

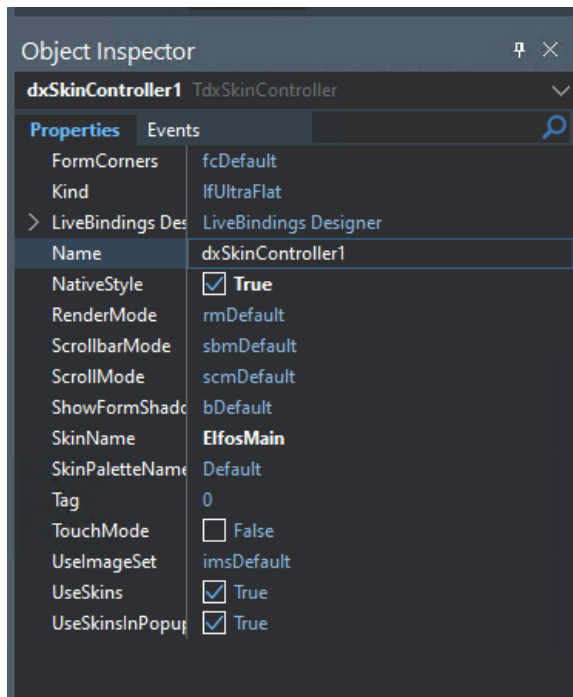
# Com canviar Skin de components

## Contents

dxSkinController.....	2
Com canviar la Skin del control individual .....	3
Canvieu automàticament els diferents tipus de components .....	5
cxPageController .....	5
cxButtons.....	5
Saber quin propietat canviar .....	7

## dxSkinController

DevExpress té un component anomenat **dxSkinController** que permet utilitzar i controlar Skins en Delphi.



La propietat que ens interessa més és

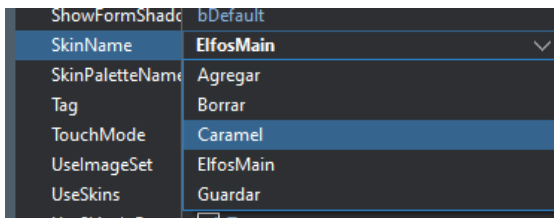
**NativeStyle**. Si està configurat a:

**True**: Els components utilitzen l'estil natiu del sistema operatiu. Per exemple: en Windows 10, els components tindran un estil quadrat, mentre que en Windows 11 utilitzarà un estil arrodonit.

**False**: En lloc de la finestra del sistema operatiu, s'utilitzarà l'estil de la Skin.

Hauria estat millor configurar-lo a false, però he hagut de configurar-lo a true perquè alguns components que no són de DevExpress no suporten les skins. Per tant, l'aplicació es trencava a causa d'això.

Per superar aquest problema, només els components que volíem canviar l'experiència es van configurar a false i se'ls va aplicar la skin. Altres components van utilitzar l'estil natiu.

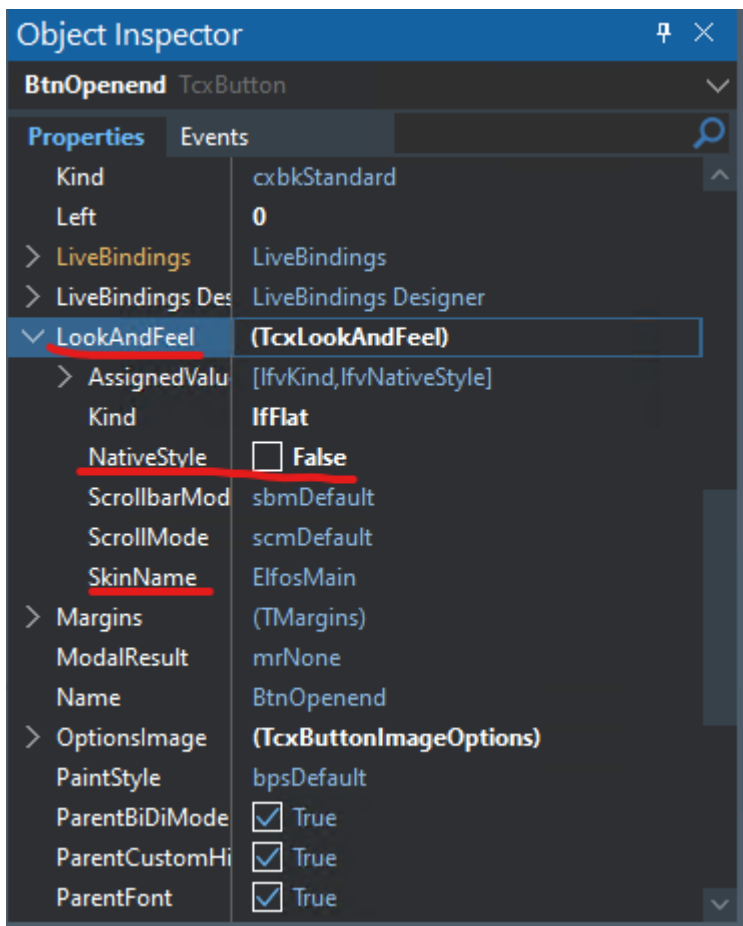


Una altra propietat important és **SkinName**, aquesta opció ens permet utilitzar la Skin que desitgem. DevExpress té la seva pròpia skin, però també podem utilitzar skins personalitzades

## Com canviar la Skin del control individual

Per canviar la Skin d'un controlador individual per anar al component específic. El component ha de ser component DevExpress (cx).

Per fer-ho un a un, la propietat que hem de veure és **LookAndFeel**

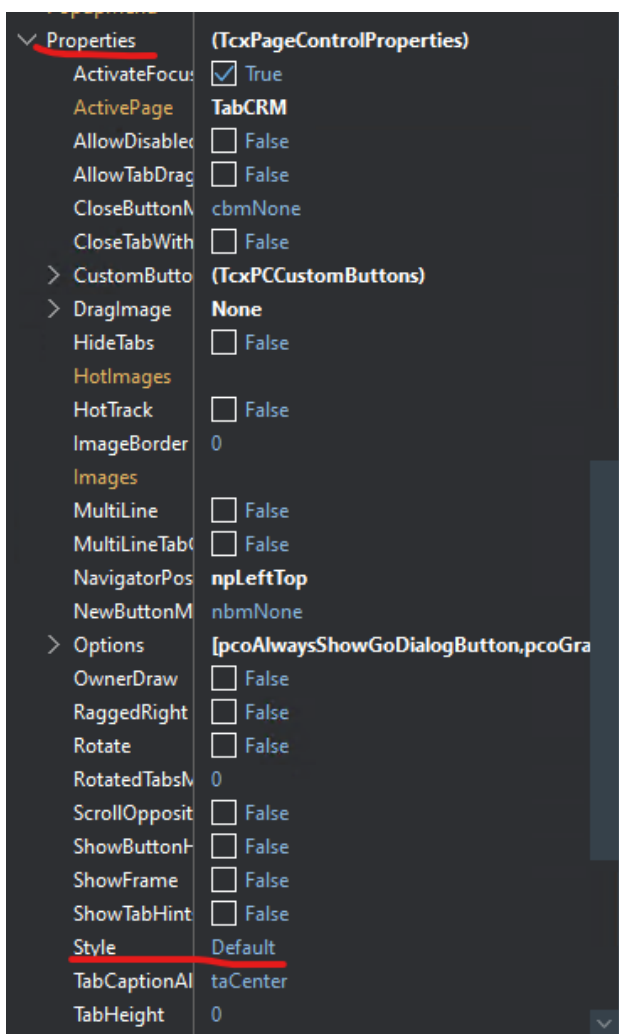


Té sub propietat de les quals hem de veure aquestes:

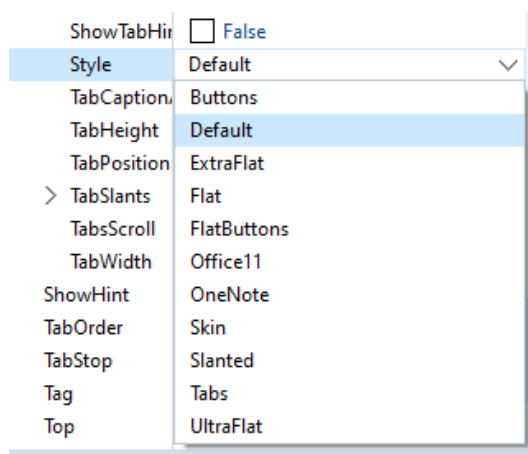
**NativeStyle:** És com abans nomes aplicar a aquest component.

**SkinName:** Normalment heretarà el dxskincontrolador assignat a la Skin, però es pot canviar aquí per a components individuals.

cxPagecontroller té una propietat més que hem de veure, **Propietats** :



Aquesta pestanya té un subproperti que ens permet controlar l'estil d'una pestanya a cxPageControll. Es diu Estil: abans de fer el canvi es va establir a **Slanted** , però volem que es posi **Default** o **Skin** perquè prengui la Skin que triem.



## Canvieu automàticament els diferents tipus de components

No volem canviar els diferents components individualment. Com que Elfos té molts components, trigarà molt de temps. Per tant, per superar aquesta **UC\_TraducirIdioma** unitat s'utilitza per canviar automàticament els diferents components en funció del seu tipus, llegenda o pista, etc.

La sintaxi bàsica és així

```
if componente is (TcxPageControl) then
begin
    TcxPageControl(Componente).Properties.Style           := 11;
    TcxPageControl(Componente).Font.Name                 := 'Roboto';
    TcxPageControl(Componente).Font.Style                := [fsBold];
    TcxPageControl(Componente).Font.Color                := $00333333;
    TcxPageControl(Componente).LookAndFeel.NativeStyle := False;
end;
```

Primer es comprova el tipus del component, després es canvien manualment diferents propietats. Les línies més importants és

**LookAndFeel.NativeStyle** ha de ser false si no funciona

```
TcxPageControl(Componente).LookAndFeel.NativeStyle := False;
```

Això s'ha d'establir en fals, perquè abans vam establir que els components seran d'estil natiu. Però si volem canviar-los s'han de posar en **fals**.

Una altra línia important és on posem la Skin que utilitzaran els nostres components. Si no en posem cap, utilitzarà la Skin que està configurada a dxSkinController. Però si volem utilitzar Skins diferents la sintaxi és així.

S'ha de canviar **LookAndFeel.SkinName** a skin que volem aplicar

```
TcxPageControl(Componente).LookAndFeel.NativeStyle := False;
TcxGrid(Componente).LookAndFeel.SkinName           := 'ElfosMain';
end;
```

### cxPageController

cxPageController té una propietat més per canviar, ja que com abans **Properties.Style** s'ha d'establir per defecte o Skin. Per tant, hem d'utilitzar això.

```
TcxPageControl(Componente).Properties.Style := 11;
```

En lloc d'un nom que utilitza un valor enter, he trobat que 11 és o Bé Skin o Per defecte.

### cxButtons

Elfos té diferents tipus de botons, de manera que necessitem skins diferents per a cadascun que depenen del seu títol.

Per tant, hem d'utilitzar codi que tingui en compte això, de manera que la sintaxi que utilitzem és aquesta

```

if componente is (TcxButton) then
begin
    TcxButton(Componente).Font.Name           := 'Roboto';
    TcxButton(Componente).Font.Style          := [fsBold];
    TcxButton(Componente).LookAndFeel.SkinName := 'ElfosMain';
    TcxButton(Componente).Colors.DefaultText   := $008A847C;
    TcxButton(Componente).Colors.HotText       := $008A847C;
    TcxButton(Componente).LookAndFeel.NativeStyle := False;

    if TcxButton(Componente).Hint <> 'NO' then
    begin
        sCaption := TcxButton(Componente).Caption;

        if (sCaption = 'Modificar') or (sCaption = 'Modifica') or (sCaption = 'Guardar') or
            (sCaption = 'Guardar cambios') or (pos(sCaption, 'uardar') > 0) or (sCaption = 'Calcular') then
        begin
            TcxButton(Componente).Colors.DefaultText := clWhite;
            TcxButton(Componente).Colors.HotText       := clWhite;
            TcxButton(Componente).LookAndFeel.SkinName := 'Guardar';
        end;

        if (sCaption = 'Salir') or (sCaption = 'Cancelar') or (sCaption = 'Cancela') or (sCaption = 's
        begin
            TcxButton(Componente).Colors.DefaultText := clWhite;
            TcxButton(Componente).Colors.HotText       := clWhite;
            TcxButton(Componente).LookAndFeel.SkinName := 'Guardar';
        end;

        if (sCaption = 'Agrega') or (sCaption = 'Agregar') or (sCaption = 'Nuevo') or (sCaption = 'Nue
            or (sCaption = 'Añadir') or (sCaption = 'Crear') then
        begin
            TcxButton(Componente).Colors.DefaultText := clWhite;
            TcxButton(Componente).Colors.HotText       := clWhite;
            TcxButton(Componente).Colors.NormalText    := clWhite;
            TcxButton(Componente).LookAndFeel.SkinName := 'Agregar';
        end;

        if (sCaption = 'Aceptar') or (sCaption = 'Acepta') and (sCaption = 'Seleccionar') then
        begin
            TcxButton(Componente).Colors.DefaultText := clWhite;
            TcxButton(Componente).Colors.HotText       := clWhite;
            TcxButton(Componente).LookAndFeel.SkinName := 'Guardar';
        end;
        if (sCaption = 'Borrar') or (sCaption = 'Borra') then
        begin
            TcxButton(Componente).Colors.DefaultText := clWhite;
            TcxButton(Componente).Colors.HotText       := clWhite;
            TcxButton(Componente).LookAndFeel.SkinName := 'Borrar';
        end;
    end;
end;

```

Primer apliquem la Skin d'ElfosMain a tots els botons, però si tenen AGREGAR escrit sobre ells se'ls assigna la Skin que té botó Verd. Vaig crear diferents Skins on l'únic canvi era el botó. Abans només calia canviar els colors. Però si s'utilitzen Skins, els colors no es poden canviar.

Si volem saltar-nos l'assignació de Skin a un component podem utilitzar la següent sintaxi

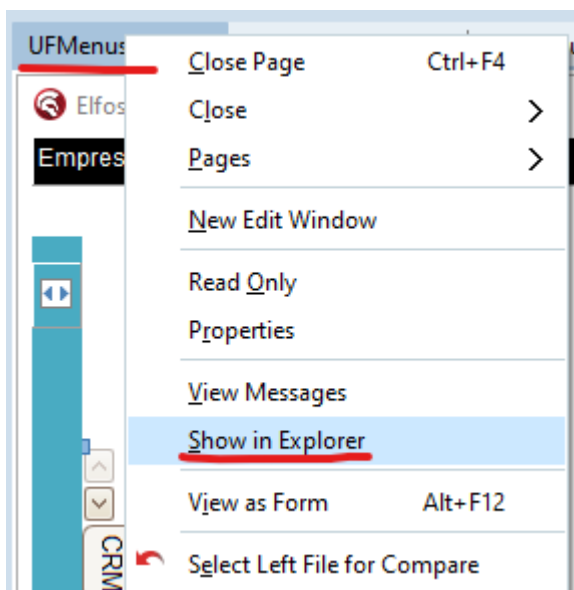
```
if Tpanel(Componente).Hint = 'NO' then Exit; ↵
```

Aquí es pot canviar la propietat de **Hint** de manera que no hi tingui cap escrit, s'omet l'assignació de Skin.

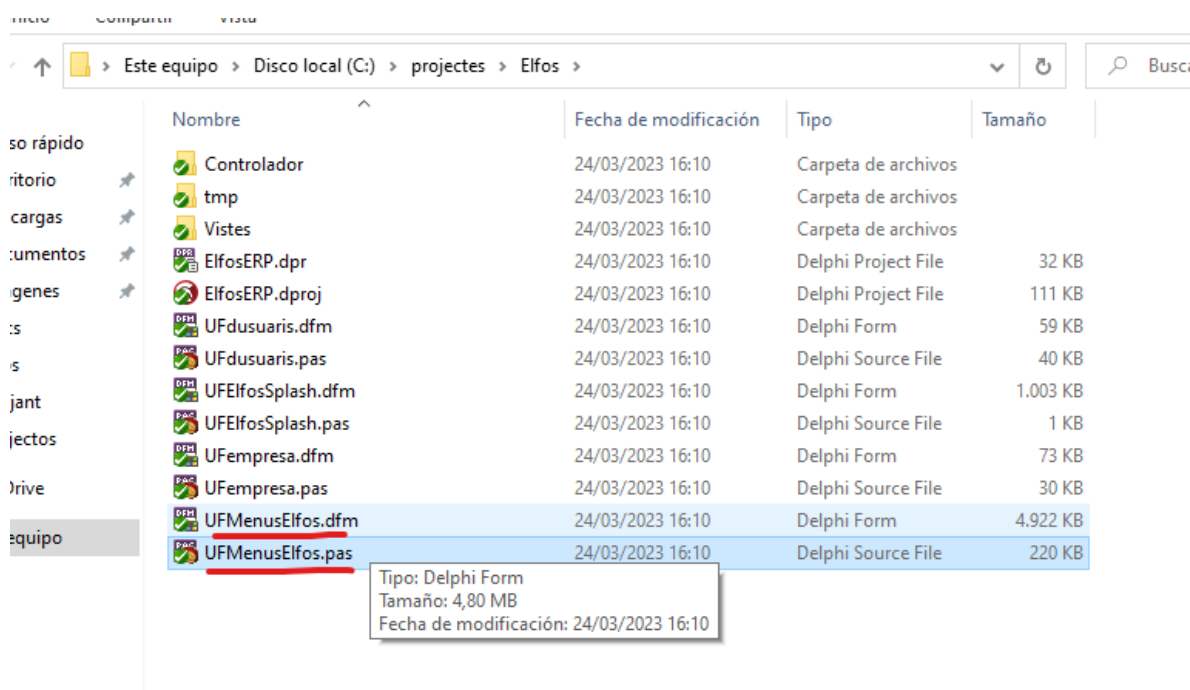
## Saber quin propietat canviar

Si vols saber què has de posar a la UC\_TraducirIdioma canviar la propietat que vols canviar, Segueix aquests passos

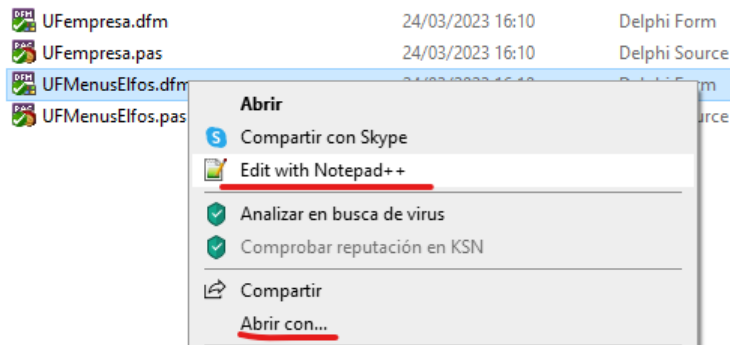
Cerqueu on es desa el fitxer de la unitat a la carpeta. Feu clic amb el botó dret al nom del fitxer obert a la barra de pestanyes Delphi i seleccioneu Mostra a l'Explorador.



Això obrirà l'Explorador de fitxers de Windows a la ubicació on s'emmagatzema el fitxer. Es ressaltarà l'arxiu. Acabarà en extensió de fitxer .pas però el fitxer que volem acaba en .dfm. Hauria de ser el mateix nom + .dfm.



Ara feu clic amb el botó dret al fitxer .dfm i obriu-lo a l'editor de text que vulgueu, en el meu cas Bloc de notes ++. No s'hauria d'obrir a Delfos.



S'obrirà un fitxer on es guarda tota la informació de disseny de la unitat Delphi. Només cal que cerqueu el nom del component i en mostrarà les propietats.

```

top = 1
Width = 65
Height = 23
Margins.Top = 6
Align = alRight
Caption = 'Agenda'
LookAndFeel.Kind = lfFlat
LookAndFeel.NativeStyle = False
TabOrder = 5
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
OnClick = BtnArribaClick
ExplicitLeft = 944
end
object BtnNotasArriba: TcxButton
Left = 1013
Top = 1
Width = 65
Height = 23
Hint = #13#10
Margins.Top = 6
Align = alRight
Caption = 'Notas'
LookAndFeel.NativeStyle = False
LookAndFeel.SkinName = 'ElfosMain'
TabOrder = 6
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = [fsBold]
ParentFont = False
OnClick = BtnNotasArribaClick
ExplicitLeft = 1000

```

Aquí podeu seleccionar qualsevol propietat que vulgueu canviar i escriure-la en UC\_TraducirIdioma

Per exemple, si voleu incloure el color de la font del TcxButton, podeu copiar la propietat que diu

Font.Color = clWindowText

I adaptar-lo per a la traducció.



Per adaptar-vos podeu veure altres transslacions i només cal canviar el nom de la propietat com aquí

```
--o--
TcxButton(Componente).Font.Name      := 'Roboto';
TcxButton(Componente).Font.Style     := [fsBold];
TcxButton(Componente).LookAndFeel.SkinName := 'ElfosMain';
TcxButton(Componente).Colors.DefaultText := $008A847C;
TcxButton(Componente).Colors.HotText   := $008A847C;
```