
Hoofdstuk 10

Dialogen

- Openen en opslaan van bestanden.
- Fonts en kleuren.
- Printer dialogen.

10

Dialogen

De tab Dialogs heeft een aantal dialogen voor bestanden, kleuren, fonts, images, kalender en rekenmachine en de printer.



OpenDialog
SaveDialog
SelectDirectoryDialog
ColorDialog
FontDialog
FindDialog
ReplaceDialog
OpenPictureDialog
SavePictureDialog
CalendarDialog
CalculatorDialog
PrinterSetupDialog
PrintDialog
PageSetupDialog

10 We maken weer een nieuwe les aan:

- Maak een nieuwe map in de lessen-map: **L10_Dialogs**
- Klik op **L03_Start.lpi** om dit project te starten
- De constanten **modTxt** en **verTxt** aanpassen naar **L10_Dialogs** en **Ver: 1.0**
- Met **Bestand|Opslaan Als** de pascalfile opslaan als **L10_DialogsP.pas**.
- Kies **Ja** op de vraag of de directory moet worden toegevoegd.
- Kies **Nee** op de vraag of de oude bestanden moeten worden verwijderd.
- Met **Project|Project Opslaan Als** het project opslaan: **L10_Dialogs.lpr**.
- Met **Project|Project Opties** de project titel aanpassen met: **L10_Dialogs**.

De manier om een Dialog te gebruiken is: de methode **Execute** te starten. We hebben dat in een aantal voorbeelden al gezien. De **Execute** methode geeft "True" terug als er een bestand is gekozen en "False" als op sluiten of annuleren is geklikt. In dat geval zullen we geen actie willen ondernemen.

Verder kunnen er een aantal initiële voorwaarden en filters worden in gesteld. Een titel geeft de mogelijkheid om wat duidelijker te zijn over het doel van de gebruikte Dialog. De hint ontbreekt.

OpenDialog, SaveDialog, SelectDirectoryDialog

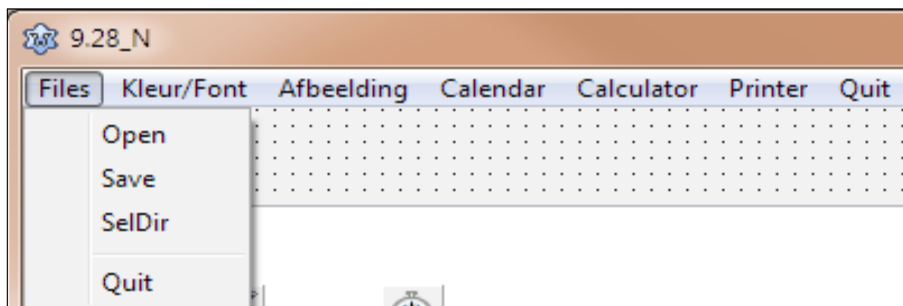
Laten we eerst eens spelen met de Open- Save- en SelectDir Dialogs.

Plaats een Open, een Save en een SelectDirectory dialoog op de Form door te dubbelklikken op SelectDirectoryDialog, OpenFileDialog, SaveDialog en in de tab Dialogs.

Noem de Dialogs: **DlOpen**, **DlSave** en **DlSelDir**.

Nu hebben we nog iets nodig om de Dialogs te starten en we moeten iets hebben om een bestand in te openen en te saveen, een Memo.

Voeg de menu-items: Open, Save en SelectDir toe aan een Menu File:



10

We doen dit door om MainMenu te klikken. De Menu-editor opent en we kunnen vóór Quit een item "Files" invoegen. Rechtsklikken op "Quit", dan Nieuw Item (voor) invoegen kiezen en dit item **MnFiles** noemen met de Caption: "Files".

Vervolgens rechtsklikken op Files en een nieuw item als submenu invoegen en dan nóg vier keer Nieuw Item (na) invoegen.

Naam	Caption
MnFlOpen	Open
MnFlSave	Save
MnFlSelDir	SelDir
MnFlSpace	-
MnFlQuit	Quit

MnFlSpace is er alleen om een separator in het menu te laten zien.

Kies in de OnClick Event van **MnFlQuit**: MnQuitClick.

Voeg ook een memo toe en noem het **MeData**.

We hebben nu alles om eens wat te openen en te saven. Ga naar de Eventhandler van **MnFlOpen** en dubbelklik op **OnClick**. Lazarus zal nu een nieuwe procedure aanmaken:

```
procedure TForm1.MnFlOpenClick(Sender: TObject);
begin
    ...
end;
```

Als we op **Files|Open** klikken wordt deze procedure aangeroepen en kunnen we hier de gewenste code uitvoeren.

We kunnen nu in deze procedure programmeren wat we voor zinnig houden. Laten we de procedure aanpassen met de volgende code:

```
procedure TForm1.MnFlOpenClick(Sender: TObject);
var
    s: string;
begin
    ① DLOpen.Title:= modTxt + 'Selecteer file.';
    ② Sb.Panels[0].Text:= '';
    ③ If DLOpen.Execute = true then
        begin
            ④ s:= DLOpen.FileName;
            ⑤ MeData.Lines.LoadFromFile(s);
            ⑥ Sb.Panels[0].Text:= s;
        end;
end;
```

10

- ① Eerst passen we de titel aan van het Opendialoog scherm. Het is altijd verstandig om even aan te geven wat er eigenlijk geopend moet worden.
- ② We willen de naam van het geopend bestand in de StatusBar weergeven. Voor het geval, dat er op annuleren wordt geklikt, willen we niets weergeven.
- ③ Nu starten we de Opendialoog met de opdracht `DLOpen.Execute`. Het resultaat is `true` wanneer er inderdaad een bestand is gekozen om te openen, `false` wanneer niets wordt gekozen of op "annuleren" wordt geklikt.
- ④ Wanneer een bestand is gekozen (het resultaat `true` is), dan wordt in `s` de bestandsnaam gezet.
- ⑤ Dan laden we het bestand in het memo, kunnen we zien wat we hebben geopend.
- ⑥ En tenslotte laten we even de naam van het geopende bestand zien in `panel[0]` van de statusbar.

Als we compileren en het programma starten, dan zien we de volgende dialoog verschijnen, na op Files | Open te hebben geklikt:

We kunnen nu iedere file kiezen. Het zal duidelijk zijn, dat alleen tekstfiles goed leesbaar worden in de Memo. Dus de .exe of .o bestanden zullen niet al te veel zinnige informatie geven.

Klik op een bestand, b.v. L10_DialogsP.pas en de tekst zal zichtbaar worden in MeData.

Er zijn een aantal zaken in te stellen bij de OpenFileDialog.

DefaultExt geeft de default extensie. Schijnt niet echt te werken in de OpenFileDialog.

Het is mogelijk om in "FileName" al een bestand aan te wijzen. Als bekend is welk bestand moet worden geopend is het misschien niet zo handig om er nog eens om te vragen.

De "Filter" is bruikbaar. Hier kan een groep bestanden gedefinieerd worden. De filter bestaat uit een omschrijving en een extensie, gescheiden door |. B.v.:

```
Tekst bestanden|*.txt|Pascal bestanden|*.pas|Resource bestanden |*.lrs|Alle bestanden|*.*
```

10

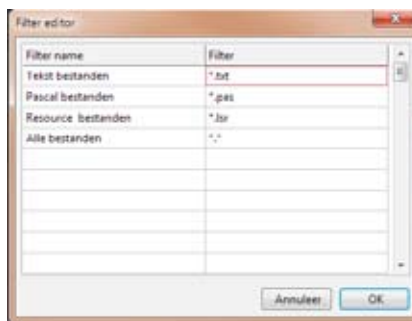
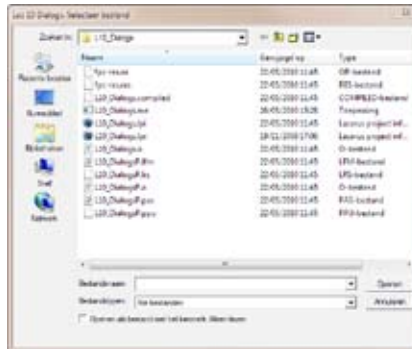
Deze filter resulteert in de volgende bestandstypen:

Het is overigens verstandig om altijd "Alle bestanden" mee te geven. Er kan dan in ieder geval gekeken worden of er iets in de gekozen map staat.

Het is ook mogelijk om in het programma zelf de filters aan te passen. Het type filter kan afhankelijk gemaakt worden van vorige selecties of één OpenFileDialog kan voor meerdere dialogen gebruikt worden binnen de applicatie.

De aanpassing van de Filter zou dan als volgt gaan:

```
DlOpen.Filter:= 'Tekst bestanden|*.txt
                |Pascal bestanden|*.pas
                |Resource bestanden|*.lrs
                |Alle bestanden|*.*';
```



Verder kunnen de filters ook met de Filter-editor worden ingesteld. Die wordt gestart door te dubbelklikken op "Filter" in de Object-inspector. Er verschijnt dan de Filter Editor, waar bestandsnaam en bestandstype kunnen worden ingevoerd.

De `FilterIndex` geeft aan, welke filters kunnen worden gebruikt. Als een filter moet worden ingesteld, dan is index 0: geen voorkeur, 1: Tekst bestanden *.txt, 2: Pascal bestanden, etc. Om de `OpenDialog` met Pascal bestanden moet worden geopend, moet de "FilterIndex" op 2 gezet worden.

De `InitialDir` geeft de Map aan, waar de `OpenDialog` mee start.

```
DlOpen.InitialDir:= 'e:\';
```

zal de `OpenDialog` in de root van drive E laten starten.

Een andere elegante oplossing is om de `OpenDialog` in de "current directory" op te starten. Dit kan eenvoudig door de volgende code toe te voegen aan de `MnF1Open` procedure

```
DlOpen.InitialDir:= GetCurrentDir;
```

Er zijn een groot aantal opties mogelijk in de Eigenschap "Options". Het veranderen van map kan voorkomen worden, alleen `ReadOnly` bestanden worden getoond of niet getoond, in totaal een 24 opties, die kunnen worden ingesteld.

10 Tenslotte kan de titel eenmalig worden ingesteld of in het programma, afhankelijk van de dan geldende context worden aangepast, zoals met de meeste Eigenschappen binnen Lazarus.

SaveDialog

De SaveDialog biedt de mogelijkheid om bestanden op te slaan. Aan het menu-item "Save" koppelen we in de OnClickEvent de volgende code:

```
procedure TForm1.MnFlSaveClick(Sender: TObject);
begin
  ① DlSave.Title:= modTxt + 'Sla bestand op.';
  ② Sb.Panels[1].Text:= '';
  ③ If DlSave.Execute = true
    then
      begin
        ④ Sb.Panels[1].Text:= DlSave.FileName;
        ⑤ MeData.Lines.SaveToFile(DlSave.FileName);
      end;
end;
```

① De titel van de SaveDialog box wordt gezet op:

```
'Les 10 Dialogs. Sla bestand op. '
```

② Verder maken we de tekst in `Sb.Panels[0]` schoon.

③ We voeren nu de OpenFileDialog uit. Als we een bestand hebben gekozen.

④ zetten we de bestandsnaam in `Sb.Panels[0]` en

⑤ verder schrijven de inhoud van MeData terug naar de harddisk.

10

Het is zaak om op te passen met overschrijven van al bestaande bestanden. Daarom is het zeer aan te raden de optie

```
ofOverwritePrompt
```

"aan" te zetten. Als dan een bestaand bestand overschreven gaat worden, komt er een waarschuwing, die om bevestiging vraagt bij wegschrijven.

SelectDirectoryDialog

Verder is er nog de mogelijkheid om een map te kiezen en die te gebruiken in de Dialogs. De OnClick procedure van `DlSelDir` laat een voorbeeld zien.

```
procedure TForm1.MnFlSelDirClick(Sender: TObject);
var
  s: string;
begin
  DlSelDir.Title:= modTxt + 'Select map.';
  if DlSelDir.Execute = true then
  begin
    DlOpen.InitialDir:= DlSelDir.FileName;
    DlSave.InitialDir:= DlSelDir.FileName;
    s:= 'Initial dirs van DlOpen en DlSave zijn aangepast.'
      + #13 + #10;
    s:= s + 'Probeer DlOpen en DlSave maar om te checken';
    ShowMessage(s);
  end;
end;
```

De procedure is gelijk aan die van de twee andere dialogen. Er wordt een pad naar een map geselecteerd en dit pad wordt gekopieerd in de eigenschap `InitialDir` van de `DlOpen` en `DlSave`.

10 Voeg de volgende berichten toe aan het const blok:

```
mes01 = 'Selecteer bestand';      //proc: MnFlOpenClick
mes02 = 'Sla bestand op';        //proc: MnFlSaveClick
mes03 = 'Selecteer map';        //proc: MnFlSelDirClick
mes04 = 'Start dirs voor DlOpen en DlSave zijn aangepast';
mes05 = 'Probeer DlOpen en DlSave om te controleren.';
```

ColorDialog, FontDialog.

Laten we in MainMenu1 een nieuw menu-item maken: **MnColor** met de Caption Kleur/Font. Hierin maken we weer een submenu aan en voegen daar **MnClColor** met de Caption "Kleur" en **MnClFont** met Caption "Font" aan toe.



We dubbelklikken in de tab Dialogs op ColorDialog en op FontDialog om deze twee componenten toe te voegen aan Form. ColorDialog noemen we **DlKleur** en FontDialog noemen we **DlFont**. Voeg een **OnClickEvent** toe aan **MnClKleur**:

```
procedure TForm1.MnClKleurClick(Sender: TObject);
begin
    DlKleur.Title:= modTxt + 'Select een kleur';
    if DlKleur.Execute = true then
        MeData.Color:= DlKleur.Color;
end;
```

We zetten de titel van de Kleurdialog en starten **DlKleur**. Als er een kleur is gekozen (resultaat = "True"), kan de nieuwe kleur gebruikt worden om **MeData** hiervan te voorzien. Hetzelfde gebeurt met de Fontdialog. Dubbelklik op menu-item Font (in Kleur/Font). Lazarus zal dan de procedure **MnClFontClick** aanmaken. En hier voegen we de volgende code toe:

```
procedure TForm1.MnClFontClick(Sender: TObject);
begin
    If DlFont.Execute = true then
        MeData.Font:= DlFont.Font;
end;
```

Er zijn wat opties te zetten: Alleen true type of alleen fixed type, e.d.

We kunnen ook een aparte dialoog voor de letterkleur maken, maar die zit al in de font dialoog zelf.

10