How to write a custom TAction to control the visibility of a TStatusBar

Question

I am trying to write a custom action that will set the visible property of a *TStatusBar* on and off. I assigned this action to a menu item and when I select this menu item at runtime the status bar is hidden. The problem is that the menu item (connected to the action) is disabled, so I can't view the statusbar again. I think that it's a matter of how the *TMenuActionLink* behaves (the action controls the *Enabled* property of the menu). I tried to set the *Enabled* property in the action to true, but no avail. The menu is still disabled. Is there any way to do this?

I think that the best solution would be to write an action, which will have a *StatusBar* property and, in case this property was assigned, set the statusbar's visibility in the overridden *Execute* method. Here's an example:

```
{ ... }
  TMyAction = class(TAction)
  protected
    FStatusBar: TStatusBar;
    procedure Notification(AComponent: TComponent; Operation: TOperation);
      override;
    procedure SetStatusBar(AValue: TStatusBar);
  public
    constructor Create(AOwner: TComponent); override;
    function Execute: Boolean; override;
  published
    property StatusBar: TStatusBar read FStatusBar write SetStatusBar;
  end;

{ ... }

constructor TMyAction.Create(AOwner: TComponent);
begin
  inherited Create(AOwner);
  DisableIfNoHandler := false;
  FStatusBar := nil;
  Caption:='Turn On / Off Status Bar';
end;

function TMyAction.Execute: Boolean;
begin
  Result := inherited Execute;
  if Assigned(FStatusBar) then
  begin
    FStatusBar.Visible := not FStatusBar.Visible;
    Checked := FStatusBar.Visible;
  end;
end;

procedure TMyAction.Notification(AComponent: TComponent; Operation: TOperation);
begin
  inherited Notification(AComponent, Operation);
  if (Operation = opRemove) and (AComponent = StatusBar) then
    StatusBar := nil;
end;

procedure TMyAction.SetStatusBar(AValue: TStatusBar);
begin
  if FStatusBar <> AValue then
  begin
    FStatusBar := AValue;
    if Assigned(FStatusBar) then
    begin
      FStatusBar.FreeNotification(Self);
      Checked := FStatusBar.Visible;
    end
    else
```

```
      Checked := false;
  end;
end;
```

Original resource:	The Delphi Pool
Author:	Serge Gubenko
Added:	2013-01-27
Last updated:	2013-01-27