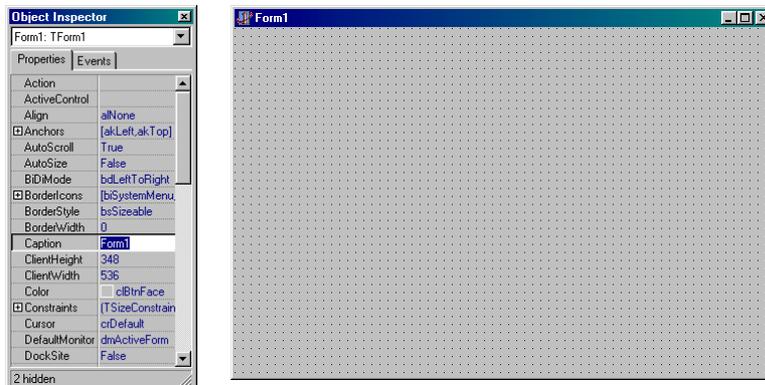


## II. Az alkalmazások ablaka

### Új projekt

*Forms* modul - *TForm* osztály- egy új osztály *TForm1* - példánya a *Form1*.



DFM állomány - *View as Text- View as Form*

DFM adatok szövegesen / binárisan (*Text DFM* menüpont)

```
object Form1: TForm1
  Left = 192
  Top = 107
  Width = 544
  Height = 375
  Caption = 'Form1'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  PixelsPerInch = 96
  TextHeight = 13
end
```

## Az ablakok eseményeinek kezelése

*TNotifyEvent*,

```
type TNotifyEvent = procedure (Sender: TObject) of object;
```

*OnClick*,

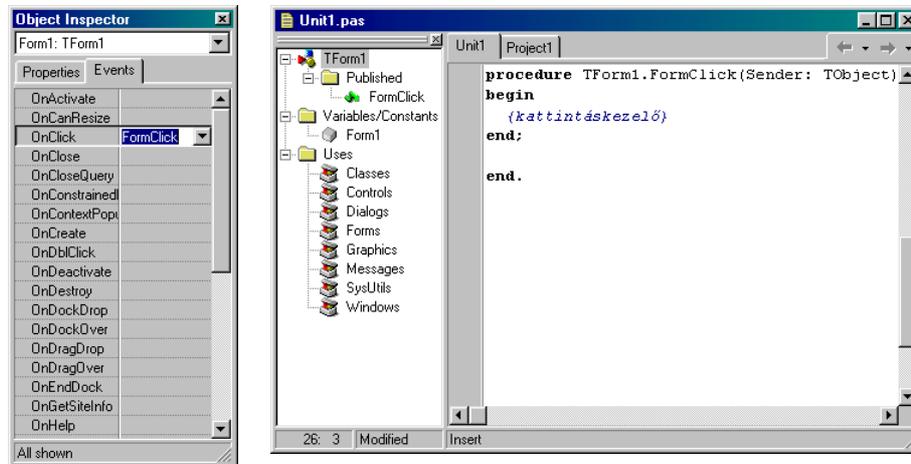
```
property OnClick: TNotifyEvent;
```

Az *interface* rész.

```
type
  TForm1 = class(TForm)
    procedure FormClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

## Az implementation rész

```
procedure TForm1.FormClick(Sender: TObject);  
begin  
    {kattintáskezelő}  
end;
```



## Az ablak létrehozása, zárása, lebontása

A *TForm* alapeseeménye az *OnCreate* (*TNotifyEvent*)  
létrehozáskor nincs még készen az ablak,  
*Project|Options...* dialógus *Forms* lapja  
(*Auto-create form*) / (*Available form*)

*OnDestroy* (*TNotifyEvent* típus).

*OnClose* (*TCloseEvent*)

```
type  
    TCloseAction = (caNone, caHide, caFree, caMinimize);  
    TCloseEvent = procedure(Sender: TObject;  
        var Action: TCloseAction) of object;  
property OnClose: TCloseEvent;
```

*OnClose - OnCloseQuery*

```
type TCloseQueryEvent = procedure(Sender: TObject;  
    var CanClose: Boolean) of object;  
property OnCloseQuery: TCloseQueryEvent;
```

## Az ablak megjelenése és aktiválása

*OnShow*

*OnHide*

*OnActivate*,

*OnDeactivate*

*OnResize*

*OnCanResize*

*OnPaint*,

## Billentyű és egérkezelés az ablakon

*OnKeyDown*,

*OnKeyUp*

```
type TKeyEvent = procedure (Sender: TObject; var Key: Word;  
    Shift: TShiftState) of object;
```

```
type TShiftState = set of (ssShift, ssAlt, ssCtrl, ssLeft,  
    ssRight, ssMiddle, ssDouble);
```

### ***OnKeyPress***

```
type TKeyPressEvent = procedure (Sender: TObject;  
                                var Key: Char) of object;
```

### ***OnShortCut (TShortCutEvent)***

### ***OnClick (TNotifyEvent)***

### ***OnDblClick, (TNotifyEvent)***

### ***OnMouseDown (TMouseEvent)***

### ***OnMouseUp (TMouseEvent)***

```
type  
  TMouseEvent = procedure (Sender: TObject;  
                          Button: TMouseButton; Shift: TShiftState;  
                          X, Y: Integer) of object;
```

```
property OnMouseDown: TMouseEvent;
```

```
type TMouseButton = (mbLeft, mbRight, mbMiddle);
```

### ***OnMouseMove (TMouseMoveEvent)***

```
type TMouseMoveEvent = procedure (Sender: TObject;  
                                 Shift: TShiftState; X, Y: Integer) of object;
```

### **Egyéb ablakesemények**

***OnHelp, OnDragDrop, OnDragOver OnDockDrop, OnDockOver OnMouseWheel.***

## **Az Ablakok tulajdonságai**

### ***Name string***

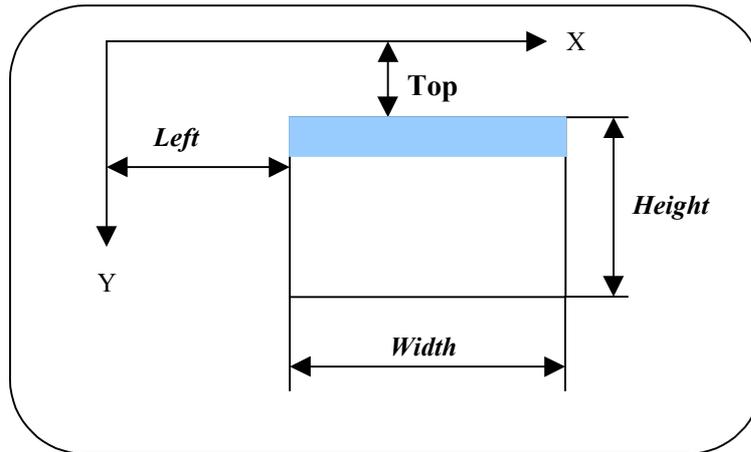
```
Unit1 in 'Unit1.pas' {Ablak};
```

```
Application.CreateForm(TAblak, Ablak);
```

```
type  
  TAblak = class(TForm)  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;
```

```
var  
  Ablak: TAblak;
```

## Ablakok elhelyezkedése



### **Position - TPosition**

*poDesigned, poScreenCenter, poDesktopCenter  
poMainFormCenter, poOwnerFormCenter*

### **WindowState**

```
type TWindowState = (wsNormal, wsMinimized, wsMaximized);
```

### **Align**

```
type TAlignSet = set of TAlign;  
type TAlign = (alNone, alTop, alBottom, alLeft,  
alRight, alClient);
```

### **AutoSize**

**ClientWidth**

**ClientHeight**

**ClientOrigin**

**BorderWidth.**

## Az ablakok „működése”

**Enabled**

**Active**

**KeyPreview**

**Hint**

**ShowHint**

**AutoScroll**

## Az ablakok megjelenése

**Formstyle**

**TFormStyle** - *fsNormal, fsStayOnTop*

**Visible** (boolean)

**Showing** (boolean)

**Caption** (string)

**BorderIcons**

```
type TBorderIcon = (biSystemMenu, biMinimize,  
biMaximize, biHelp);  
TBorderIcons = set of TBorderIcon;
```

## **BorderStyle**

```
type
  TFormBorderStyle = (bsNone, bsSingle, bsSizeable,
    bsDialog, bsToolWindow, bsSizeToolWin);
```

## **Icon TIcon**

### **Cursor - TCursor**

```
type TCursor = -32768..32767;
```

<i>crArrow</i>	2	
<i>crCross</i>	3	
<i>crIBeam</i>	4	
<i>crHourGlass</i>	11	
<i>crHelp</i>	20	
<i>crHandPoint</i>	21	

## **Canvas**

### **Font**

#### **Color**

## **Ablakkezelő metódusok**

### **Metódusok az ablakok létrehozására és lebontására**

#### **(Application.CreateForm)**

##### **Create**

```
constructor Create(AOwner: TComponent); override;
```

##### **CreateNew DFM nélkül**

```
constructor CreateNew(AOwner: TComponent;
  Dummy: Integer = 0); virtual;
```

##### **Close – CloseQuery**

### **Ablak megjelenítő metódusok**

#### **Hide**

#### **Show (ShowModal).**

#### **SetBounds**

**Invalidate** érvénytelenné teszi az ablak területét

**Update, Refresh** újrafestés azonnal.

**Repaint = Invalidate + Update**

#### **ScaleBy**

#### **ScrollBy**

### **Egyéb ablakkezelő metódusok**

#### **Print**

#### **Broadcast**

```
procedure Broadcast(var Message);
```