

How to implement a custom component paint method that is dependent on outside data

QUESTION

I want to create a custom component that will have its normal (inherited) paint method.

However, if certain graphic data in the main program is present, I want the component to first run its inherited paint routine and then execute a procedure in the main program that has access to the data (also in the main program).

What is the best way to accomplish this?

You solve this by using a custom event to which you can attach a form method as needed:

```
type
  TMyComp = class( TCustomControl ) {or TGraphicsControl}
  private
    FAfterPaint: TNotifyEvent; {may use custom event type}
  protected
    procedure Paint; override;
    procedure DoAfterPaint; virtual;
  published
    property AfterPaint: TNotifyEvent read FAfterPaint write FAfterPaint;
  end;

procedure TMyComp.Paint;
begin
  inherited;
  DoAfterPaint;
end;

procedure TMyComp.DoAfterPaint;
begin
  if assigned( FAfterPaint ) and NeedsCustomPaint then
    FAfterPaint( self );
end;
```

Original resource:	The Delphi Pool
Author:	Peter Below
Added:	2012-07-08
Last updated:	2012-07-08

Copyright © Peter Johnson (DelphiDabbler) 2002-2018