

4

Разработване на приложение LocateLookupExample

В настоящето упражнение ще бъде представено разработването на приложение за работа с бази от данни в което ще се използват методите за търсене на информация в набори от данни - Locate и Lookup. Примерна сесия на работа с приложението **LocateLookupExample** е показана на Фиг. 1.

Locate and Lookup Example

Search Method: ☒ Locate ☐ Lookup ☐ Manual

Locate Options: ☐ Partial Key ☐ Case Insensitive

EmpNo: First Name: Last Name:

☒ Restrict ☒ Add First Name and Last Name to Log File

Result: EmpNo: 28, FirstName: Ann, LastName: Bennet, PhoneExt: 5, HireDate: 1.2.1991 г., Salary: 34 483 лв.

Log:

FirstName	LastName
Leslie	Johnson
Luke	Leung
Bill	Parker
Scott	Johnson
Ann	Bennet

Save Load Delete Clear

EmpNo	FirstName	LastName	PhoneExt	HireDate	Salary
12	Terri	Lee	256	1.5.1990 г.	45332
14	Stewart	Hall	227	4.6.1990 г.	34482.625
15	Katherine	Young	231	14.6.1990 г.	24400
20	Chris	Papadopoulos	887	1.1.1990 г.	25050
24	Pete	Fisher	888	12.9.1990 г.	23040
28	Ann	Bennet	5	1.2.1991 г.	34482.8
29	Roger	De Souza	288	18.2.1991 г.	25500
34	Janet	Baldwin	2	21.3.1991 г.	23300
36	Roger	Reeves	6	25.4.1991 г.	33620
37	Willie	Stansbury	7	25.4.1991 г.	39224
44	Leslie	Phong	216	3.6.1991 г.	40350

Record Count = 42 Employee: Ann Bennet - Salary: 34 483 лв. Time: 0 ms

Фигура 1: Примерна сесия на работа с приложението **LocateLookupExample**.

Приложението **LocateLookupExample** ще показва данни от таблица "employee", които ще са съхранени във външен xml файл.

Приложението ще визуализира информация за броя на записите в таблицата, информация за текущо избрания служител (employee) и времето за което е било изпълнено последното търсене (в милисекунди).

Избора на метод за търсене се осъществява в секцията [Search Method], чрез избор на едноименния радио бутон за целта, съответно: Locate (използва се метода Locate), Lookup (използва се метода Lookup) и Manual (използва се последователно търсене чрез сравнение).

Настройването на опциите при търсене чрез метода Locate, се осъществява в секцията [Locate Options], като са възможни четири варианта за полетата с отметки: двете полета изключени, двете полета включени и включено по едно от полетата. Опцията [Partial Key] дава възможност за търсене дори при частично съвпадение на критерия. Опцията [Case Insensitive] дава възможност при търсене да се различават малки и главни букви.

Секцията за въвеждане на стойности за търсене включва три раздела: първи раздел в който се намира текстовото поле означено с етикет [EmpNo] и поле с отметка [Restrict]. При включена опция на полето с отметка потребителя може да въвежда само цифрите от '0' до '9'. Втория раздел съдържа текстовите полета означени съответно с етикетите [First Name] и [Last Name]. При включена опция на полето с отметка [Add First Name and Last Name to Log File], програмата добавя въведения критерий за търсене в специален списък (секцията [Log]). Третия раздел включва бутона [Search], при натискането на който се извиква съответния метод за търсене указан в секцията [Search Method].

Когато потребителя е съхранил списъка с опциите за търсене, т.е. натиснал е бутона [Save] при непразен списък, на по-късен етап той може да зареди по-рано съхранения списък чрез натискане на бутона [Load]. При необходимост от изтриване на конкретен елемент от списъка, потребителя може да използва за целта бутона [Delete], а при необходимост от изтриване на всички елементи в списъка, потребителя може да натисне бутона [Clear].

Секцията [Result] включва текстов етикет, в който се визуализира подробна информация (т.е. конкатенирани стойностите на всички полета от таблицата), за намерения в момента запис. Без значение кой от методите за търсене се използва в момента (Locate, Lookup или Manual), при намерен запис отговарящ на критерия за търсене, информацията се визуализира в етикета.

1. СЪЗДАВАНЕ НА ПРИЛОЖЕНИЕТО LocateLookupExample

A. Стартирайте RAD Studio и изпълнете следните стъпки:

Стъпка 1: От главното меню изберете **File ► New ► Windows VCL Application – Delphi (или C++ Builder)** в зависимост от езика, който искате да използвате.

Стъпка 2: От главното меню изберете **File ► Save All...** Задайте име на модула, който ще съдържа кода с имплементация на главната форма, което да бъде: **MainUnit.pas** (за Delphi) или **MainUnit.cpp** (за C++ Builder), а проектния файл запишете като: **LocateLookupExp.dproj** (за Delphi) или **LocateLookupExp.cbproj** (за C++ Builder).

Стъпка 3: Главната форма на приложението трябва да има настройки и компоненти, каквито са показани по-долу:

```
object FormMain: TFormMain
  Caption = 'Locate and Lookup Example'
  ClientHeight = 451
  ClientWidth = 604
  Constraints.MinHeight = 490
  Constraints.MinWidth = 620
  Position = poScreenCenter
  OnCreate = FormCreate
  OnDestroy = FormDestroy

  object TableEmployees: TClientDataSet
    FileName = 'Employee.xml'

    object TableEmployeesEmpNo: TIntegerField
      FieldName = 'EmpNo'
    end
    object TableEmployeesFirstName: TWideStringField
      FieldName = 'FirstName'
    end
    object TableEmployeesLastName: TWideStringField
      FieldName = 'LastName'
    end
    object TableEmployeesPhoneExt: TWideStringField
      FieldName = 'PhoneExt'
    end
    object TableEmployeesHireDate: TDateTimeField
      FieldName = 'HireDate'
    end
    object TableEmployeesSalary: TFloatField
      FieldName = 'Salary'
    end
    AfterScroll = TableEmployeesAfterScroll
  end //object TableEmployees: TADOTable
```

```
object DataSourceEmployees: TDataSource
    DataSet = TableEmployees
end

object ClientDataSetLog: TClientDataSet
    FieldDefs = <
        item Name = 'FirstName' DataType = ftString Size = 15 end
        item Name = 'LastName' DataType = ftString Size = 20 end>
    StoreDefs = True
    //call Create DataSet from ClientDataSetLog's context menu
    object ClientDataSetLogFirstName: TStringField
        FieldName = 'FirstName'
    end
    object ClientDataSetLogLastName: TStringField
        FieldName = 'LastName'
    end
    Active = True
end //object ClientDataSetLog: TClientDataSet

object DataSourceLog: TDataSource
    DataSet = ClientDataSetLog
end

object RadioGroupSearchMethod: TRadioGroup
    Left = 8
    Top = 8
    Width = 201
    Height = 49
    Caption = ' Search Method '
    Columns = 3
    Items.Strings = (
        'Locate'
        'Lookup'
        'Manual')
    ItemIndex = 0
    OnClick = RadioGroupSearchMethodClick
end

object GroupBoxLocateOptions: TGroupBox
    Left = 216
    Top = 8
    Width = 193
    Height = 49
    Caption = ' Locate Options '

    object CheckBoxPartialKey: TCheckBox
        Left = 9
        Top = 21
        Width = 73
        Height = 17
        Caption = 'Partial Key'
    end
end
```

```
object CheckBoxCaseInsensitive: TCheckBox
    Left = 87
    Top = 21
    Width = 98
    Height = 17
    Caption = 'Case Insensitive'
end
end //object GroupBoxLocateOptions: TGroupBox

object GroupBoxSearch: TGroupBox
    Left = 8
    Top = 60
    Width = 401
    Height = 73

    object LabelEmpNo: TLabel
        Left = 9
        Top = 10
        Width = 33
        Height = 13
        Caption = 'EmpNo'
    end

    object EditEmpNo: TEdit
        Left = 8
        Top = 26
        Width = 61
        Height = 21
        MaxLength = 8
        NumbersOnly = True
        OnChange = EditEmpNoChange
        OnEnter = EditEmpNoEnter
    end

    object CheckBoxRestrict: TCheckBox
        Left = 8
        Top = 50
        Width = 60
        Height = 17
        Caption = 'Restrict'
        Checked = True
        OnClick = CheckBoxRestrictClick
    end

    object Bevel1: TBevel
        Left = 76
        Top = 6
        Width = 5
        Height = 65
        Shape = bsLeftLine
    end
end
```

```
object LabelFirstName: TLabel
    Left = 86
    Top = 10
    Width = 55
    Height = 13
    Caption = 'First Name:'
end

object LabelLastName: TLabel
    Left = 198
    Top = 10
    Width = 54
    Height = 13
    Caption = 'Last Name:'
end

object EditFirstName: TEdit
    Left = 85
    Top = 26
    Width = 105
    Height = 21
    MaxLength = 15
    OnChange = EditEmpNoChange //same as EditEmpNo OnEnter event
    OnEnter = EditFirstNameEnter
end

object EditLastName: TEdit
    Left = 197
    Top = 26
    Width = 105
    Height = 21
    MaxLength = 20
    OnChange = EditEmpNoChange //same as EditEmpNo OnEnter event
    OnEnter = EditFirstNameEnter //same as EditFirstName OnEnter event
end

object CheckBoxAddToLog: TCheckBox
    Left = 85
    Top = 50
    Width = 221
    Height = 17
    Caption = 'Add First Name and Last Name to Log File'
    Checked = True
end

object Bevel2: TBevel
    Left = 309
    Top = 6
    Width = 5
    Height = 65
    Shape = bsLeftLine
end
```

```
object BitBtnSearch: TBitBtn
    Left = 317
    Top = 14
    Width = 75
    Height = 49
    Caption = 'Search'
    Enabled = False
    OnClick = BitBtnSearchClick
    Glyph = {choice image ...}
    NumGlyphs = 2
end
end //object GroupBoxSearch: TGroupBox

object GroupBoxResult: TGroupBox
    Left = 8
    Top = 137
    Width = 401
    Height = 53
    Caption = ' Result '

    object LabelResult: TLabel
        AutoSize = False
        Left = 8
        Top = 15
        Width = 385
        Height = 31
        Caption = ' Info:'
        Color = 16119285
        Transparent = False
        Layout = tlCenter
        WordWrap = True
        Transparent = False
    end
end //object GroupBoxResult: TGroupBox

object GroupBoxLog: TGroupBox
    Left = 416
    Top = 8
    Width = 180
    Height = 182
    Anchors = [akLeft, akTop, akRight]
    Caption = ' Log '

    object SpeedButtonLogSave: TSpeedButton
        Left = 8
        Top = 151
        Width = 41
        Height = 25
        Caption = 'Save'
        Flat = True
        OnClick = SpeedButtonLogSaveClick
    end
end
```

```
object SpeedButtonLogLoad: TSpeedButton
    Left = 49
    Top = 151
    Width = 41
    Height = 25
    Caption = 'Load'
    Flat = True
    OnClick = SpeedButtonLogLoadClick
end

object SpeedButtonDeleteLogItem: TSpeedButton
    Left = 90
    Top = 151
    Width = 41
    Height = 25
    Caption = 'Delete'
    Flat = True
    OnClick = SpeedButtonDeleteLogItemClick
end

object SpeedButtonLogClear: TSpeedButton
    Left = 131
    Top = 151
    Width = 41
    Height = 25
    Caption = 'Clear'
    Flat = True
    OnClick = SpeedButtonLogClearClick
end

object DBGridLog: TDBGrid
    Left = 8
    Top = 16
    Width = 164
    Height = 131
    Anchors = [akLeft, akTop, akRight]
    DataSource = DataSourceLog
    Options = [dgTitles, dgIndicator, dgColLines, dgRowLines, dgTabs,
        dgRowSelect, dgConfirmDelete, dgCancelOnExit]
    OnDblClick = DBGridLogDblClick
    Columns = <
        item FieldName = 'FirstName' Width = 62 Visible = True end
        item FieldName = 'LastName' Width = 62 Visible = True end>
end
end //object GroupBoxLog: TGroupBox

object DBGridEmployees: TDBGrid
    Left = 0
    Top = 200
    Width = 587
    Height = 221
    Anchors = [akLeft, akTop, akRight, akBottom]
    DataSource = DataSourceEmployees
```



```

Options = [dgTitles, dgIndicator, dgColumnResize, dgColLines,
           dgRowLines, dgTabs, dgRowSelect,
           dgAlwaysShowSelection, dgCancelOnExit]
Columns = < item FieldName = 'EmpNo' Visible = True end
           item FieldName = 'FirstName' Visible = True end
           item FieldName = 'LastName' Visible = True end
           item FieldName = 'PhoneExt' Visible = True end
           item FieldName = 'HireDate' Visible = True end
           item FieldName = 'Salary' Width = 81 Visible = True end>
end

object StatusBar: TStatusBar
  Height = 25
  Panels = <
    item Text = ' Record Count = ' Width = 140 end
    item Text = ' Employee' Width = 300 end
    item Text = ' Time: 0 ms' Width = 50 end>
end
end //object FormMain: TFormMain

```

След като бъдат добавени и настроени всички компоненти към формата, тя трябва да изглежда подобно на тази показана на Фиг. 2.

Фигура 2: Главната форма на приложението **LocateLookupExample** по време на етапа на проектиране.

Стъпка 4: В класа TFormMain (в декларацията на класа) добавете в частта **private** променлива MyCase от тип Integer и 4 потребителски метода: UpdateStatusLine(), LocateSearch(), LookupSearch() и ManualSearch(), както е показано по-долу:

```
type
  TFormMain = class(TForm)
    DBGridEmployees: TDBGrid;
    //...
    procedure CheckBoxRestrictClick(Sender: TObject);
  private
    { Private declarations }
    MyCase: Integer;
    procedure UpdateStatusLine();
    procedure LocateSearch();
    procedure LookupSearch();
    procedure ManualSearch();
  public
    { Public declarations }
  end;
```

Стъпка 5: В имплементационната част на формата (т.е. в частта **implementation**), добавете реализация на декларираните методи UpdateStatusLine(), LocateSearch(), LookupSearch() и ManualSearch(), както е показано по-долу:

```
procedure TFormMain.UpdateStatusLine;
begin
  StatusBar.Panels.Items[0].Text :=
    ' Record Count := ' + IntToStr(TableEmployees.RecordCount);
  StatusBar.Panels.Items[1].Text := ' Employee: ' +
    TableEmployees.FirstName.AsString + ' ' +
    TableEmployees.LastName.AsString + ' - Salary: ' +
    FloatToStrF(TableEmployeesSalary.AsFloat, ffCurrency, 15, 0);
  if (LabelResult.Caption <> '') then LabelResult.Caption := '';
end;
```

```
procedure TFormMain.LocateSearch;
var
  Result: Boolean;
  Options: TLocateOptions;
begin
  Result := False; Options := [];

  if CheckBoxPartialKey.Checked then Options := [loPartialKey];

  if CheckBoxCaseInsensitive.Checked then
    Options := Options + [loCaseInsensitive];

  case MyCase of
    0: Result :=
      TableEmployees.Locate('EmpNo', EditEmpNo.Text, Options);

    1: Result :=
      TableEmployees.Locate('FirstName;LastName',
        VarArrayOf([EditFirstName.Text, EditLastName.Text]), Options);

    2: Result :=
      TableEmployees.Locate('FirstName', EditFirstName.Text, Options);

    3: Result :=
      TableEmployees.Locate('LastName', EditLastName.Text, Options);
  end;

  if (Result = True) then
  begin
    LabelResult.Caption :=
      'EmpNo: ' + TableEmployeesEmpNo.AsString + ', ' +
      'FirstName: ' + TableEmployeesFirstName.AsString + ', ' +
      'LastName: ' + TableEmployeesLastName.AsString + ', ' +
      'PhoneExt: ' + TableEmployeesPhoneExt.AsString + ', ' + #13#10 +
      'HireDate: ' + DateTimeToStr(TableEmployeesHireDate.AsDateTime) +
      ', Salary: ' +
      FloatToStrF(TableEmployeesSalary.AsFloat, ffCurrency, 15, 0);
    LabelResult.Color := RGB(245, 255, 245);
  end
  else
  begin
    LabelResult.Caption := 'No records found.';
    LabelResult.Color := RGB(255, 245, 245);
  end
end;
```

```
procedure TFormMain.LookupSearch;
var
  V: Variant;
begin
  case MyCase of
    0: V := TableEmployees.Lookup('EmpNo', EditEmpNo.Text,
      'EmpNo;FirstName;LastName;PhoneExt;HireDate;Salary');
    1: V := TableEmployees.Lookup('FirstName;LastName',
      VarArrayOf([EditFirstName.Text, EditLastName.Text]),
      'EmpNo;FirstName;LastName;PhoneExt;HireDate;Salary');
    2: V := TableEmployees.Lookup('FirstName', EditFirstName.Text,
      'EmpNo;FirstName;LastName;PhoneExt;HireDate;Salary');
    3: V := TableEmployees.Lookup('LastName', EditLastName.Text,
      'EmpNo;FirstName;LastName;PhoneExt;HireDate;Salary');
  end;

  if (not VarIsNull(V)) then
  begin
    LabelResult.Caption :=
      'EmpNo: ' + AnsiString(V[0]) + ', ' +
      'FirstName: ' + AnsiString(V[1]) + ', ' +
      'LastName: ' + AnsiString(V[2]) + ', ' +
      'PhoneExt: ' + AnsiString(V[3]) + ', ' + #13#10 +
      'HireDate: ' + DateTimeToStr(V[4]) + ', ' +
      'Salary: ' + FloatToStrF(Double(V[5]), ffCurrency, 15, 0);
    LabelResult.Color := RGB(245, 255, 245);
  end
  else
  begin
    LabelResult.Caption := 'No records found.';
    LabelResult.Color := RGB(255, 245, 245);
  end;
end;
```

```
procedure TFormMain.ManualSearch;
var
    Found: Boolean;
    Bookmark: TBookmark;
begin
    Found := False;
    Bookmark := TableEmployees.GetBookmark();
    TableEmployees.DisableControls();
    TableEmployees.First();
    while (not TableEmployees.Eof) do
    begin
        case MyCase of
            0: if (TableEmployeesEmpNo.AsString = EditEmpNo.Text) then
                Found := True;
            1: if ((TableEmployeesFirstName.AsString = EditFirstName.Text) and
                (TableEmployeesLastName.AsString = EditLastName.Text)) then
                Found := True;
            2: if (TableEmployeesFirstName.AsString = EditFirstName.Text) then
                Found := True;
            3: if (TableEmployeesLastName.AsString = EditLastName.Text) then
                Found := True;
        end;
        if Found then Break; //изход от цикъла
        TableEmployees.Next();
    end; //while...
    if Found then
    begin
        LabelResult.Color := RGB(245, 255, 245);
        LabelResult.Caption :=
            'EmpNo: ' + TableEmployeesEmpNo.AsString + ', ' +
            'FirstName: ' + TableEmployeesFirstName.AsString + ', ' +
            'LastName: ' + TableEmployeesLastName.AsString + ', ' +
            'PhoneExt: ' + TableEmployeesPhoneExt.AsString + ', ' + #13#10 +
            'HireDate: ' + DateTimeToStr(TableEmployeesHireDate.AsDateTime) +
            ', Salary: ' +
            FloatToStrF(TableEmployeesSalary.AsFloat, ffCurrency, 15, 0);
        TableEmployees.FreeBookmark(Bookmark);
        TableEmployees.EnableControls();
    end
    else //Found = false
    begin
        LabelResult.Caption := 'No records found.';
        LabelResult.Color := RGB(255, 245, 245);
        TableEmployees.GotoBookmark(Bookmark);
        TableEmployees.FreeBookmark(Bookmark);
        TableEmployees.EnableControls();
    end;
end;
```

Стъпка 6: Създайте следните манипулатори (обработчици) на събития и въведете сорс-кода в редактора за код (**Code Editor**):

```
procedure TFormMain.FormCreate(Sender: TObject);
begin
    if (TableEmployees.Active) then TableEmployees.Close();
    TableEmployees.FileName :=
        ExtractFilePath(Application.ExeName) + 'Employee.xml';
    if (not TableEmployees.Active) then TableEmployees.Open();
end;
//-----
procedure TFormMain.FormDestroy(Sender: TObject);
begin
    if (TableEmployees.Active) then TableEmployees.Close();
end;
//-----
procedure TFormMain.TableEmployeesAfterScroll(DataSet: TDataSet);
begin
    UpdateStatusLine();
end;
//-----
procedure TFormMain.RadioGroupSearchMethodClick(Sender: TObject);
begin
    if (RadioGroupSearchMethod.ItemIndex = 0) then
    begin
        CheckBoxCaseInsensitive.Enabled := True;
        CheckBoxPartialKey.Enabled := True;
    end
    else
    begin
        CheckBoxCaseInsensitive.Enabled := False;
        CheckBoxPartialKey.Enabled := False;
    end
end;
//-----
procedure TFormMain.SpeedButtonDeleteLogItemClick(Sender: TObject);
var Msg: AnsiString;
begin
    if (ClientDataSetLog.RecordCount > 0) then
    begin
        Msg := 'Are you sure you want to delete record?';
        if (MessageDlg(Msg, mtConfirmation, mbYesNo, 0) = mrYes) then
        begin
            ClientDataSetLog.Delete();
        end
    end
    else
    begin
        ShowMessage('No record to be deleted!');
    end
end;
```

```
procedure TFormMain.SpeedButtonLogClearClick(Sender: TObject);
var Msg: AnsiString;
begin
    if (ClientDataSetLog.RecordCount > 0) then
    begin
        Msg := 'Are you sure you want to clear Log?';
        if (MessageDlg(Msg, mtConfirmation, mbYesNo, 0) = mrYes) then
        begin
            ClientDataSetLog.EmptyDataSet();
        end
    end
    else
    begin
        ShowMessage('No records to delete!');
    end
end;
```

```
procedure TFormMain.SpeedButtonLogLoadClick(Sender: TObject);
begin
    if (FileExists('Log.xml')) then
    begin
        ClientDataSetLog.LoadFromFile('Log.xml');
    end
    else
    begin
        ShowMessage('The [Log.xml] is not exists! First save the file.');
```

//-----

```
procedure TFormMain.SpeedButtonLogSaveClick(Sender: TObject);
begin
    ClientDataSetLog.SaveToFile('Log.xml', dfXML);
    ShowMessage('The [Log.xml] was successfully saved!');
```

//-----

```
procedure TFormMain.DBGridLogDblClick(Sender: TObject);
begin
    if (ClientDataSetLog.RecordCount > 0) then
    begin
        EditFirstNameEnter(Self);
        EditFirstName.Text := ClientDataSetLog.FirstName.AsString;
        EditLastName.Text := ClientDataSetLog.LastName.AsString;
    end
end;
```

```
//-----  
procedure TFormMain.EditEmpNoEnter(Sender: TObject);  
begin  
    EditEmpNo.Color := clWindow;  
    EditFirstName.Color := clBtnFace;  
    EditLastName.Color := clBtnFace;  
    EditFirstName.Text := '';  
    EditLastName.Text := '';  
end;  
  
//-----  
  
procedure TFormMain.EditEmpNoChange(Sender: TObject);  
begin  
    if (LabelResult.Caption <> '') then  
    begin  
        LabelResult.Caption := '';  
        LabelResult.Color := RGB(245, 245, 245);  
    end;  
    BitBtnSearch.Enabled :=  
        ((EditEmpNo.Text <> '') and  
        (EditFirstName.Text = '' and (EditLastName.Text = ''))) or  
        ((EditEmpNo.Text = '' and  
        (EditFirstName.Text <> '' or (EditLastName.Text <> '')));  
end;  
  
//-----  
  
procedure TFormMain.EditFirstNameEnter(Sender: TObject);  
begin  
    EditFirstName.Color := clWindow;  
    EditLastName.Color := clWindow;  
    EditEmpNo.Color := clBtnFace;  
    EditEmpNo.Text := '';  
end;  
  
//-----
```



```

procedure TFormMain.BitBtnSearchClick(Sender: TObject);
var S, E: Int64;
begin
    LabelResult.Caption := '';
    if (EditEmpNo.Text <> '') then MyCase := 0 else
    if ((EditFirstName.Text <> '') and
        (EditLastName.Text <> '')) then MyCase := 1 else
    if ((EditFirstName.Text <> '') and
        (EditLastName.Text = '')) then MyCase := 2 else
    if ((EditFirstName.Text = '') and
        (EditLastName.Text <> '')) then MyCase := 3;
    //комбинацията от стойности FirstName, LastName се добавя в Log-a
    //само ако не съществува вече там
    if ((MyCase > 0) and (CheckBoxAddToLog.Checked)) then
    begin
        if (not ClientDataSetLog.Locate('FirstName;LastName',
            VarArrayOf([EditFirstName.Text, EditLastName.Text]), [])) then
        begin
            ClientDataSetLog.Append();
            ClientDataSetLog.FirstName.AsString := EditFirstName.Text;
            ClientDataSetLog.LastName.AsString := EditLastName.Text;
            ClientDataSetLog.Post();
        end;
    end;
    case RadioGroupSearchMethod.ItemIndex of
        0: begin
            S := GetTickCount();
            LocateSearch();
            E := GetTickCount();
        end;
        1: begin
            S := GetTickCount();
            LookupSearch();
            E := GetTickCount();
        end;
        2: begin
            S := GetTickCount();
            ManualSearch();
            E := GetTickCount();
        end;
    end;
    StatusBar.Panels.Items[2].Text := ' Time: ' + IntToStr(E-S) + ' ms';
end;

```

```

procedure TFormMain.CheckBoxRestrictClick(Sender: TObject);
begin
    EditEmpNo.NumbersOnly := CheckBoxRestrict.Checked;
end;

```