

USANDO O TREEVIEW

O TreeView é um excelente componente visual que nos dá a possibilidade de visualizar dados em forma de “árvore”. O resultado final é um visual mais ou menos igual o Windows Explorer (no lado esquerdo onde mostra as pastas).

O exemplo que vamos mostrar aqui é de como “preencher” um TreeView de forma dinâmica, ou seja, de dentro de um programa. Fazer isto diretamente no componente nem tem tanta graça assim. ☺

Vamos lá, mãos a massa !

Imaginemos que os dados que irão compor o nosso TreeView virão de um banco de dados (isso sim é bem legal). Vamos então ver a fonte de dados.

Uma tabela com a seguinte estrutura:

TRV_CODIGO – INTEGER
TRV_DESCRI – VARCHAR(30)
TRV_CODPAI – INTEGER

Vamos identificar cada um dos campos. O campo “TRV_CODIGO” é a nossa chave primária (nunca repete e é seqüencial). O campo “TRV_DESCRI” é a descrição que irá aparecer no TreeView. Por fim, o campo “TRV_CODPAI” é o código do item ao qual este está vinculado, ou seja, o código do pai deste registro na árvore que será gerada no TreeView.

Vamos exemplificar uma árvore e o seu preenchimento na tabela OPCOES que propomos acima:

	TRV_CODIGO	TRV_DESCRI	TRV_CODPAI
Opção 1	1	Opção 1	0
Opção 1.1	2	Opção 1.1	1
Opção 1.2	3	Opção 1.2	1
Opção 1.3	4	Opção 1.3	1
Opção 1.3.1	5	Opção 1.3.1	4
Opção 1.3.2	6	Opção 1.3.2	4
Opção 2	7	Opção 2	0
Opção 2.1	8	Opção 2.1	7
Opção 2.1.1	9	Opção 2.1.1	8
Opção 2.1.2	10	Opção 2.1.2	8
Opção 2.1.3	11	Opção 2.1.3	8
Opção 2.2	12	Opção 2.2	7
Opção 2.3	13	Opção 2.3	7
Opção 2.4	14	Opção 2.4	7
Opção 2.4.1	15	Opção 2.4.1	14
Opção 2.4.2	16	Opção 2.4.2	14
Opção 3	17	Opção 3	0
Opção 4	18	Opção 4	0
Opção 4.1	19	Opção 4.1	18
Opção 4.2	20	Opção 4.2	18
Opção 5	21	Opção 5	0
Opção 5.1	22	Opção 5.1	21

Nos cabe agora um pouco de definição:

Um TreeView é composto de várias linhas. Estas linhas são chamadas de Nódulos (ou Nodes no inglês). Um Nódulo pode ser “pai” de outro Nódulo, criando assim a nossa árvore. O próprio TreeView pode ser considerado um Nódulo que contém vários outros Nódulos. O objeto que é utilizado como Nódulo do TreeView é o TTreeNode (sugestivo, não ?).

Bem, com isso matamos metade do problema. Agora vamos para a outra metade que é colocar os dados da nossa tabela no TreeView.

Para este trabalho, vamos criar uma procedure onde passaremos o Nódulo e o código do seu pai.

Uma consideração que vale mencionarmos é que: Para não complicarmos muito vamos, neste exemplo, usar o bom e velho BDE (sejam gentis quando me criticarem por isso)..

A procedure é a seguinte:

```
procedure TForm1.MontaTree(Nod: TTreeNode; Pai: Integer);  
var  
    Node: TTreeNode;  
    QFil: TQuery;  
    QPai: TQuery;  
  
begin  
    QFil := TQuery.Create(nil);  
    QFil.DatabaseName := 'Alias';  
  
    QPai := TQuery.Create(nil);  
    QPai.DatabaseName := 'ALIAS';  
  
    QFil.SQL.Add('SELECT * FROM OPCOES');  
    QFil.SQL.Add('WHERE TRV_CODPAI = '+IntToStr(Pai));  
    QFil.SQL.Add('ORDER BY TRV_CODIGO');  
    QFil.Open;  
  
    while not QFil.Eof do begin  
        if Nod = nil then  
            Node := TreeView1.Items.Add(nil, QFil.FieldName('TRV_DESCRI').AsString)  
        else  
            Node := TreeView1.Items.Add(Nod, QFil.FieldName('TRV_DESCRI').AsString);  
  
        QPai.SQL.Clear;  
        QPai.SQL.Add('SELECT COUNT(*) AS QUANT FROM OPCOES');  
        QPai.SQL.Add('WHERE TRV_CODPAI = '+  
            IntToStr(QFil.FieldName('TRV_CODIGO').AsInteger));  
        QPai.Open;  
  
        if QPai.FieldName('QUANT').AsInteger > 0 then  
            MontaTree(Node, QFil.FieldName('TRV_CODIGO').AsInteger);  
  
        QPai.Close;  
        QFil.Next;  
    end;  
  
    QFil.Close;  
    QFil.Free;  
    QPai.Free;  
end;
```

Vamos entender o que foi escrito ? É bem fácil. Passamos para a procedure o nóduo pai (sobre o qual iremos inserir os filhos) e o código do pai. Bem, se é a primeira vez que chamamos a procedure devemos passar o valor nil como o nóduo e o valor 0 (zero) como o código do pai.

Assim que o filho é inserido, é feito uma checagem para ver se este filho possui filhos. Em caso positivo a mesma procedure será chamada passando como nóduo pai o nóduo que acabou de ser criado, bem como o seu código).

Portanto, só para completarmos, a forma de chamarmos a nossa procedure é:

```
MontaTree(nil, 0);
```

Depois incrementaremos o recurso de colocarmos ícones nos nós, que também é igualmente fácil.

Divirtam-se !

Marcelo Torres