VFP 环境下创建应用系统工具栏

宋　群　（南京电子技术研究所　210013）

摘要：本文主要介绍在 VFP 环境下，创建所要设计的应用系统的工具栏的两种设计方法。

关键词：类　工具栏　属性　方法　事件驱动

一、引言

VISUAL FOXPROW 3.0 提供的创建工具栏手段，概括起来有两种：一种是根据应用系统的需要，使用 VFP 提供的工具裁剪 VFP 系统所有的工具栏中的工具按钮，组成应用系统自己的工具栏。这样组成的工具栏，可以作为应用系统的工具栏，但其工具按钮的功能只能是 VFP 系统早就定义的功能，用户无法定义自己的功能。另一种方法是使用类设计技术设计用户屏幕表单集的工具栏，贵刊 1998 年第 7 期中《创建 VISUAL FOXPRO 表单工具栏》对这种方法有较详细的介绍。但这种方法设计的工具栏尽管其按钮的功能可由用户自己定义，但仅对表单集起作用，而不能对整个应用系统起作用。

为了真正使所设计的应用系统操作方便，真正符合 WINDOWS 的风格（象 MICROSOFT WORD 等许多软件一样，具有方便的系统工具栏。），笔者根据 VFP 提供的面向对象的类设计技术，设计带有定义了自己功能的工具按钮的工具栏的类，使用事件驱动的原理，使得该工具栏能够对整个应用系统起作用。这样就解决了工具栏既能对整个应用系统起作用，又具有应用系统自己定义的功能的问题。

二、介绍两种生成数据库应用系统独特的工具栏的设计方法

1. 用先进的可视化类设计器创建定制工具栏

首先建立自己的类库 Mystyle. VCX，在 Mystyle. VCX 中以 CommandButton 为基类生成自己的类 ToolButton，设置其属性和方法，特别是 click event，以实现其操作手段的功能；然后在类库 Mystyle. VCX 中又以 ToolBar 为基类生成自己的类 MyToolBar，添加类库 Mystyle. VCX 中的 ToolButton 对象为工具栏中各特定的按钮，设置对象的属性和方法特别是 click event 以完成其功能。最后在事件 init event 中设置属性 Dock 使此工具栏置于屏幕的顶部、左边、右边或底部，抑或初始化时不出现。

设计类 TOOLBUTTON 的 CLICK EVENT 方法是解决自定义功能的关键，其设计思想如下：

（以笔者设计的一个应用系统为例）：

初始化:有关环境参数的设置、变量定义

↓

当前数据库为用户定义的数据库吗?

否 以下工具按钮隐含 →

↓ 是

CASE THIS.NAME='EXIT' 退出当前数据表

CASE THIS.NAME='CLEAR' 清屏、数据表彻底删除

CASE THIS.NAME='CLOSE' 关闭当前数据库及其数据表

↓

当前数据表为用户定义的数据库中的数据表吗?

否 以下工具按钮隐含 →

↓ 是

CASE THIS.NAME='NEW' 添加记录

CASE THIS.NAME='OUT' 数据表输出到文本

CASE THIS.NAME='BROWSE' 数据表浏览

CASE THIS.NAME='IN' 文本信息录入到数据表

↓

当前数据表为空吗? 是 以下工具按钮隐含 →

↓ 否

CASE THIS.NAME='SCREEN' 屏幕可视化编辑

CASE THIS.NAME='FIRST' 记录指针到第一条记录

CASE THIS.NAME='PREV' 记录指针向前一条记录

CASE THIS.NAME='NEXT' 记录指针向后一条记录

CASE THIS.NAME='LAST'记录指针到最后一条记录

CASE THIS.NAME='DELETE' 删除一条记录

CASE THIS.NAME='PRINT' 报表打印

CASE THIS.NAME='LOCATE' 数据表查询

CASE THIS.NAME='RELALOC' 有关联数据表查询

CASETHIS.NAME='ALLDEL'删除数据表中所有记录

↓

应用系统主程序中若有以下几条语句,运行该程序即能使该工具栏有效,这是解决工具栏对整个应用系统有效的关键。

```
…
PUBLIC oToolBar
oToolBar = CreateObject("mytoolbar")
oToolBar.Show
…
```

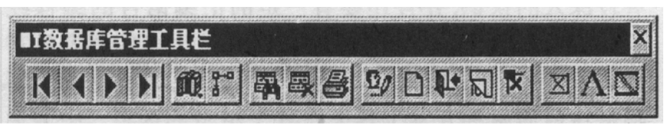
2.通过程序化的手段创建定制工具栏

```
PUBLIC oToolBar
oToolBar = CREATEOBJECT(" MyToolBar")
oToolBar.Show
  …
  DEFINE CLASS ToolButton AS CommandButton
  …
    procedure click
    …
    end PROC
  ENDDEFINE
  DEFINE CLASS MyToolBar AS ToolBar
  …
  ADD object BtnTop AS ToolButton
    WITH Name =" "
      Picture =" "
  ADD object BtnPrev AS ToolButton
    WITH Name =" "
      …
  …
  procedure init
    This.Dock(0)
  endPROC
  end DEFINE
```

其中 Dock(x)中 x 为 0:工具栏停放顶部;1:工具栏停放左边;2:工具栏停放右边;3:工具栏停放底部;-1:不停放工具栏。

三、结束语

使用这两种方法中的任何一种生成的应用系统工具栏,可以停放在应用系统界面的顶部、左边、右边、底部,也可以和其他系统工具栏一样,任意拖动停放在应用系统窗口的任意位置。每一工具按钮均带有状态提示显示其主要功能。下面是笔者设计的某应用系统工具栏的全貌:



(来稿时间:1998 年 8 月)