How to make a TCollectionItem contain a TCollection

> ***QUESTION***
>
> I would like to create a component that contains a *TCollection* with *TCollectionItems*. Then I want to contain another *TCollection* with *TCollectionItems* within the *TCollectionItems*. I am trying to create a collection of sections, which would contain a collection of items for each section.

It's not difficult to implement such functionality. One thing you need to care about is the valid owner for your collections, presumably, the main control would be the best choice. Below is an example of such a component:

```
{ ... }
type
  TYourCollectionItem = class;
  TYourCollection = class;
  TColControl = class;

  TYourCollectionItem = class(TCollectionItem)
  protected
    FFirstString: string;
    FChildCollection: TYourCollection;
    procedure SetIndex(Value: Integer); override;
    function GetDisplayName: string; override;
  public
    constructor Create(Collection: TCollection); override;
    destructor Destroy; override;
    procedure Assign(Source: TPersistent); override;
  published
    property FirstString: string read FFirstString write FFirstString;
    property ChildCollection: TYourCollection
      read FChildCollection write FChildCollection;
  end;

  TYourCollection = class(TOwnedCollection)
  protected
    function GetItem(Index: Integer): TYourCollectionItem;
    procedure SetItem(Index: Integer; Value: TYourCollectionItem);
  public
    constructor Create(AOwner: TPersistent);
    property Items[Index: Integer]: TYourCollectionItem
      read GetItem write SetItem;
  end;

  TColControl = class(TComponent)
  protected
    FCollection: TYourCollection;
  public
    constructor Create(AOwner: TComponent); override;
    destructor Destroy; override;
  published
    property Collection: TYourCollection
      read FCollection write FCollection;
  end;

{ ... }

constructor TYourCollectionItem.Create(Collection: TCollection);
begin
  inherited Create(Collection);
  FChildCollection := TYourCollection.Create(Collection.Owner);
end;

destructor TYourCollectionItem.Destroy;
begin
  FChildCollection.Free;
  inherited Destroy;
```

```
end;

procedure TYourCollectionItem.SetIndex(Value: Integer);
begin
  inherited SetIndex(Value);
  ShowMessage(IntToStr(Value));
end;

function TYourCollectionItem.GetDisplayName: string;
begin
  Result := FFirstString;
end;

procedure TYourCollectionItem.Assign(Source: TPersistent);
begin
  FFirstString := TYourCollectionItem(Source).FFirstString;
  FChildCollection.Assign(TYourCollectionItem(Source).ChildCollection);
end;

constructor TYourCollection.Create(AOwner: TPersistent);
begin
  inherited Create(AOwner, TYourCollectionItem);
end;

function TYourCollection.GetItem(Index: Integer): TYourCollectionItem;
begin
  Result := TYourCollectionItem(inherited GetItem(Index));
end;

procedure TYourCollection.SetItem(Index: Integer;
  Value: TYourCollectionItem);
begin
  inherited SetItem(Index, Value);
end;

constructor TColControl.Create(AOwner: TComponent);
begin
  inherited Create(AOwner);
  FCollection := TYourCollection.Create(Self);
end;

destructor TColControl.Destroy;
begin
  FCollection.Free;
  FCollection := nil;
  inherited Destroy;
end;
```

Original resource:	The Delphi Pool
Author:	Serge Gubenko
Added:	2009-10-26
Last updated:	2009-10-26