

How to create a TTreeView with a three state checkbox

QUESTION

I tried many combinations of *GW_STYLE* with *TVS_CHECKBOXES* or *BS_AUTO3STATE* and I can't get a three state checkbox. All I have is a plain 2 state box. Any ideas?

Actually, you can have any number of checkbox states you like. The number of the images in the state image list determines the number of the states. By default, the image list has two bitmaps: checked and unchecked. But you are always able to add yours for a third (forth ...) state. The code below shows a *TTreeView* with checkboxes and a third state. I've tested it on D4 and it seemed to work alright. You can set the third state to the tree node by setting 3 to the *StateIndex* property in the form's *OnCreate* event or in any other suitable place:

```
MyTreeView1.Items[0].StateIndex := 3;
```

```
{ ... }
type
  TMyTreeView = class(TTreeView)
  protected
    procedure CNNotify(var Message: TWMNotify); message CN_NOTIFY;
    procedure CreateParams(var Params: TCreateParams); override;
  public
    procedure AddNewStateImage;
  end;

{ ... }

procedure TMyTreeView.CreateParams(var Params: TCreateParams);
begin
  inherited CreateParams(Params);
  Params.Style := Params.Style or TVS_CHECKBOXES;
end;

procedure TMyTreeView.CNNotify(var Message: TWMNotify);
begin
  with Message do
    if NMHdr^.code = NM_CUSTOMDRAW then
      AddNewStateImage;
  inherited;
end;

procedure TMyTreeView.AddNewStateImage;
var
  XImageList: TImageList;
  XImage: HIMAGELIST;
  XBitMap: TBitMap;
  i: integer;
begin
  XImage := TreeView_GetImageList(Handle, TVSIL_STATE);
  if (XImage <> 0) and (ImageList_GetImageCount(XImage) < 4) then
  begin
    XImageList := TImageList.Create(Self);
    XBitMap := TBitMap.Create;
    try
      XImageList.ShareImages := true;
      XImageList.Handle := XImage;
      XBitMap.Width := XImageList.Width;
      XBitMap.Height := XImageList.Height;
      XImageList.Draw(XBitMap.Canvas, 0, 0, 2, false);
      XImageList.Add(XBitMap, nil);
    finally
      XImageList.Free;
      XBitMap.Free;
    end;
    for i := 0 to Items.Count - 1 do
```

```
    if Items[i].StateIndex > 0 then  
        Items[i].StateIndex := Items[i].StateIndex;  
    end;  
end;
```

Original resource:	The Delphi Pool
Author:	Serge Gubenko
Added:	2009-10-28
Last updated:	2009-10-28

Copyright © Peter Johnson (DelphiDabbler) 2002-2018
