

可以列出这些字体,但其中有些字体可能是不能在 FoxPro for Windows 下使用的。在使用某一非标准字体之前,最好用 AFONT() 函数确证该字体是否可用。

AFONT() 函数的使用格式是: AFONT(<数组>[, <指定字体>[, <指定字体大小>]]) , 它返回当前有效的字体、或指定字体可使用的字体大小到一个数组中,或返回指定字体大小对该字体是否有效。

4.更好地使用文本编辑器

要快捷地选择一行文本,包括行尾的回车(CR)返回(LF)字符,可通过三点按该行来选中;

移动或拷贝被选文本,通过“拖曳-放下”而不是“剪切-复制-粘贴”的动作来完成将更为简捷。在拷贝被选文本时,应在“拖曳-放下”的同时按住 Ctrl 键不放完成。“拖曳-放下”动作所作用的文本将不放置到剪裁板中,你使用“拖曳-放下”动作时,将保持剪裁板中的原有的内容不被重写;如果要将大段程序代码进行缩进或非缩进,可先选择该段文本,再在 Text 菜单基中选择 Indent / Undent 选项。

5.解决图象的色彩与设计效果不同的问题

有时候,你可能发现正在显示的图象的色彩与你设计的预期效果略有出入,这是由于显示和设计时使用的调色板不同引起的。你不妨执行 SET PALETTE OFF 命令使用该 .BMP 图象或 OLE 对象被设计时的调色板而不是使用 FoxPro 默认的调色板来显示图象。但需注意的是,该命令将影响其后所有的图象的显示。

6.有效地使用墙纸

有些位图可能在一种分辨率下看起来很好,而在另一种分辨率下却失真十分严重。因而当使用一个位图作为墙纸时,最好使用在不同屏幕分辨率下看起来都不错的位图。

如果在你的应用程序中使用位图作为墙纸,而且希望在不同的屏幕分辨率下,都能保证特定的屏幕对象总在墙纸上特定的位置,例如:一个按钮控制必须总是落在一杆秤的称钩上,则可针对不同分辨率和显示方式,分别设计相应的位图文件,调整好位置,用不同的文件名保存;然后在应用程序的开始处,用 FONTMETRIC(1) 函数或其它函数或实用程序,检测当前的屏幕分辨率和显示方式,然后用复制命令将对应的位图文件复制成你的应用程序中调用的位图文件名,以便将之作为墙纸使用。

7.FoxPro 主窗口属性的设置

你可以用 MODIFY WINDOW SCREEN 命令修改 FoxPro 主窗口的属性。这个命令允许你修改包括窗口字体、字体大小、字体风格、墙纸、色彩、图标文件、窗口标题、

大小等等在内的窗口属性。你也可以用 ZOOM WINDOW SCREEN 命令最大化或最小化 FoxPro 主窗口。

MODIFY WINDOW SCREEN 命令恢复 FoxPro 主窗口大小为它启动时的大小,在退出 FoxPro 时,主窗口的大小将被保存到 Windows 系统目录的 FOXPRO.INI 文件中。

ZOOM WINDOW SCREEN NORM 命令则恢复 FoxPro 主窗口到你执行 ZOOM WINDOW SCREEN MAX 或 ZOOM WINDOW SCREEN MIN 命令之前的该窗口大小。

注意:你可以通过执行多个 MODIFY WINDOW 命令来调整一个窗口的不同属性,后一命令并不影响前一命令对窗口属性的设置,除非在后一命令中对同一属性再次进行了设置。

如想通过系统菜单而不是执行命令来改变主窗口的字体及其大小和风格,其秘招是:按住 Shift 键不放,这时 Text 菜单基中的 Font 选项将临时变成 Screen Font... 选项,选择该选项即可改变主窗口的字体及其大小和风格;如果是在屏幕生成器中,则可以通过在 Window Styles 对话框中建立一个 DESKTOP 窗口并运行之来设置 FoxPro 主窗口的字体。

8.控制图标(最小化窗口)的屏幕位置

一般地,在 FoxPro for Windows 中,将一个窗口最小化后,其图标将落在屏幕的左下角,如果希望将一个图标放到一个特定的地方,不妨仿照如下处理:

```
DEFINE WINDOW x FROM 10,10 TO 20,20 MINIMIZE FLOAT CLOSE
ACTIVATE WINDOW x NOSHOW &&非显式激活窗口
ZOOM WINDOW x MIN &&该窗口最小化为图标
MOVE WINDOW x TO 1,1 &&将窗口移到左上角
SHOW WINDOW x &&显示该图标
```

9.让鼠标右按钮发挥作用

在 FoxPro for Windows 中,鼠标右按钮一般没有使用,因而你可以定义鼠标右按钮,让它仿真某些键盘按键的功能,以充分发挥它的作用。

譬如,将鼠标右按钮定义成回车键,可用如下一条命令定义:

```
ON KEY LABEL RIGHTMOUSE KEYBOARD
CHR(13)
```

这样,你就可以按鼠标右按钮代替按回车键来执行命令、选择对象等操作。

10.快速同时关闭或临时隐藏多个窗口

多个窗口的同时关闭,往往需通过执行命令 CLEAR

ALL 或 CLEAR WINDOWS 来完成。如果需要在系统菜单中快速地同时关闭多个窗口,其秘招是:按住 Shift 键不放,这时 File 菜单基中的 Close 选项将临时变成 Close All 选项,选择之,即可将打开的所有窗口关闭。

类似地,你可以同时按住 Shift+Ctrl+Alt 组合键临时地隐藏所有的窗口,以观看 FoxPro 主窗口中的内容,但一放开这些组合键,隐藏的窗口又显示出来。

11. 口令保护设计的技巧

如果要给你的应用一个口令保护,不妨在请求口令的地方,将口令字符的字体设置成相当小以致于不能阅读,或使用一种密码字体,该字体无法阅读,或这种字体下所有字符都是一样的。

12. 建立一个最前面 (Frontmost) 窗口

你可以用 IN DESKTOP 子句建立一个最前面 (Frontmost) 窗口,这个窗口存在于 FoxPro 主窗口之上,它与正常的用户窗口不同的是:最前面窗口总是保留在任何其它窗口的最前面,即使在该窗口之后你又打开了其它窗口也是如此。

WONTOP() 函数将总是返回最前面窗口名。在事件驱动程序设计中这是特别重要的。IN DESKTOP 子句也能用来模拟一个浮动调色板 (floating palette), 浮动调色板是 Windows 和 Macintosh 软件产品的一个共有特性,它总是处于活动窗口的前面;也可用来设计诸如中文之星 2.0 的链条菜单一样的产品。

最前面窗口的位置由 Windows 坐标系统决定,而普通的用户自定义窗口则使用 FoxPro 坐标系统。

13. 重建过时的坏索引

过时的被破坏了的索引可能会使一个正常的数据库操作出错,这时你应该试图重建该索引,但是最好是重新建立一个索引而不是使用 REINDEX 命令来重建索引,因为 REINDEX 命令无法对一个被损坏了的索引文件重建索引。

14. 不定期重建 .CDX 文件

经常不定期地重建 .CDX 文件,可以减少 .CDX 文件的大小,并优化查询性能。当重建 .CDX 标记 (TAG) 时,你首先应该用 DELETE TAG ALL 命令删除所有已经存在的标记,再重新建立。

15. 定期紧缩数据库及备注文件

应该定期紧缩 (PACK) 你的数据库,它将清除已作删除标记的记录,减少数据库的大小并优化性能,相应地还

应使用 PACK MEMO 命令将被删记录占用的备注字段空间清理出来,以减少相应的备注文件的大小,特别是对那些你频繁地进行修改编辑的备注字段来说更应如此。

16. BROWSE 和 SCAN 命令的优化

如果想要提高 BROWSE 和 SCAN 命令的性能,可定期地用 SORT 命令以经常要使用的记录顺序对数据库进行排序。

17. 多库查询的优化

为更有效率地进行多库查询,应基于联结条件对被查询的各数据库分别进行排序。

18. 索引与查询

如果没有指定数据库的索引,系统将自动优化对它的查询。因此,你可执行不带对象名的 SET ORDER TO 命令,移去指定的索引,以优化查询。

如果你需要执行 SET DELETE ON, 便于以后的数据库操作对已被作删除标记的记录不发生作用,则你应该建立一个基于 DELETED() 表达式的 .CDX 索引标记,这样可提高查询的速度。

19. 用 SQL 命令向数据库添加记录

应该用 INSERT - SQL 命令向数据库中添加记录。较之 APPEND BLANK 和 REPLACE 命令的组合使用,INSERT - SQL 命令将提供更好的性能和数据完整性。

20. 屏幕输入输出的优化

为保证良好的性能,在 FoxPro for Windows 中设计屏幕的输入输出时必须仔细考虑。在 FoxPro for MS-DOS 环境,屏幕显示为 25 行 80 列,每次就有 2000 个屏幕元素需要刷新;而在 Windows 环境,标准的 VGA 模式将有 640 x 480 个象素点,每次屏幕刷新时,将有超过 300,000 个象素点需要被重画和刷新,较之字符模式,比率大约为 150:1,其刷新效率将低得多。因此,一般设计中,当屏幕输出被改变时,应使用 SHOW GETS 命令以便仅仅刷新屏幕的 GET 对象,而不对非屏幕对象作无谓的刷新操作。如果在你的应用中使用了具有许多对象的屏幕,或者你的机器主频较慢,那么使用 SHOW GETS 命令将有戏剧性的效果。

但 SHOW GETS 命令对没有被改变的 GET 对象也进行刷新,这也是多余的工作。因而更有效的方案是,用 SHOW GET 命令仅对已作改变的 GET 对象进行更新,而不是用 SHOW GETS 刷新屏幕中所有的 .GET 对象。

另外,在 FoxPro for Windows 中用位图文件建立的

图象按钮和图象复选框,用 SHOW GETS 命令刷新这类屏幕控件时,将要耗费更长的时间,设计时要加以考虑。

21.尽可能少地运行其它应用程序

减少运行其它 Windows 应用程序的个数,可以大大提高 FoxPro 应用程序的运行效率。Windows 平台是一个多任务协作环境,如果有过多的其它应用程序被运行,意味着分配给 FoxPro 的系统资源相应减少了,这将大大降低其运行的速度和性能。

22.使用编译型常量

使用 #DEFINE 指示器定义编译型常量,可以减少在你的应用程序中使用内存变量的个数,节省内存的占用。

23.墙纸将消耗过多内存

因为在窗口中使用墙纸(wallpaper)将耗费许多的内存,因而为减少你的应用程序对内存的消耗,在屏幕窗口的色彩设计中最好不要选用墙纸。

24.减少过多的色彩对内存的占用

如果在你的应用程序中过多地使用不同的色彩也将耗费许多的内存,而且不使用系统本身提供的色彩模式(color schemes)而直接利用 RGB() 函数等进行色彩的调配时,其内存占用更加明显。因而,只要可能,你应尽量使用 16 色的 .BMP 文件,它将只需要很少的内存占用,并且在 VGA 显示器上能显示得最好。除非你为了在跨平台的应用中使用 MS-DOS 的色彩集以保持视觉上的兼容性,你应该允许用户用 Windows 的控制板选择他自己的色彩。

25.给 FoxPro 分配足够的内存

FoxPro 在启动时,并不为它所有的资源分配内存,系统将根据 FoxPro 的配置,从 Windows 的自由内存池中给 FoxPro 划分出一段内存空间。因此,你应该在 CONFIG.FPW 配置文件中用 MEMLIMIT 为你的应用程序设置足够的内存。

26.如何为 FoxPro for MS-DOS 分配所有自由内存

如果你将 FoxPro for MS-DOS 产品对应的 PIF 文件中的 KB Desired 设置项设置为 -1, Windows 将为它分配所有的自由内存。

27.程序控制在屏幕对象间移动

使用系统内存变量 _CUROBJ 可以控制将当前编辑域移动到特定的屏幕对象。

将下面的代码放到某一 @...GET 对象的 VALID 代码片段中,它将使编辑光标在不离开该对象的前提下改变另一个屏幕对象对应的时间变量 X 的值。

```
x = Time()
_CUROBJ = _CUROBJ
RETURN .T.
```

当你设置 _CUROBJ 的值时,应避免在代码中直接使用实际的对象号,而可使用诸如: _CUROBJ = OBJNUM(x) 的方式间接引用对象号;否则当你在屏幕生成器中改变某对象的顺序时,该对象对应的对象号也发生了变化,可能将引起不希望的结果。

28.如何建立无线按钮

在 FoxPro for Windows 中,无线按钮(Invisible)是作为按钮(push)的一种特殊类型而放置在按钮工具中的,因此,如果要在屏幕上设计无线按钮,必须选择按钮工具,在打开相应的 Push Button 对话框中,选择 Type 方框中的 Invisible 按钮类型,即可在屏幕上产生无线按钮。

29.建立图象按钮

在 FoxPro for Windows 中,你可以使用 SHOW GET 命令设计诸如按钮、复选框、无线按钮等屏幕图象按钮。

而且,你可以为这些控件的三个状态(激活、选择、非活化)分别指定一个对应的 .BMP 位图文件,如:

```
SHOW GET mybtn,7PROMPT 'EXITEN. BMP,
EXITSEL. BMP, EXITDIS. BMP'
```

将为图象控件(可能是按钮、不可见按钮或无线按钮等控件之一)的第七个按钮的三种状态分别设置不同的位图文件进行显示。

30.对象的自动色(Automatic)的功用

屏幕对象的色彩设置中其中有一种叫做 Automatic 的自动色,它的作用就是保证屏幕对象的色彩,包括填充色(Fill Color)、画笔色(Pen Color)或窗口背景色(Background Color)等,与用户临时选用的当前色彩模式保持一致。也就是说,当用户选用的色彩模式发生变化时,自动色将自动调整屏幕对象相应的色彩与选用的色彩模式保持一致。系统默认的色彩为自动色。

如果你希望用户能在应用时通过选用或定义色彩模式的方式再设置屏幕对象的色彩,则一定要保证该对象的色彩被设置成了自动色。