

董 事 长:周慕昌 社 长:孙毓林
主 编:孙毓林(兼) 副主编:王锡生
编 辑 部:张延淑 庄永祺 肖艳 商波
公关广告部:杨红(主任) 邢卫红 高宁 陈静
主 办:电子工业出版社计算机与微电子
发展研究中心

编辑出版:《软件世界》杂志社
地 址:北京复兴路乙 20 号
通讯地址:北京 162 信箱(100036)
编辑部电话:821 2233 转 3431 或 5048
公关广告部电话:8283915 8212233 转 3445 或 5048
印 刷:电子部科技情报所印刷厂
国内总发行:北京报刊发行局
订 购:全国各地邮局
邮发代号:82-469
刊 号:ISSN 1005-2348/CN11-3394
广告许可证号:京海工商广字 004 号
每期售价:2.00 元 全年定价:24.00 元
出版日期:1995 年 5 月 21 日

《软件世界》1994 年合订本 征订启事

《软件世界》月刊 1994 年合订本现已出版。在此合订本中,介绍了国内外各种软件新技术和各类软件新产品;汇集了软件开发、应用中的经验与体会,以及用机技巧;系统讲解了 UNIX System V 程序设计技术和 FoxPro 的安装与使用技术与经验;反映了软件产业与软件市场的信息与动态;为参加软件水平与资格考试的读者提供了有关辅导材料;为学习专业英语的读者提供了英汉对照软件专业时文选读;配合软件中文化工作,集中介绍了汉字信息处理技术、中文平台技术与标准、中文化软件产品等。此外,合订本中还收进了国家软件著作权登记机关公布的登记软件和自由软件联谊会发布的自由软件,并且还包含了介绍世界最大软件公司——微软(Microsoft)公司最新技术与产品的两期专稿。内容丰富、全面,实用性强,信息含量大,可供从事软件开发、生产、销售人员,计算机用户和软件专业的院校师生阅读、参考。

合订本每本定价 35 元。凡需要者,可向《软件世界》杂志社汇款订购。

1993 年《软件世界》杂志合订本尚有部分剩余,每册 20 元。如需,亦可订购。

联系地址:北京复兴路乙 20 号

通信地址:北京 162 信箱《软件世界》杂志社

邮政编码:100036

电 话:8212233 转 3431

联系人:肖 艳

软件世界

SOFTWARE WORLD

1995 年 第 5 期 (总第 99 期)

目 录

4 产业动态

开发与应用

- 8 Powerbuilder 获取 Sybase 存储过程
返回参数的方法 张艳燕 郭 立
- 10 用 Visual BASIC 3.0 在 Windows For Workgroups
网络上开发并建立 Visual BASIC 应用程序到
EXCEL 的 DDE 连接 高志成
- 12 通用数据库编辑器的设计 陶 波
- 16 数据库 MEMO 型字段绘图工具 GDES 的
设计与应用 叶颖生 李鸿书

实践与经验

- 19 一种手工恢复文件的方法 邓元木
- 19 UCSDOS(3.1)特显功能及其使用 马 良
- 20 Foxpro2.5 索引文件的使用技巧和
应注意的一些问题 王 平
- 22 如何将高分辨率图形在打印机上放大输出 ... 汪前明
- 24 C 语言向 DOS 操作系统传输键盘命令的
一种简便方法 侯万春
- 25 优化 Windows 运行环境的一些体会 ... 崔智 金西

产品大观

- 27 企业事务处理信息系统的粘合剂——
TUXEDO 事务处理管理软件 杨惠民
- 30 写作者的必备工具——电脑写作字典 吴鹏奇
- 32 硬盘的保护神——ADMPLUS 常 如
- 34 UCWIN3.1——希望汉字系统进军 Windows
市场新成果
- 35 应用系统开发工具 PowerHouse 孙大奇
- 36 '95 第六届全国计算机软件交流交易会
获奖优秀软件

软件评测

- 37 看'94 中文 Windows 平台 刘 弋 杜 军

软件新天地

- 39 抓住机遇 迎接挑战——
谈中文信息处理技术及其产品化 张宝元

多媒体创作园地

- [11] 多媒体节目中的声音 李素端

病毒曝光

- [13] 著名电脑病毒档案大公开

软件水平考试

- [14] 关于软件资格与水平考试答读者问 本刊编辑部

技术讲座

- [16] 面向对象语言和 C++ 讲座
第四讲 基于 C++ 语言面向对象的
分析和设计 高 峻 蒋维杜

知识园地

- [51] 英汉对照软件专业外文选读
—— 编程人员的热点 刘 欣

软件市场

- [52] 小荷才露尖尖角 我看黑马电子新技术公司
..... 依 波
[53] 出版软件

技术专题

- [55] 计算机支持协同工作(CSCW)概述
[56] CSCW 的历史回顾和问题焦点
[59] CSCW 工具的基本概念与体系结构
[62] CSCW 环境下的 CAI 软件 一个通用的
电子教室系统模型 程 莉 朱家铨
[64] CSCW 有关术语

MAIN CONTENTS

- Design of General Database Editor(12)
Design and Application of Drawing Tool of Database
MEMO Type Fields(16)
A Method of Manual Restoring Files(19)
Display Functions of UCDS 3.1 and Their Usage(19)
Some Experiences in Maximizing Windows Running
Environment(25)
TUXEDO--A Transaction Processing Management
Software (27)
Writer's Necessary Tools --Computer Writing Dictionaries
(30)
A Survey of '94 Chinese Platforms of Windows(37)
Sounds in Multimedia Titles(41)
Lectures on Object-oriented Languages and C++
Chapter 4 Object-Oriented Analysis and Design Based
on C++ (46)
An Introduction to Computer-Supported Cooperative Work
(55)

KJS35/0405

原
书
缺
页

原
书
缺
页

原
书
缺
页

原
书
缺
页

原
书
缺
页

原
书
缺
页

原
书
缺
页

Powerbuilder 获取 Sybase 存储过程返回参数的方法

() 张艳燕 郭立

Sybase 数据库以其独特的客户/服务器体系结构和优良的多线索技术,被广泛应用于交易系统中。我们已成功地应用 Sybase 数据库和面向对象的数据库开发工具 PowerBuilder 完成了秦皇岛煤碳交易所和重庆商品交易所计算机交易管理系统,利用 Sybase 系统大大提高了系统开发、调试的效率,投入运行后系统性能稳定且很好地满足了交易系统高可靠性、实时性等方面的要求。

交易系统一般可分为前台交易系统和后台管理系统两大组成部分,前台交易系统由主持人子系统和出市代表子系统组成,后台管理系统由交易部管理子系统、结算部管理子系统、交割部子系统、信息部子系统、研发部子系统、总裁查询子系统等部分组成。

PowerBuilder 是面向目标的、专业的客户/服务器开发工具,使开发人员能迅速地建立复杂的图形应用,便于维护和修改,但是速度不高。前台交易系统,特别是出市代表子系统,多个用户同时操作,对响应速度要求很高,为了提高系统响应速度,就要充分利用 Sybase 服务器的潜力,即充分利用 Sybase 的 SQL 语言写成的存储过程。

前台交易系统对如下几张表进行操作:意向表、需求表、资源表及成交表,这些表分别用来存储出市代表的交易意向、需求(买入)、资源(卖出)及成交情况。

在交易过程中,当出市代表每提交一笔交易意向时,交易系统根据价格优先、时间优先的原则与满足条件的多笔已提交在买/卖盘中的意向自动匹配成交,这一过程首先判断是否有满足匹配条件的交易意向,若有则自动匹配成交,将成交情况存入成交表中,并根据买入和卖出将未成交部分存入需求表或资源表中,否则直接根据买入或卖出将提交的意向存入需求表或资源表中。出市代表提交一笔交易意向的操作在交易过程中频繁出现,为了提高系统反应速度,一般用存储过程来完成,并返回成交价和成交数量。类似的情况很多,在此就不一一列举,因此存储过程与 PowerBuilder 之间的参数传递问题就变得非常重要。

返回参数的几种方法

Sybase 存储过程返回参数有三种方法:

方法一:用 Select 语句返回参数

存储过程:

```
create procedure t1 as
//Select 语句返回两个参数:服务器的时间和日期
select convert(char,getdate(),8),
        convert(char,getdate(),2)
```

Power Builder:

```
string time,date;
//定义过程
DECLARE t1 PROCEDURE FOR t1;
EXECUTE t1; //执行过程
//取回存储过程用 Select 语句返回的两个参数
FETCH t1 INTO :time, :date;
CLOSE t1;
```

方法二:用 return 语句返回参数

下面从存储过程用 return 语句返回给 PowerBuilder 一个参数,并用 Select 语句返回一组参数。

存储过程:

```
create procedure t2 as
select sequence_no,quantity from intention
return 9//返回参数9
```

PowerBuilder:

```
int rc,myresult
//定义过程
DECLARE t2 PROCEDURE FOR @rc=&.t2;
EXECUTE t2; //执行过程
//取值到再无满足条件的记录
Do While SqlCa.SqlCode=0
    FETCH t2 INTO :rc, :myresult;
loop
//下面取出用 return 语句返回的参数9
FETCH t2 INTO :rc;
CLOSE t2;
```

方法三:用 output 参数返回参数

下面从 PowerBuilder 传给存储过程一个参数 name,存储过程用 output 参数返回给 PowerBuilder 一个参数 outvar

存储过程:

```

//创建过程,变量 outvar 后加参数 output 表示
outvar 是返回变量
create procedure t1 @name char(3), @outvar int
output as
begin
begin transaction
insert into.....
//执行过程出错,返回参数 outvar=-100
if @@error!=0 or @@rowcount=0
begin
select @outvar=-100
rollback transaction
return
end
//执行过程正确,返回参数 outvar=100
select @outvar=100
commit transaction
return
end

```

PowerBuilder

```

long rc
//定义过程,变量后加 output 表示返回参数
DECLARE t11 PROCEDURE FOR t1
    @name="asd", @outvar=0 output;
EXECUTE t11; //执行过程
//取到返回值100或-100
FETCH t11 INTO :rc;
CLOSE t11;

```

实 例 分 析

下面以交易系统中交易员提交一笔卖出意向,交易系统根据匹配原则自动匹配成交或者把新提交卖出意向存入可供表,说明 Sybase 存储过程如何返回参数给 Powerbuilder。

提交意向时,PowerBuilder 传给存储过程 f __intsell11 两个参数 seller 和 seque sell,存储过程用 output 参数返回给 PowerBuilder 两个参数 outvar 和 outvarp。

存储过程:

```

create procedure f __intsell11 @seque sell int, @seller
char(3),
    @outvar int output, @outvarp int output as
begin
begin transaction
//定义变量:成交量和成交价
declare @strikequan int, @strikeprice int, .....
//首先判断需求表中是否有符合匹配条件的记

```

录

if 有符合匹配条件的记录

begin

//变量赋值

```
select @strikequan=10, @strikeprice=
8
```

插入成交表

insert into strike.....

```
//插入过程出错,返回参数 outvar=-1,
outvarp=-1
```

if @@error!=0 or @@rowcount=0

begin

select @outvar=-1, @outvarp=-1

rollback transaction

return

end

end

else //无符合匹配条件的记录

begin

//变量赋值

select @strikequan=0, @strikeprice=0

//插入可供表

insert into offering.....

```
//插入过程出错,返回参数 outvar=-1,
outvarp=-1
```

if @@error!=0 or @@rowcount=0

begin

select @outvar=-1, @outvarp=-1

rollback transaction

return

end

end

//过程执行正确,返回成交量和成交价

```
select @outvar = @strikequan, @outvarp = @s-
trikeprice
```

commit transaction

return

end

PowerBuilder:

long rc,rcp

//定义过程,变量后加 output 表示返回参数

DECLARE procint1 PROCEDURE FOR f __intsell11

@seque sell= :seque no, @seller=:depnum;

@outvar=0 output, @outvarp=0 output;

EXECUTE procint1; //执行过程

//得到返回的成交量和成交价

FETCH procint1 INTO :rc, :rcp

CLOSE procint1;

用 VISUAL BASIC3.0 在 WINDOWS FOR WORKGROUPS 网络上开发并建立 VISUAL BASIC 应用程序 到 EXCEL 的 DDE 连接

河北大学 高志成

如可用 EXCEL 卓越的数据分析、数据管理和图表等功能来处理网络上其它计算机采集的实时数据,是开发网络上实时监控程序所要解决的问题之一。

下面谈谈在 WINDOWS FOR WORKGROUPS 3.1 (以下简称 WFW)网络上用 VISUAL BASIC3.0(以下简称 VB)及动态数据交换(Dynamic Data Exchange 简称 DDE)解决这一问题的一些技巧和方法。

1. 假设 WFW 网上某一计算机名为 COMAPUTER1,在此机上用 VB 建立一个基于 WINDOWS 的应用程序 VBSERVER.EXE 作为 DDE 服务器(SERVER)。步骤如下:

(1)启动 VB,若 VB 正在运行,从 FILE 菜单,选择 New Project 菜单项,窗体 Form1 被建立。

(2)将窗体 Form1 的 LinkTopic 属性改为 VBToVB;LinkMode 属性改为 1 Source。

(3)在窗体 Form1 上,加控制数组 Text1(0)~Text1(5)。方法是用左键双击工具箱中的正文框工具,将正文框 Text1 的 Index 属性改为 0,再将其拷贝,粘贴 5 次,排好即可。

(4)在窗体 Form1 上加定时器 Timer1。

(5)从 FILE 菜单,选择 New Module 菜单项,Module1 被建立。

(6)将下代码加到 Module1(.bas)模块的 general declarations 声明段下,下面的语句如有多行,请在程序模块中写成单行。

```
Type NETSHAREINFO
    ShareName As String * 65
    ServerApp As Long 'LPSTR ServerApp
    lpSerTopic As long 'LPSTR lpSerTopic
    lpbPassword1 As Long 'LPBYTE lpbPassword1
    cbPassword1 As Long 'DWORD cbPassword1;
    dwPermissions1 As Long 'DWORD dwPermissions1;
    lpbPassword2 As Long 'LPBYTE lpbPassword2;
    cbPassword2 As Long 'DWORD cbPassword2;
    dwPermissions2 As Long 'DWORD dwPermissions2;
    lpSeltem As Long 'LPSTR lpSeltem;
    AddItems As long 'LONG cAddItems;
    lpNDdeShareItemInfo As Long
End Type
```

```
Declare Function NtdeShareAdd Lib "NDDEAPI.DLL" (Server
    As Any, ByVal Level AS Integer, ChareInfo As
    NETSHAREINFO, ByVal Size As Long) As Integer.
Declare Function lstrcpy Lib "KERNEL" (szDest As Any,
    szSource As Any) As Long
```

(7)将下列代码加到窗体 Form1 的 Form_Load 事件过程

```
Sub Form_Load()
    Dim ShareName As String
    Dim ServerName As String
    Dim SerTopic As String
    Dim SerItem As String
    Dim ReadOnlyPassword As String
    Dim FullAccessPassword As String
    Dim ShareInfo As NETSHAREINFO
    Dim ShareInfoSize As Long
    Dim Result As Integer
    ShareName = "SeDDESource $" + Chr$(0)
    ServerName = "VBSERVER" + Chr$(0)
    SerTopic = "VBTOVB" + Chr$(0)
    SerItem = Chr$(0)
    ReadOnlyPassword = Chr$(0)
    FullAccessPassword = Chr$(0)
    ShareInfo.ShareName = ShareName
    ShareInfo.ServerApp = lstrcpy (ByVal ServerName,
    ByVal ServerName)
    ShareInfo.lpSerTopic = lstrcpy (ByVal SerTopic, ByVal
    SerTopic)
    ShareInfo.lpSeltem = lstrcpy (ByVal SerItem, ByVal
    SerItem)
    ShareInfo.cbPassword1 = 0
    ShareInfo.lpbPassword1 = lstrcpy (ByVal
    ReadOnlyPassword, ByVal ReadOnlyPassword)
    ShareInfo.dwPermissions1 = &H1 Or &H2 Or &H4 Or
    &H8 Or &H10
    ShareInfo.cbPassword2 = 0
    ShareInfo.lpbPassword2 = lstrcpy (ByVal
    FullAccessPassword, ByVal FullAccessPassword)
    ShareInfo.dwPermissions2 = &H1 Or &H2 Or &H4 Or
    &H8 Or &H10
    ShareInfo.lpNDdeShareItemInfo = 30
    Result = NDdeShareAdd (ByVal 0%, 2, ShareInfo, Len
    (ShareInfo))
```

```

timer1.Interval=1000
timer1.Enabled=True
End Sub

```

(8)将下列代码加到 Timer1_Timer 事件过程

```

Sub Timer1_Timer()
    Dim k
    For k=0 To 5
        Randomize Timer
        Text1(k).Text=Format$(Rnd*100,"0")
    Next
End Sub

```

(9)从 FILE 菜单,选择 Make EXE File...菜单项,在 Make EXE File 对话框的 File Name 正文框内键入 C:\WINDOWS\VBSERVER.EXE,则服务器应用程序 VBSERVER.EXE 被建立且存在 C:\WINDOWS 下。

(10)从 WFW 程序管理器 Program Manager 的 FILE 菜单,选择 Run...菜单项,运行 VBSERVER.EXE,则在六个正文框内每秒都显示一个随机数。

2. 在 WFW 网络的另一计算机(设机名为 COMPUTER2)上用 VB 建立一个基于 WINDOWS 的应用程序 CLANDSE.EXE 即作为计算机 COMPUTER1 上程序 VBSERVER.EXE 的客户(Client),同时又是本机 COMPUTER2 上 EXCEL 的服务器

首先启动 EXCEL(设路径为 C:\MSEXCEL),将一新工作表以 VBCLIENT.XLS 存在 C:\MSEXCEL 下。

创建 CLANDSE.EXE 的步骤如下:

(1)启动 VB,若 VB 正在运行,从 FILE 菜单,选择 New Project 菜单项,窗体 Form1 被建立。

(2)将窗体 Form1 的 LinkTopic 属性改为 VBToExcel;LinkMode 属性改为 1_Source。

(3)在窗体 Form1 上,加控制数组 Text1(0)~Text1(5)方法同一的第3步。

(4)将下列代码加到 Module1(.bas)模块的 general declarations 声明段下:

```

Declare Function FindWindow% Lib "user"
(ByVal lpCalssName As Any, ByVal lpCaption As Any)

```

(5)将下列代码加到窗体 Form1 的 Form_load 事件过程:

```

Sub Form_Load()
    Dim lpClassName As String
    Dim lpCaption As String
    Dim SerComputer As String

```

```

Dim SerTopic As String
Dim SerItem As String
Dim Index
lpClassName="XLMain"
lpCaption="Microsoft Excel"
If FindWindow%(lpClassName, lpCaption)
    =0 Then
    exl%=Shell("C:\MSEXCEL5\
EXCEL C:\MSEXCEL5\VBCLIENT
XLS",1)
    exl%=DoEvents()
End If
SerComputer="\\COMPUTER1"
SerTopic=seDDSource $"
For Index=0 To 5
    SerItem="Text1(" + Format$(Index,"
0")+")"
    Text1(Index).LinkMode=0
    Text1(Index).LinkTopic =
SerComputer+"\" + NDDE $"r" +
SerTopic
    Text1(Index).LinkItem=SerItem
    Text1(Index).LinkMode=1
Next Index
End Sub

```

(5)从 RUN 菜单,选择 Start 菜单项,运行此程序,则在六个正文框内每秒都将同步显示计算机 COMPUTER1 上服务器 VBSERVER.EXE 对应正文框传来的随机数,若 EXCEL 没有运行,此程序将自动启动 EXCEL 和 VBCLIENT.XLS。

(7)从 FILE 菜单,选择 Make EXE File...菜单项,创建 EXE 文件 CLANDSE.EXE。

三、在 EXCEL 的工作表 VBCLIENT.XLS 内,选择单元格 A1, A2...A6,分别键入 =CLANDSE | VBTOEXCEL! 'TEXT1(0)', =CLANDSE | VBTOEXCEL! 'TEXT1(1)'..., =CLANDSE | VBTOEXCEL! 'TEXT1(5)',则在六个单元格 A1. A2...A6 内每秒都将同步显示本机 COMPUTER2 上 CLANDSE.EXE 对应正文框传来的随机数,当然是计算机 COMPUTER1 上服务器 VBSERVER.EXE 对应正文框传来的随机数。

上述仅以六个元素的控制数组及产生的随机数为例,在实际问题中,可根据实时的数量来定义数组的大小,当数据传到 EXCEL 后,就可用其强大的功能来处理这些数据。

通用数据库编辑器的设计

湖南省邮电管理局信息中心 陶 波

在管理信息系统的开发中,最基本的操作是对数据库的操作。目前国内应用的数据库中仍以DBF类的数据库为主流产品。单位自行开发的应用系统中,大多数只能在屏幕上显示一条要操作的记录,上下记录、左右字段的操作很不直观不方便。在FoxBase中虽然提供BROW命令解决这一问题,但由于其窗口大小固定、色彩单一、功能简单,因此人们在实际很少使用它。

笔者在设计某管理信息系统中,模仿文本编辑器的某些操作方式,设计了一个较为完善的通用数据库编辑器,实现对数据库各种操作及一些特殊的接口。

一、设计的基本思想

1. 过程框图

请看右图

2. 说明

1) 获取库结构信息

库结构信息应包括被编辑数据库的记录数,字段数,每个字段的字段名、类型、长度、小数位。若字段名不是汉字,可另设一中英文字段名对照库,从中获得相应的汉字字段名;也可用其他方法获得。

2) 画出编辑窗口

根据给出的参数(左上角、右下角的行列号)画一个窗口。可以是表格方框(列号差须为偶数);也可以是凸凹式的立体方框。

3) 设置记录号和字段号

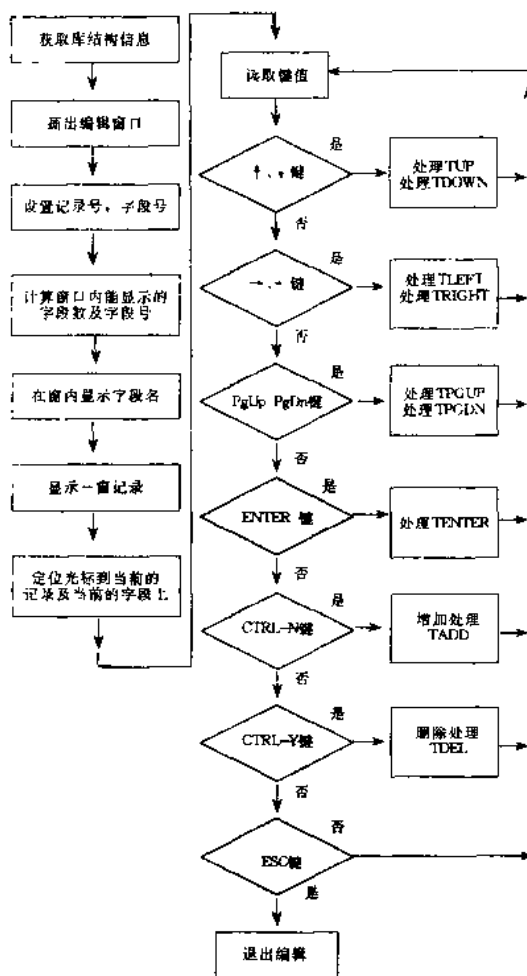
本次是初始设置,应设为首记录、第一个字段,作为当前值。

4) 计算窗内能显示的字段数及字段号

从设置的字段号起始横向向后累计字段的长度及

字段的个数,直到字段的长度和刚好小于窗口的宽度。

5) 在窗内显示字段名



根据4)中计算出的字段号和字段数,在窗内第一

行显示 1) 中得出的字段名。

6) 显示一窗记录

根据 4) 中计算出的字段号和字段数, 在窗内第二行开始到窗底, 显示一窗口的记录的对应的字段值。

7) 定位光标

将光标定位在 3) 中设置的记录号及字段号上, 此时的位置就是当前的行列。

8) 进入读取键值的循环体

根据读出的键值, 做不同的操作:

- ↑ : 光标移到上一个记录
- ↓ : 光标移到下一个记录
- : 光标移到同一个记录的右边的字段
- ← : 光标移到同一个记录的左边的字段

PgUp PgDn: 记录向上或向下翻页

ENTER: 修改光标所在行列的记录的字段的值

CTRL+N: 增加一个新记录

CTRL+Y: 删除光标所在的当前记录

ESC : 退出数据库编辑

9) 循环体中的各种处理

●处理 TUP:

① 若当前记录是首记录, 则不操作。

② 若当前记录位于窗口中间, 则当前记录去掉光标, 以上一个记录为新当前记录, 将光标移到新的当前记录上。

③ 若光标位于窗口第二行且这条记录非首记录, 则向前移动一个记录作为当前记录, 重新显示一窗记录。这样产生了向下滚屏的效果。

处理 TDOWN 与此类似, 处理 TPGUP、处理 TPGDN 亦可参考此设计。

●处理 TRIGHT:

① 若当前字段是最后一个字段, 则不操作。

② 若当前字段位于窗口中间, 则当前字段去掉光标, 以下一个字段为新的当前字段, 将光标移到新的字段记录上。

③ 若光标位于窗口内最右边的字段上且该字段不是最后一个字段, 则以其后面的一个字段作为当前字段, 再向前推算出窗口最左边的字段号; 重新在窗口内显示字段名, 重新显示一窗记录。这样就产生了向左滚屏的效果。

处理 TLEFT 与此类似。

●处理 TENTER:

根据 1) 中获取的库机构信息, 按其对应的格式读入所在字段的新的值。

●处理 TADD:

若在窗口内当场增加一个记录, 则从第一个字段

开始输入字段值(其过程与处理 TENTER 一样), 光标逐步向右移动(其过程与处理 TRIGHT 一样)

●处理 TDEL:

删除当前记录后, 要注意新的当前光标的定位和新的当前记录的设定。

二、例 程

例程是用编译型数据库语言 CLIPPER 所写。限于篇幅, 例程仅供读者对设计思想做深入理解, 实际设计中还须考虑诸多次要因素。

例程说明:

●主程序: MAIN.PRG, 其中调用数据库编辑器函数。

●数据库编辑器函数: TEDITD()

其输入参数:

row1, col1: 窗口左上角的行列值

row2, col2: 窗口右下角的行列值

clr1, clr2: 窗口的边框色和内底色

其主要变量:

ltrow, ltcol: 窗口内左上角的行列

rlrow, rrcol: 窗口内右下角的行列

cr1, cr2: 窗口边框颜色和内底色

recd: 当前的记录号

rowd, cold: 光标所在的当前行, 当前列

fdnum: 库字段总数

fdnumw: 要显示的字段个数

fdnuml: 屏中左边字段的逻辑序号

fdnumr: 屏中右边字段的逻辑序号

fdnumd: 当前光标所在字段的逻辑序号

1 ≤ fdnuml ≤ fdnumd ≤ fdnumr

≤ fdnumw ≤ fdnum

fd[n]: 字段的物理序号

其中 n: 逻辑序号 from 1 to fdnumw

dbsa[m][4]: 库结构数组

其中 [1]: 字段名 [2]: 字段类型

[3]: 字段长度 [4]: 小数位长度

m: 字段的物理序号

dbsa[m][2] 数组

其中 [1]: 字段汉字名 [2]: 大者

(字段长度, 字段汉字名长度)

m: 字段的物理序号

newfd[m]: 字段序号的字符型(三位)

其中 m: 字段的物理序号, m = fd[n]

```
fdnuml:=1
fdnumr:=1
fdnumd:=1
fdnumw:=fdnum
j:=1
do while j< fdnumw
  fdli:=1
```

```

j=j+1
enddo
fdum1=j
if fdum1>fdnumw
    fdnum1=fl
    fdnumr=fr
    return
endif
fdl=1
j=1
j=fdnumi
do while j<=fdnumw
    fdl=fdl+dbaac[fd[j]][2]
    if (fdl+i)>(rlcol-ltcol+1)
        j=j+1
        exit
    elseif j=fdnumw
        exit
    endif
    j=j+1
enddo
fdnumr=j
if fdnum1=fdnumr
    fdnumd=fdnuml
elseif fdnumd<=fdnumr
    fdumdd+=
endif
endif
setcolor(cr2)
@ltrow,ltcol clear to rtrow,rtcol
fdl=ltcol+1
j=fdnuml
do while j<=fdnumr
    @ltrow,fdl say dbaac[fd[j]][1:]
    fdl=fdl+dbaac[fd[j]][2]+1
    j=j+1
enddo
return

function showrec(n)//显示一窗记录
rec=recno()
if n=0 //从上至下
    takip(ltrow+1)-rowd
    k=ltrow+1
    do while k<=rtrow
        col=ltcol
        j=fdnuml
        do while j<=fdnumr
            ffd=fieldget(fd[j])
            @k,col+1 say ffd
            col=col+dbaac[fd[j]][2]+1
            j=j+1
        enddo
        k=k+1
    if takip(1)<>1
        if k=rtrow
            @k,ltcol say space(rlcol-ltcol)
        elseif k<rtrow
            @k,ltcol clear to rtrow,rtcol
        endif
    exit
endif
enddo
elseif n=1 //从下至上

```



```

tskip(rrow - rowd)
k = rrow
do while k >= lrow + 1
col = lcol
j = fdnuml
do while j <= fdnumr
ffd = fieldget(fld[j])
@ k.col + j say ffd
col = col + dbase(fld[j]*2) + 1
j = j + 1
enddo
tskip(-1)
k = k - 1
enddo
endif
go rec
return

```

```

function showrec2(cir, nno) // 将所
cold = jscold() // 在记录字段上色
ffd = fieldget(fld[fdnumd])
@ rowd . cold say ffd
return

```

```

function showrec3() // 去掉上次光
cold = jscold() // 标的颜色
ffd = fieldget(fld[fdnumd])
@ rowd . cold say ffd
return

```

```

function jscold() // 计算当前字段的起始列
col = lcol
if fdnumd <> 1
for j = fdnuml to fdnumd - 1
col = col + dbase(fld[j]*2) + 1
next
endif
return col + 1

```

```

function tskip(n) // 上下移动记录
if n = 0
return 0
elseif n > 0
for i = 1 to n
skip 1
if eof()
return i - 1
endif
next
elseif n < 0
i = 1
for i = 1 to n

```

```

skip - 1
if eof()
return i - 1
endif
next
endif
return i - 1

```

```

function tup()
rec = recno()
if tskip(-1) <> 1
? chr(7)
go rec
return
endif
go rec
if rowd = lrow + 1
tskip(-1)
showrec1(0)
showrec2(3)
else
showrec3(0)
tskip(-1)
rowd = rowd - 1
showrec2(3)
endif
return

```

```

function tright()
if fdnumd = fdnumw
? chr(7)
return
elseif fdnumd = fdnumr
showfdn(1)
showrec1(0)
showrec2(3)
else
showrec3(0)
fdnumd = fdnumd + 1
showrec2(3)
endif
return

```

```

function tadd()
append blank
rec = recno()
recnum = recnum + 1
rowd = lrow + 1
return

```

```

function taddins()

```

```

if fdnuml != 1
fdnuml = 1
fdnumd = fdnuml
showfdn(0)
showrec1(0)
endif
fdnumd = fdnuml
do while fdnumd <= fdnumw
if taster() = K_ESC
exit
endif
if fdnumd = fdnumw
if fdnuml <> 1
fdnuml = 1
fdnumd = fdnuml
showfdn(0)
showrec1(0)
else
showrec3(0)
fdnumd = fdnuml
endif
showrec2(3)
exit
elseif fdnumd = fdnumr
showfdn(1)
showrec1(0)
else
showrec3(0)
fdnumd = fdnumd + 1
endif
enddo
return

```

```

function tgetpara() // 获取数据库结构信息
recnum = reccount()
fdnum = fcount()
dbase = dbstruct()
for i = 1 to fdnum
dbase[i][1] = chuniname() // 取得中文字段名
next
for i = 1 to fdnum
if len(trim(dbase[i][1])) > dbase[i][3]
dbase[i][1] = len(trim(dbase[i][1]))
else
dbase[i][2] = dbase[i][9]
endif
next
for i = 1 to fdnum
newid[i] = atr(i, 3)
next
return

```

三、功能扩展

仅仅局限于前面所述的功能,笔者认为仍有许多不足。读者理解了设计思想之后,不难扩展出下列功能:

1. 在编辑过程中改变窗口的大小和颜色
2. 对每种操作赋予一定的操作权限
3. 对有关操作给出输出接口

4. 可选择需要显示的字段
5. 可定义能修改的或不能修改的字段
6. 根据索引条件显示记录
7. 修改库结构
8. 其它扩展功能

在开发应用系统中,调用数据库编辑器,可以将数据库的应用提高到一个新的层次上,使我们的管理信息系统使用更为方便灵活。笔者愿与有兴趣的读者进行交流。