颜色对列表, 包含3个颜色对:

RGB (0, 10, 0, 255, 255, 255),

RGB (120, 20, 0, 200, 0, 25),

RGB (0, 255, 255, 0, 0, 0)

· 颜色模式

一个颜色模式(color scheme)由10个颜色对组成。

在FoxPro中用SET COLOR, SET COLOR OF, SET COLOR SCHEME, SET COLOR SET命令可以设定颜色。FoxPro中的许多命令如@...SAY可设要显示元素的对应颜色。

§ 2.5.6 FoxPro的打印管理

Windows采用图形设备接口(GDI), 对打印机来讲, Windows管理打印机, 应用程序如FoxPro一般不直接控制打印机。Windows的打印管理器(Print Manager)管理所有打印机。

FoxPro要打印的内容, 在图形设备接口中生成代码, 这些代码是Windows内部所使用的语言。这些代码确定了Windows需要知道打印内容。但不告诉打印机如何去打印。然后这些代码输出到打印机驱动程序中, 它会把GDI代码转换为打印机懂得的指令。这些打印机懂得的指令转送到打印机管理器(Print Manager), 管理器把这些打印代码存入磁盘中的一个假脱机文件中。如果为特定打印机创造假脱机文件的速度比将此文件送到打印机快, 管理器就会管理一个打印队列。

管理器把假脱机文件送入打印机中, 打印出应用程序要打印的内容, 然后从磁盘上删除假脱机文件。

以上是FoxPro在Windows中打印的一般过程。一个打印机驱动程序一般由Windows加载。可见FoxPro的打印编程十分简单, 丝毫不用考虑打印机的型号。用户用FoxPro编写打印程序时, 不用再考虑打印机的任何内容, 直接用FoxPro的SET DEVICE、SET PRINTER, 将输出由屏幕转向打印机就可以了。

第三章 FoxPro的窗口编程

窗口是FoxPro程序的主要输入输出对象,本章介绍窗口编程的所有内容,包括建立窗口、改变窗口、释放窗口等,这些对xBASE用户来说都是全新的内容。通过本章的学习,我们可以设法把xBASE的单窗口界面升级到FoxPro的多窗口界面中来。

§ 3.1 FoxPro有关窗口的命令函数

以下命令、函数用以编写窗口程序:

ACTIVATE WINDOW: 显示并激活一个或多个自定义窗口或FoxPro系统窗口;

CLEAR WINDOW: 在屏幕上清除窗口,并从内存中释放所有用户自定义窗口;

CREATE SCREEN: 创建一个屏幕,显示屏幕格式窗口;

DEACTIVATE WINDOW: 在屏幕上清除窗口,并把指定窗口处于非活动状态,但在内存中保留这些窗口定义;

DEFINE WINDOW: 创建一个窗口并指定属性;

HIDE WINDOW: 隐藏一个处于活动状态的用户定义窗口或FoxPro系统窗口;

MODIFY SCREEN: 显示屏幕格式窗口,编辑一屏幕;

MODIFY WINDOW: 修改一个已定义窗口;

MOVE WINDOW: 把窗口移动到指定位置;

RELEASE WINDOW: 从内存中释放指定窗口,并在屏幕上清除;

RESTORE WINDOW: 从一窗口文件或备注字段中,把窗口定义恢复到内存中;

SAVE WINDOW: 把指定窗口定义存入一窗口文件或备注字段中;

SET BORDER: 定义窗口或菜单的边框类型;

SHOW WINDOW: 在不激活的情况下显示指定窗口;

SYS(2016): 返回由最近一次SHOW GETS WINDOW指定的窗口名;

WBORDER(): 返回一个窗口是否有边框;

WCHILD(): 返回当前或指定的父窗口的子窗口或子窗口名;

WCOLS(): 返回指定窗口的列数;

WEXIT(): 返回窗口是否存在;

WFONT(): 返回一个窗口当前字模的名称、大小和字体;

WLAST(): 返回在当前窗口以前激活的窗口名,或判断给定窗口是否为前一个激活窗口;

WLCOL(): 返回窗口左上角的列坐标;

WLROW(): 返回窗口左上角行坐标;

WMAXIMUM(): 判指定窗口是否最大化;

WMINIMUM(): 判指定窗口是否最小化;

WONTOP(): 如果指定窗口在所有其它的活动窗口之上, 则返回真, 否则为假;
WOUTPUT(): 如果当前输出是直接输出到指定窗口时, 则返回真, 否则为假;
WPARENT(): 返回当前或指定窗口的父窗口;
WREAD(): 如果指定窗口与当前READ命令有关, 则返回真, 否则为假;
WROWS(): 返回指定窗口的行数;
WTITLE(): 返回指定窗口的标题;
WVISIBLE(): 如果指定窗口没有隐藏, 则返回真, 否则为假;
ZOOM WINDOW: 改变窗口大小;

§ 3.2 Windows关于窗口的管理

Windows提供了一整套的窗口管理机制, FoxPro可以充分地利用其提供的方法, 建立Windows统一格式的窗口, 在此, 我们首先了解Windows的窗口处理方法。

§ 3.2.1 Windows字体

a. Windows字符集

Windows提供的字体, FoxPro可以很方便地使用。Windows所有字符集多数为ASCII字符集, 所有字符都用32 ~ 127这96个数字来表示, 这与DOS下的字符一样。Windows共提供以下4种字符集:

Windows字符集: 它是应用程序最常用字符集, 该字符集与ANSI字符集完全相同。

OEM字符集: OEM字符集中的第32 ~ 127号字符与ASCII和Windows字符集完全相同。OEM字符集中的其它字符(0 ~ 31和128 ~ 255)与DOS下程序显示一致, 这些字符完全不同于Windows字符集。

Symbol字符集: 此字符集字符用于表达数字和科学公式。

特殊的字符集: 如EBCDIC字符集, 完全不同于OEM或Windows字符集。

b. Windows字体

Windows提供了三种字体: 点阵(Raster)字体、矢量字体、TrueType字体。

点阵字体以位图形式存储, 位图的分辨率是根据输出设备而定的。此种字体有很大局限性。

矢量字体以GDI调用集合方式存储。

TrueType字体以确定字符轮廓的点和线的集合方式存储, 这种字体是Windows 3.1新提供字体, 此种字体使显示和打印成为一体, 是一种所见即所得的字体。在FoxPro中所讲的字体, 通常指此种字体。TrueType字体可以被随意进行缩放、旋转。可以用在不同的打印机上打印, 与设备无关。TrueType缩放字体没有失真, 使用起来极其方便。

Windows 3.1提供了13种英文体, 中文Windows还提供了2种TrueType中文字体。需要时, 用户可加装其它TrueType字体。

§ 3.2.2 Windows关于窗口编程

有关窗口的编程,Windows提供了统一的编程格式。

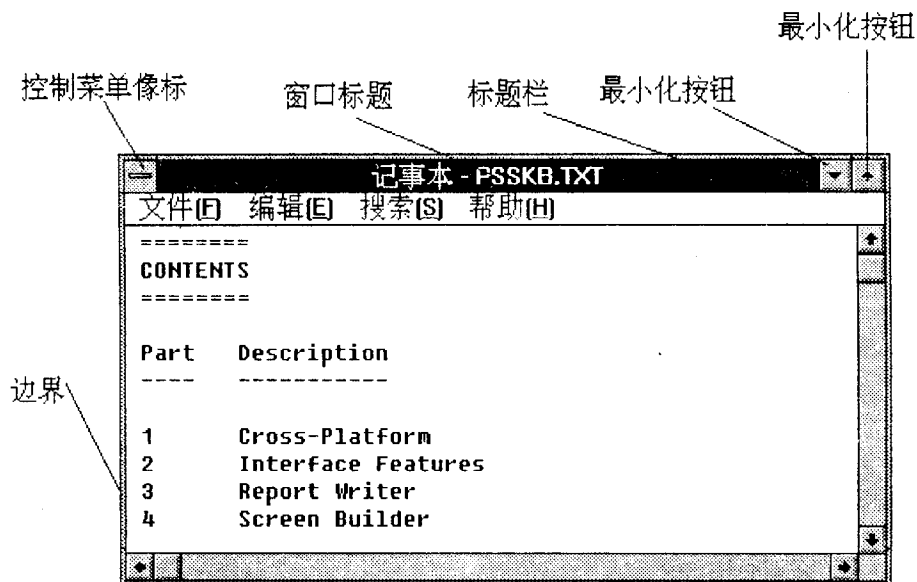


图3.2.2 Windows窗口组成

首先,定义一个窗口的属性。这些属性包括窗口的形状、光标、菜单名等。

其次,创建窗口。创建一个窗口需许多参数,包括窗口名称、窗口标题、窗口风格、窗口位置等。

第三,显示、更新窗口。创建的窗口,还不能在屏幕显示,必须调用显示函数在屏幕上显示,也可更新窗口的参数。

第四,关闭窗口。窗口的定义是存于内存中,在窗口的处理中,它一直保存在内存中,一旦要退出窗口处理,不但要在屏幕上清除窗口,还应从内存中清除窗口定义。

§ 3.3 FoxPro如何创建窗口

命令DEFINE WINDOW用于创建一个用户定义窗口并定义其属性。

该命令的使用语法如下:

```
DEFINE WINDOW <窗口名>  
FROM <行1,列1> TO <行2,列2> | AT <行3,列3> SIZE <行4,列4>  
[IN [WINDOW] <窗口名2> | IN SCREEN | IN DESKTOP]
```

```

[Font<expC1>[, <expN1>]
[STYLE <expC2>]
[TITLE <expC3>]
[HALFHEIGHT]
[DOUBLE | PANEL | NONE | SYSTEM | <边界>]
[CLOSE | NOCLOSE] [FLOAT | NOFLOAT]
[GROW | NOGROW]
[MDI | NOMDI]
[MINIMIZE]
[SHADOW]
[ZOOM | NOZOOM]
[ICON FILE <expC4>]
[FILL FILE <bmp文件>]
[COLOR SCHEME <expN2> | COLOR <color pair list>]

```

窗口用该命令定义之后,放在内存中,窗口名列表放在名为Window弹出菜单的底部。第一个创建的窗口名字放在表的顶部,以后定义的随后放置。创建窗口以后,并不显示在屏幕上,只有用ACTIVATE WINDOW或SHOW WINDOW命令才可显示。下面分别介绍如何定义一个窗口。

· 窗口名

每个窗口都赋予一个名字,窗口名字最长为10个字符,名字必须由字母或下划线开始,不可用数字开始。名字中可包含任意数字、字母、下划线。

· 窗口大小位置

窗口大小可用两种方式定义。

a. FROM <行1, 列1> TO <行2, 列2>

<行1, 列1>坐标是在桌面系统或主FoxPro窗口中自定义窗口的左上角位置,<行2, 列2>为右下角坐标。左、右角坐标决定了窗口的位罝及大小。此处所讲坐标的单位为字模大小,由于Windows的字模大小随意可变,所以如果字模大小改变,相同的坐标,其实际位置不同。

在此种定义窗口大小形式中,坐标的单位由窗口双亲窗口字模决定,窗口双亲可以是另一自定义窗口或FoxPro主窗口。

b. AT <行3, 列3> SIZE <行4, 列4>

<行3, 列3>为窗口左上角坐标,<行4, 列4>是窗口的行、列数目,这种方式同样可定义窗口的位罝及大小。

与上一种定义窗口大小不同,此种方式坐标单位虽然也是字模大小,但此字模大小由本窗口的FONT和STYLE定义。若用户没有定义本窗口字模大小,则窗口字模状态为系统字模即10点。

· 窗口的父窗口

在Windows中屏幕上可有多多个窗口,它们可以互为父子关系。在FoxPro中定义的窗口也有同样关系。

a. IN [WINDOW] <窗口名2>

<窗口名2>为父窗口名,此选项使当前定义的窗口作为名为<窗口名2>的子窗口。父、子窗口互为一个整体,放入父窗口的子窗口不可移出父窗口,如父窗口移动了,则子窗口随之移动。

如果定义了父窗口,则上面用于决定窗口大小及位置的选项FROM...TO和AT...SIZE的起始坐标相对于父窗口,而与桌面或FoxPro主窗口无关。

b. IN SCREEN、IN DESKTOP

自定义窗口在缺省状态下是放于桌面或FoxPro主窗口。IN SCREEN指定把定义窗口放于FoxPro主窗口,而IN DESKTOP则把窗口放于桌面。

· FONT和STYLE

此两项决定窗口字体及字模大小与属性。前面讲过Windows中最常用的字体为TrueType字体,这里就是要决定FoxPro所用的TrueType字体属性。

FONT <expC1>[, <expN1>]用来选定本窗口的字模及大小。<expC1>是字模名称,通过此选项可选不同的字体。<expN1>为该窗口字模大小,缺省为9点。

若缺省了FONT选项,则字模大小默认为10点。若定义的字模无效,则系统用相同属性的字体来代替。

STYLE <expC2>定义该窗口输出的字体属性,缺省标准体为黑体。以下为<expC2>的字符与其对应的字体属性:

<expC2>字符	对应的字体属性
B	黑体
I	斜体
N	普通
O	轮廓线
Q	不透明
S	阴影
-	删除线
T	透明
U	下划线

用户可在命令中定义多个字符,以合成不同的修饰。例如:

```
DEFINE WINDOW test FROM 2,2 TO 20,30 FONT '黑体' 40 STYLE 'BI'
```

上句定义了一个名叫“test”的窗口,其坐标从(2,2)到(20,30)的方框,该窗口字体为汉字黑体,大小为40点,是用黑体且斜体形式显示。

· 窗口标题

窗口的标题位于窗口顶部,说明窗口的功能或名字。

TITLE <expC3>选项中<expC3>为标题的字符串。HALFHEIGHT定义窗口有半高标题条。当DEFINE WINDOW命令中有SYSTEM或FONT选项时,将不使用半高标题,而HALFHEIGHT选项指窗口使用半高标题,而不管SYSTEM和FONT。

· 窗口边界设置

自定义窗口的缺省边界是单线,用户根据需要设置不同的边界。

a. DOUBLE边界,此种边界使窗口外框为双线边界。

b. PANEL边界,此种边界使窗口外框为一条宽的边界。

c. NONE边界, 则窗口无边界。

d. SYSTEM边界, 此种边界使窗口外表与FoxPro系统窗口外表一样。在这种情况下若有GROW、ZOOM等选项, 则象系统窗口一样把相应的窗口控制项放在窗口顶部边界中。

e. <边界框>: 此时用户可自己定制边界。边界串是一系列ASCII码值; [

· 窗口关闭选择

CLOSE选项使窗口可以关闭。关闭一窗口不但从屏幕上清除, 而且从内存中清除。NOCLOSE使一窗口不被关闭, 这也是系统缺省状态。

若窗口被SYSTEM定义成系统窗口外观, 则要关闭窗口只需单击左上角隐含的关闭盒。关闭一个窗口, 也可按Ctrl+F4或从菜单中选关闭项。

· 窗口浮动性

FLOAT | NOFLOAT定义窗口是否为可浮动, 缺省时窗口不浮动, 如定义为FLOAT, 则为可浮动, 用鼠标点窗口标题栏拖动窗口, 或按Ctrl+F7移动窗口。

· 窗口大小可变性

GROW | NOGROW用来选择窗口是否可改变大小, 缺省态为NOGROW, 窗口不可改变, 如定义为GROW, 则可改变窗口的大小。

· MDI多文本界面

MDI是一种多文本界面, 它可以显示和加工多个文本。每个文本界面应用程序有一个主窗口, 在这里可以打开并处理多个文本。每个文本显示在一个子窗口。子窗口的主要区别在于它们不能移到主窗口外。在应用程序中, 主窗口称边框窗口, 它与普通主窗口的主要区别是其用户界面由特殊子窗口来填充。Windows系统负责维护MDI的子窗口, 并控制MDI界面, 所以应用程序不需保存MDI界面的大量信息。

在DEFINE WINDOW中, MDI选项使该窗口是MDI窗口, 它允许存在多文本窗口。NOMDI为系统缺省状态。

· 最小化

MINIMIZE选项使定义窗口可最小化。最小化后, 窗口缩小为一个图标。

· ZOOM | NOZOOM

ZOOM使定义的窗口可以扩展至整个桌面或整个FoxPro主窗口。系统缺省值为NOZOOM, 是不可扩大的。

· 图标

当用户在DEFFINE WINDOW中含有MINIMIZE, 则最小化后需定义一个图标, 以代表最小化后的窗口。其格式为: ICON FILE <expC4>, <expC4>为图标文件(.ICO)。

图标是一个小的图形图像,它代表应用程序最小化的窗口。对FoxPro编程人员来讲,图标是一个资源,在编程时,要先用Microsoft图像编辑器创建一个图符文件,扩展名为.ICO。

· 窗口填充背景

FILL FILE <bmp文件>用来定义窗口背景。bmp文件指一个位图文件。

位图是指用位模式来形成的图像,而不是用线模式形成图像。在Windows中有两种位图:

- a. 设备相关位图,它是一种位模式图案,存放在内存中,可以在特定输出设备上显示。
- b. 设备无关位图,它描述一个图像的外观,而不是在特定显示设备上的显示形式,它可以在任何显示设备上输出。

在FoxPro中所用到的位图文件,主要是用Microsoft的图像编辑器来画出位图图像,并存于一个位图文件中,其文件后缀名为.bmp。

在DEFINE WINDOW中定义了位图文件,则自定义窗口的背景将用此文件对应的位图铺满。

· 定义窗口颜色

COLOR SCHEME <expN2> | COLOR<色彩对表>选项将已存在的色彩表的号码或一套色彩对表定义为窗口颜色。

FoxPro提供了一个全面控制颜色的完整命令集,用户可以定义FoxPro系统的接口(如系统菜单、窗口、对话框等)的颜色,也可定义用户的目标(如菜单、窗口等)的颜色。

有两种设置颜色的方法:一是通过颜色对话框来设颜色,二是用SET COLOR命令来设颜色。

a. 色彩对(color pair)

色彩对由前景颜色和背景颜色组合而成,颜色用简写字符来表示,星号(*)用来表明是闪烁还是亮(这要根据SET BLINK的状态而定)。加号(+)表示高亮度,下面为有效的颜色及其代码:

颜色	代码
黑色	N
空白	X
蓝色	B
棕色	GR
深色	BG
绿色	G
深红	RB
红色	R
白色	W
黄色	GR+

在单色监视器上只有四种有效颜色:白色(W),黑色(N),下加线(U),反向显示(I)。在输入口令时,空白(X)是很有用处的。