

TStringAlignGrid: Uma Alternativa Para o TStringGrid

Há muito tempo eu tive dificuldades em fazer um grid sem que o mesmo estivesse vinculado a um Dataset. Obviamente para isto existe o componente TStringGrid. Porém este é muito limitado. Por exemplo: Tente colocar o texto uma de suas células centralizado, ou mesmo numa fonte diferente das demais células, ou ainda trocar a cor de fundo de apenas uma das células. Eu não vi como fazer isto !

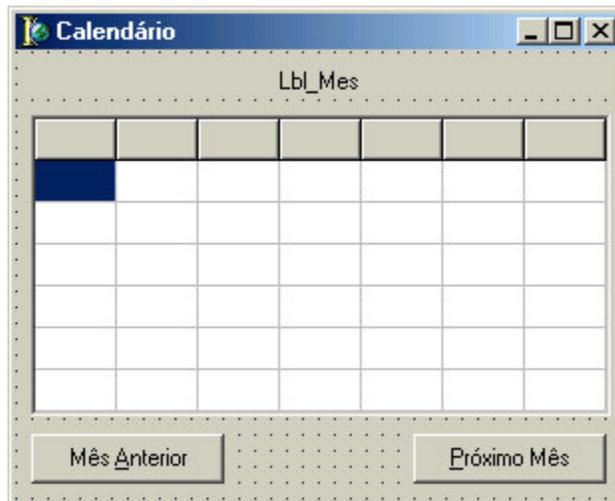
Procurando por um componente que me atendesse achei o TStringAlignGrid. Foi desenvolvido pelo Andreas Hoerstemeier (da Alemanha, eu acho). Este componente é uma derivação do TStringGrid com todos os recursos que eu precisei e muitos outros.

Vejamos as principais propriedades e qualquer um pode notar suas notórias vantagens:

HintCell[Coluna, Linha] (String)	É um hint específico para uma determinada célula. Ou seja, quando o ponteiro do mouse estiver sobre a célula especificada, o texto deste hint vai aparecer.
ShowCellHints (Boolean)	Com esta propriedade ligada (TRUE), o hint das células serão mostrados. Semelhante à propriedade ShowHint (que também deve estar ligada para que o hint das células sejam mostrados).
Alignment (TAlignment)	Alinhamento global do texto de todas as células do grid. Se todas as células têm o mesmo alinhamento, basta setar esta propriedade.
AlignCell[Coluna, Linha] (TAlignment)	Alinhamento individual de uma célula em específico.
AlignCol[Coluna] (TAlignment)	Alinhamento global de todas as células de uma determinada coluna.
AlignRow[Linha] (TAlignment)	Alinhamento global de todas as células de uma determinada linha.
Editing (Boolean)	Informa se o texto das células do grid poderá sofrer alterações em tempo de execução. Efeito global, ou seja, em todas as células do grid.
EditCell[Coluna, Linha] (Boolean)	Informa se o texto de uma célula em específico poderá sofrer alterações em tempo de execução.
EditCol[Coluna] (Boolean)	Informa se o texto de todas as células de uma coluna poderá sofrer alterações em tempo de execução.
EditRow[Linha] (Boolean)	Informa se o texto de todas as células de uma linha poderá sofrer alterações em tempo de execução.
ColorCell[Coluna, Linha] (TColor)	Informa a cor de fundo de uma determinada célula.
ColorCol[Coluna] (TColor)	Informa a cor de fundo de todas as células de uma determinada coluna.
ColorRow[Coluna] (TColor)	Informa a cor de fundo de todas as células de uma determinada linha.
CellFont[Coluna, Linha] (TFont)	Informa a fonte do texto de uma determinada célula. Como esta propriedade é do tipo TFont, pode-se informar o nome da fonte, o tamanho, o estilo, a cor, etc. Exemplo: Arial, tamanho 12, cor vermelho, negrito.
ColFont[Coluna] (TFont)	Informa a fonte do texto de todas as células de uma determinada coluna.
RowFont[Coluna] (TFont)	Informa a fonte do texto de todas as células de uma determinada linha.

É óbvio que não colocamos aqui todas as propriedades nem todas as características deste componente. Mas apenas algumas propriedades que não encontrei no TStringGrid e que são muito úteis. Existem outras propriedades igualmente interessantes, métodos e eventos que devem ser observados quando formos utilizar o componente.

Colocamos aqui um exemplo de um calendário feito utilizando o componente TStringAlignGrid. O formulário principal (e único) ficou assim:



Formulário	BorderStyle	BsSingle
	Caption	Calendário
	Height	252
	Name	Frm_Calend
	Position	PoScreenCenter
	Width	312
Label	Alignment	taCenter
	AutoSize	False
	Left	8
	Name	Lbl_Mes
	Width	289
StringAlignGrid	Alignment	taCenter
	ColCount	7
	DefaultColWidth	40
	DefaultRowHeight	20
	Editable	False
	FixedCols	0
	FixedRows	1
	Left	8
	Name	Grid
	RowCount	7
	ScrollBars	ssNone
	ShowCellHints	True
	ShowHint	True
	Top	32
Width	291	

TButton	Caption	Mês &Anterior
	Left	8
	Name	Btn_MesAnt
	Top	192
	Width	97
TButton	Caption	&Próximo Mês
	Left	200
	Name	Btn_ProxMes
	Top	192
	Width	97

Com o formulário criado e os componentes devidamente configurados (conforme especificações acima), vamos adicionar funcionalidades ao nosso calendário.

Primeiramente devemos criar uma procedure para limpar o grid. A procedure é a seguinte:

```

procedure TFrm_Calend.ResetGrid;
var
  Lin,
  Col: Integer;

begin
  //
  // Limpa todo o conteúdo do Grid (apenas da segunda linha para baixo)
  // Colocando as cores, estilo de texto e hints zerados.
  //
  for Lin := 1 to 6 do
    for Col := 0 to 6 do begin
      Grid.Cells[Col,Lin] := '';
      Grid.CellFont[Col, Lin].Style := [ ];
      Grid.CellFont[Col, Lin].Color := clWindowText;
      Grid.ColorCell[Col, Lin] := clWindow;
      Grid.HintCell[Col, Lin] := '';
    end;
end;

```

Agora uma outra procedure para montar o grid em um determinado mês:

```

procedure TFrm_Calend.MontaGrid(Data: TDateTime);
var
  DatFim: TDateTime;

  Dia,
  Mes,
  Ano,
  DiaFim: Word;

  Lin,
  Ind,
  DiaSem,
  DiaOld: Integer;

```

```

DatFer: array[0..7] of String;
HntFer: array[0..7] of String;

begin
  // Reseta o grid
  ResetGrid;

  // Feriados fixos
  // Devem ter outros, mas não me lembrei....
  DatFer[1] := '01/01';
  DatFer[2] := '01/05';
  DatFer[3] := '07/09';
  DatFer[4] := '12/10';
  DatFer[5] := '02/11';
  DatFer[6] := '15/11';
  DatFer[7] := '25/12';

  // Descrição dos feriados
  HntFer[1] := 'Confraternização Universal';
  HntFer[2] := 'Dia do Trabalho';
  HntFer[3] := 'Dia da Independência';
  HntFer[4] := 'Feriado Religioso';
  HntFer[5] := 'Dia de Finados';
  HntFer[6] := 'Proclamação da República';
  HntFer[7] := 'Natal';

  // Coloca a primeira linha com a fonte em Negrito
  Grid.FixedRowFont[0].Style := [ fsBold ];

  // Seta o texto da primeira linha (cabeçalho)
  Grid.Cells[0,0] := 'DOM';
  Grid.Cells[1,0] := 'SEG';
  Grid.Cells[2,0] := 'TER';
  Grid.Cells[3,0] := 'QUA';
  Grid.Cells[4,0] := 'QUI';
  Grid.Cells[5,0] := 'SEX';
  Grid.Cells[6,0] := 'SAB';

  // Obtém o mês para setar o label no alto do calendário
  DecodeDate(Data, Ano, Mes, Dia);
  case Mes of
    01: Lbl_Mes.Caption := 'Janeiro - '+IntToStr(Ano);
    02: Lbl_Mes.Caption := 'Fevereiro - '+IntToStr(Ano);
    03: Lbl_Mes.Caption := 'Março - '+IntToStr(Ano);
    04: Lbl_Mes.Caption := 'Abril - '+IntToStr(Ano);
    05: Lbl_Mes.Caption := 'Maio - '+IntToStr(Ano);
    06: Lbl_Mes.Caption := 'Junho - '+IntToStr(Ano);
    07: Lbl_Mes.Caption := 'Julho - '+IntToStr(Ano);
    08: Lbl_Mes.Caption := 'Agosto - '+IntToStr(Ano);
    09: Lbl_Mes.Caption := 'Setembro - '+IntToStr(Ano);
    10: Lbl_Mes.Caption := 'Outubro - '+IntToStr(Ano);
    11: Lbl_Mes.Caption := 'Novembro - '+IntToStr(Ano);
    12: Lbl_Mes.Caption := 'Dezembro - '+IntToStr(Ano);
  end;
end;

```

```

//
//  Obtém o último dia do mês, isto é feito assim:
//  - Se o mês for 12 (dezembro). Mês é igual a 1 e incrementa o ano.
//  - Senão incrementa o mês.
//  - Transformo para data, levando em consideração que o dia é
//  sempre 1 e o mês/ano será o calculado no passo anterior
//  - Subtraio 1 da data encontrada no passo anterior. Teremos a
//  última data do mês.
//  - Extraio o dia da data encontrada no passo anterior.
//
if Mes = 12 then begin
    Mes := 1;
    Inc(Ano);
end else
    Inc(Mes);

DatFim := EncodeDate(Ano, Mes, 1);
DatFim := DatFim - 1;

DecodeDate(DatFim, Ano, Mes, DiaFim);

// Loop do dia 1 até o último dia do mês

Lin := 0;
DiaOld := 9;
for Dia := 1 to DiaFim do begin
    //
    //  Transformo o ano, mês e dia para uma variável do tipo TDateTime
    //  Obtenho o dia da semana correspondente a data.
    //
    Data := EncodeDate(Ano, Mes, Dia);
    DiaSem := DayOfWeek(Data)-1;

    //  Se o dia da semana encontrado for menor que o dia da semana
    //  da data anterior é sinal que devemos mudar de linha...
    if DiaSem <= DiaOld then begin
        DiaOld := DiaSem;
        Inc(Lin);
    end;

    //  Escrevo o dia na célula
    Grid.Cells[DiaSem, Lin] := FormatFloat('00;00', Dia);

    //  Verifico se o dia é feriado...
    for Ind := 0 to High(DatFer) do
        if FormatDateTime('dd/mm', Data) = DatFer[Ind] then begin
            Grid.ColorCell[DiaSem, Lin] := clRed;
            Grid.CellFont[DiaSem, Lin].Color := clWhite;
            Grid.HintCell[DiaSem, Lin] := HntFer[Ind];
            Break;
        end;

```

```

// Se for o dia de hoje, coloco em negrito
if Data = Date then
    Grid.CellFont[DiaSem, Lin].Style := [ fsBold ];

// Se for o dia 01 coloco o foco do grid nesta célula
if Dia = 1 then begin
    Grid.Col := DiaSem;
    Grid.Row := Lin;
end;
end;
end;

```

Será necessário criar duas variáveis que devem ser visíveis em toda a Unit. Para isto coloque a definição das mesmas na cláusula Private:

```

private
{ Private declarations }
MesRef,
AnoRef: Word;

```

A seguir colocaremos no evento OnCreate do formulário o seguinte código:

```

procedure TFrm_Calend.FormCreate(Sender: TObject);
var
    Tmp: Word;
begin
    DecodeDate(Date, AnoRef, MesRef, Tmp);
    MontaGrid(Date);
end;

```

Agora colocaremos no evento OnClick do botão **Btn_MesAnt** o seguinte código:

```

procedure TFrm_Calend.Btn_MesAntClick(Sender: TObject);
begin
    Dec(MesRef);
    if MesRef < 1 then begin
        MesRef := 12;
        Dec(AnoRef);
    end;

    MontaGrid(EncodeDate(AnoRef, MesRef, 1));
end;

```

E para finalizar o projeto só falta o evento OnClick do botão **Btn_ProxMes** que terá o seguinte código:

```
procedure TFrm_Calend.Btn_ProxMesClick(Sender: TObject);
begin
  Inc(MesRef);
  if MesRef > 12 then begin
    MesRef := 1;
    Inc(AnoRef);
  end;

  MontaGrid(EncodeDate(AnoRef, MesRef, 1));
end;
```

O aplicativo rodando fica, mais ou menos, assim:



Com tudo isto o nosso calendário está pronto. Como o componente é simples de usar não é difícil de fazer coisas até mais complexas do que este exemplo. Numa utilização “profissional” deste código seria interessante ter uma tabela de feriados (fixos e móveis) com sua devida descrição, daí seria fácil colocar uma busca da data nesta mesma tabela e caso a data se tratar de um feriado, fazer as alterações que estão demonstradas no exemplo (ou outras).

É evidente que muita coisa pode ser implementada e melhorada. Mas o objetivo desta aplicação-exemplo é apenas didático. Portanto (aos programadores mais experientes) sejam caridosos ao avaliar o código.

O motivo pelo qual não coloquei o componente juntamente com o exemplo é que acho interessante que todos os interessados fizessem uma breve visita ao site do autor. Lá tem recomendações importantes para quem usa Delphi 6 em diante. O endereço é: <http://www.hoerstemeier.com> Chegando lá, clique no link “Delphi” e verá o componente TStringAlignGrid (além de outros).

Divirtam-se !

Marcelo Torres