

Знающие данные VCL имеют очень интересную особенность для визуального интерфейса. Когда мы работаем с управлением данных и компонентами набора данных, мы можем переместить фокус, используя метод **FocusControl TField**.

Однако, в некоторых случаях этого недостаточно. Даже если этот метод вызывался из модуля данных, иногда фокус не переходит к управлению, которое редактируется (например, **TDBGrid**).

Вашему вниманию представлена подходящая функция для использования на формах обслуживания набора данных, для нахождения управления редактирования поля в наборе данных, связанном с **DataSource**.

```
// ----- // GetControlField -
управление данных в окне (TForm, TPanel, и т.д.), // источник данных (TDataSource) и
имя поля, эта функция постарается найти и // вернуть управление, которое
используется для редактирования содержания поля, // даже сетку. // // от Rubem
Nascimento da Rocha (Manaus-AM/Brazil) //
----- type THackDBGrid =
class(TCustomDBGrid); function GetControlField(AOwner: TWinControl; ADataSource:
TDataSource; ADataField: String): TWinControl; const PropDataSource = 'DataSource';
PropDataField = 'DataField'; var I: Integer; Found: Boolean; RefWinControl:
TWinControl; CurrControl: TControl; PropInfo: PPropInfo; PropValue: String; Stack:
TStringList; // recurse engine AuxObject: TObject; function GridHasField: Boolean; var
_Field, _Qty: Integer; _AName: String; begin Result := False; with
THackDBGrid(CurrControl) do begin _Qty := DataLink.FieldCount; for _Field := 0 to
_Qty - 1 do begin _AName := UpperCase(DataLink.Fields[_Field].FieldName);
Result := (_AName = UpperCase(ADataField)); if Result then Break; end; end;
end; begin Found := False; RefWinControl := AOwner; I :=
Pred(RefWinControl.ControlCount); Result := NIL; Stack := TStringList.Create; try
while ((I >= 0) or (Stack.Count > 0)) and (not Found) do begin // Check if control will be
only checked checked or if all your // child controls, one-by-one, will do CurrControl :=
RefWinControl.Controls[I]; if CurrControl is TWinControl then if
TWinControl(CurrControl).ControlCount > 0 then begin
Stack.AddObject(IntToStr(I), RefWinControl); RefWinControl :=
TWinControl(CurrControl); I := Pred(RefWinControl.ControlCount); Continue;
end; PropInfo := GetPropInfo(CurrControl.ClassInfo, PropDataSource); if
Assigned(PropInfo) then begin AuxObject := TObject(GetOrdProp(CurrControl,
PropInfo)); // check if control is a data-aware grid if CurrControl is TCustomDBGrid
then Found := GridHasField else begin PropInfo :=
GetPropInfo(PTypeInfo(CurrControl.ClassInfo), PropDataField); if
```

```
Assigned(PropInfo) then      begin      // ... check if property has the search value
  PropValue := GetStrProp(CurrControl, PropInfo);      Found := (TDataSource(AuxObject)
= ADataSource) and      (UpperCase(PropValue) = UpperCase(ADataField));      end;
  end;      end;      while (Pred(I) = -1) and (Stack.Count > 0) do      begin      I :=
StrToInt(Stack[Pred(Stack.Count)]);      RefWinControl :=
TWinControl(Stack.Objects[Pred(Stack.Count)]);      Stack.Delete(Pred(Stack.Count));
end;      Dec(I);      if Found then      result := TWinControl(CurrControl);      end;      finally
Stack.Free;      end; end;
```