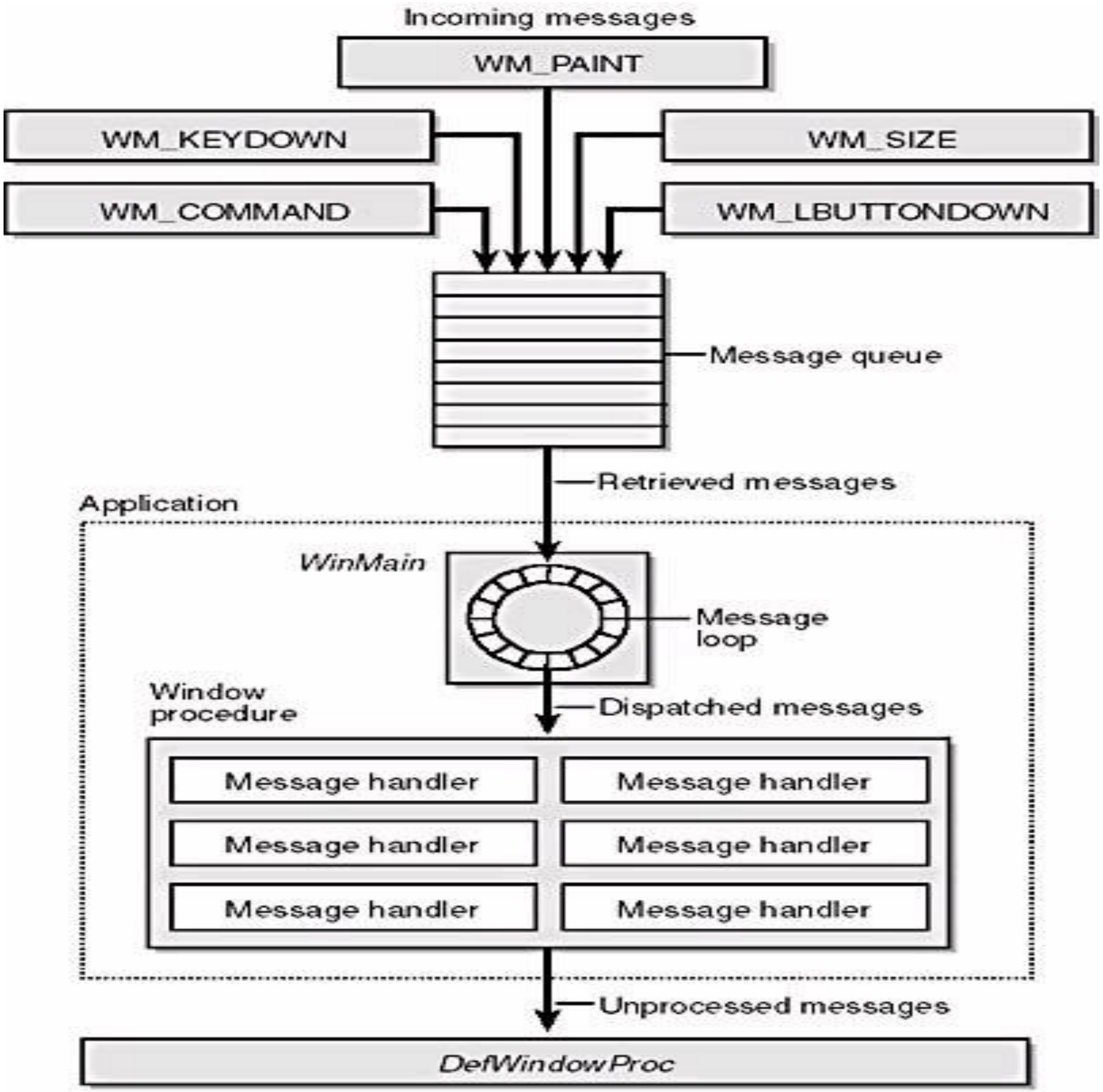
Windows SDK


Incoming messages
WM_PAINT
WM_KEYDOWN
WM_SIZE
WM_COMMAND
WM_LBUTTONDOWN
Message queue
Retrieved messages
Application
WinMain
Message loop
Window procedure
Dispatched messages
Message handler
Message handler
Message handler
Message handler
Message handler
Message handler
Unprocessed messages
DefWindowProc

Forma vizuala a unei aplicatii Windows

1. Bara de titlu
2. Bara de meniu
3. Toolbar
4. Meniul sistem
5. Barele de navigare (derulare) verticala si orizontala
6. Bara de stare
7. Zona client

Windows

- Evenimentele genereaza mesaje.
- Coada de mesaje:
  - Sistem.
  - Pentru fiecare fir cu interfata.

Windows

- ❖ O schema simplificata a acestui mecanism este:
- ✓ Windows receptioneaza evenimentul (actiunea utilizatorului, scurgerea unui interval de timp, etc)
- ✓ Windows transforma actiunea în mesaj.
- ✓ Fereastra programului primeste mesajul.
- ✓ Programul executa o bucla care preia si distribuie mesajele.

Bucla de mesaje

```
MSG msg;  // structura ce descrie mesajul
while( GetMessage( &msg, NULL, 0, 0 ) )
{
   TranslateMessage( &msg );
   DispatchMessage( &msg );
}
```

Observatie. Nu exista un asemenea cod in WinForms.

Procedura fereastra

Apelata de sistemul de operare.

```
LRESULT CALLBACK WndProc(HWND hwnd, UINT uMsg,
    WPARAM wParam, LPARAM lParam)
{ switch(uMsg) {
      case WM_PAINT:
            Deseneaza(hwnd);
            break;
      case WM_DESTROY:
            PostQuitMessage(0);
            break;
      … } }
```

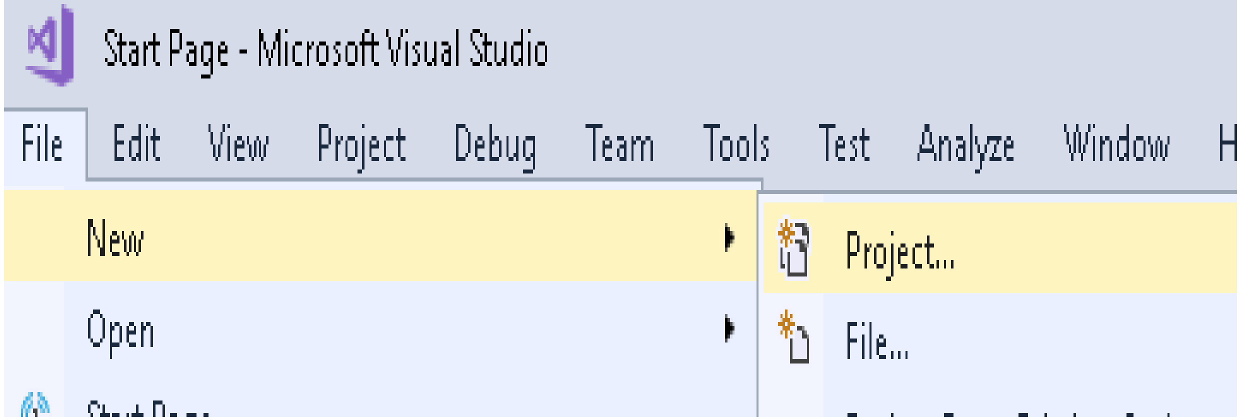
Etape in creare si lansare app GUI

- Inregistrare fereastra (structura).
- Stabilire atribute pentru fereastra.
- Desenarea ferestrei.
- Bucla de mesaje (am vazut codul mai sus).

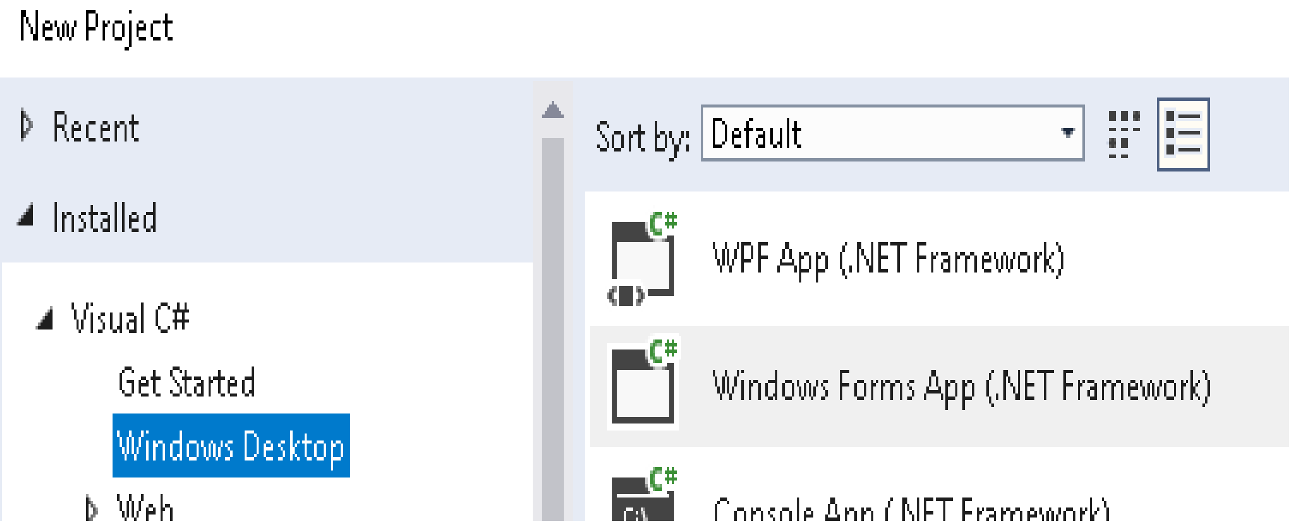
.NET - Windows Form

- WinForms este un “layer” peste controalele standard din Windows (TextBox, Button, etc.).
- Consideratii generale WinForms.
  - Are o anumita maturitate, testat foarte mult timp.
  - Controale construite de terti.
  - Foarte usor de utilizat.

Hello WinForms

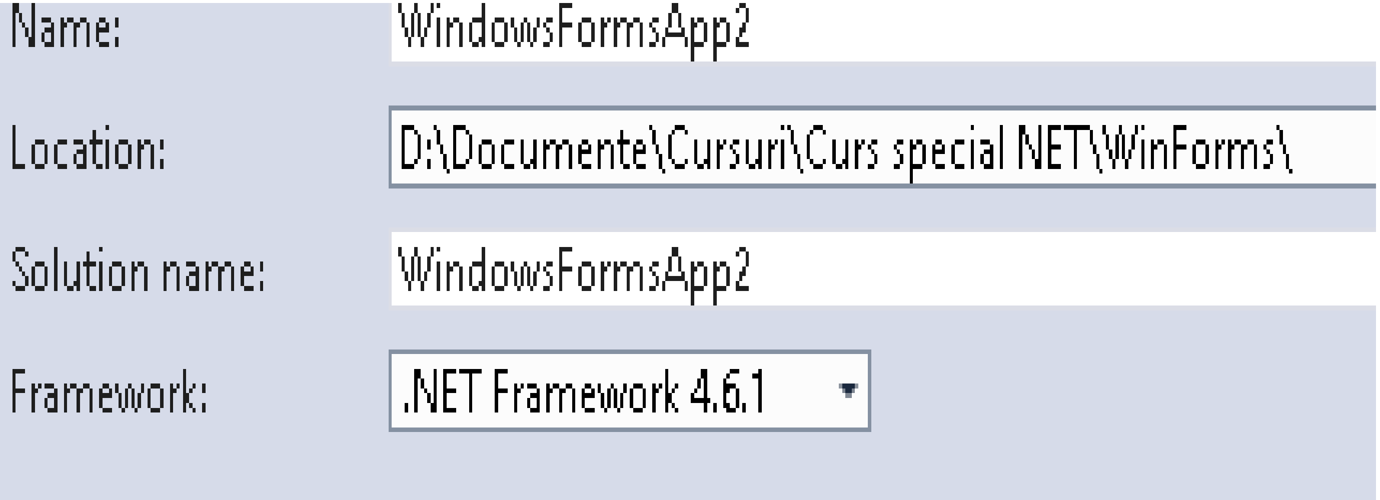


Tip proiect



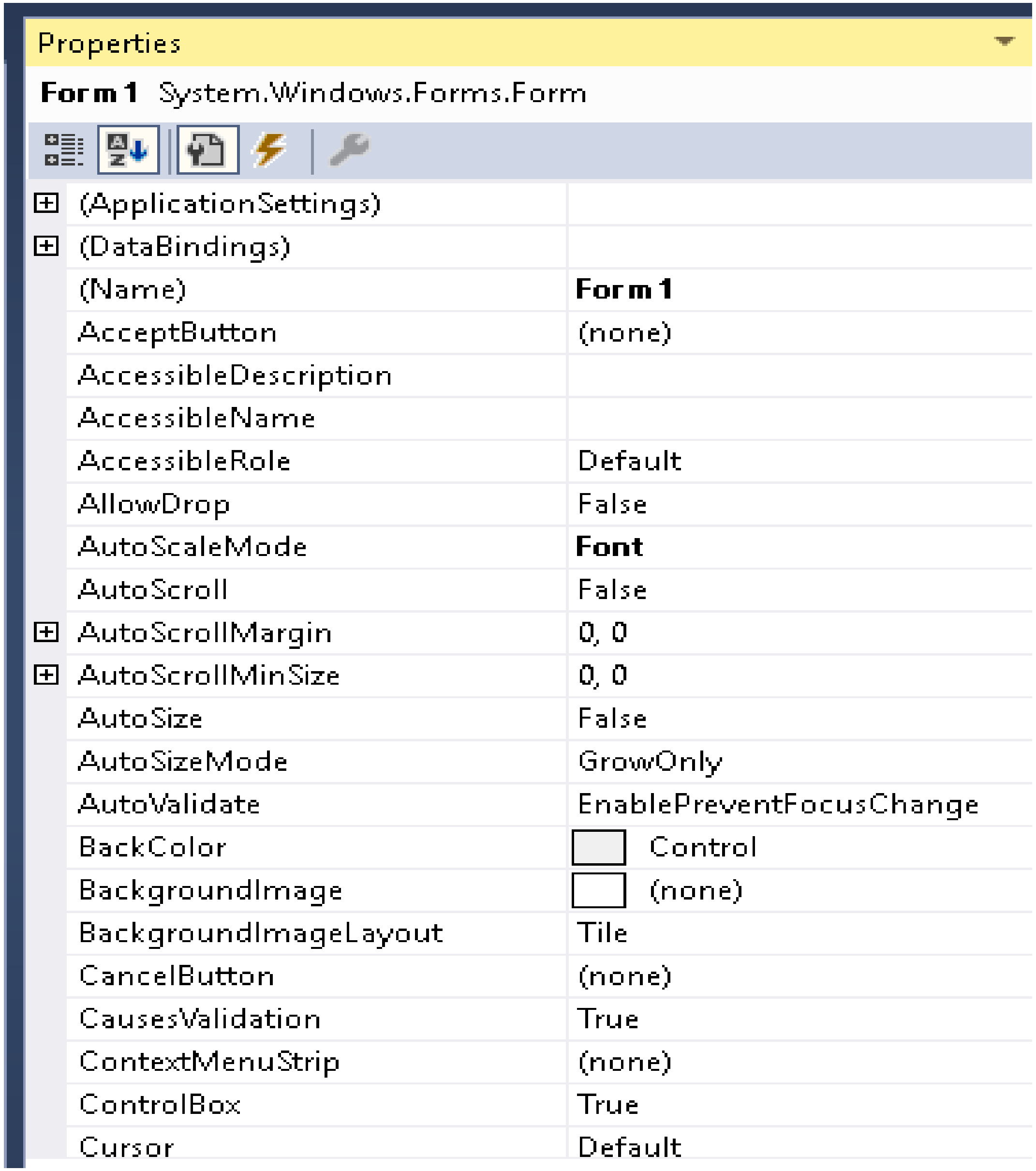
Sort by: Default

Nume proiect



Name:	WindowsFormsApp2
Location:	D:\Documente\Cursuri\Curs special NET\WinForms\
Solution name:	WindowsFormsApp2
Framework:	.NET Framework 4.6.1

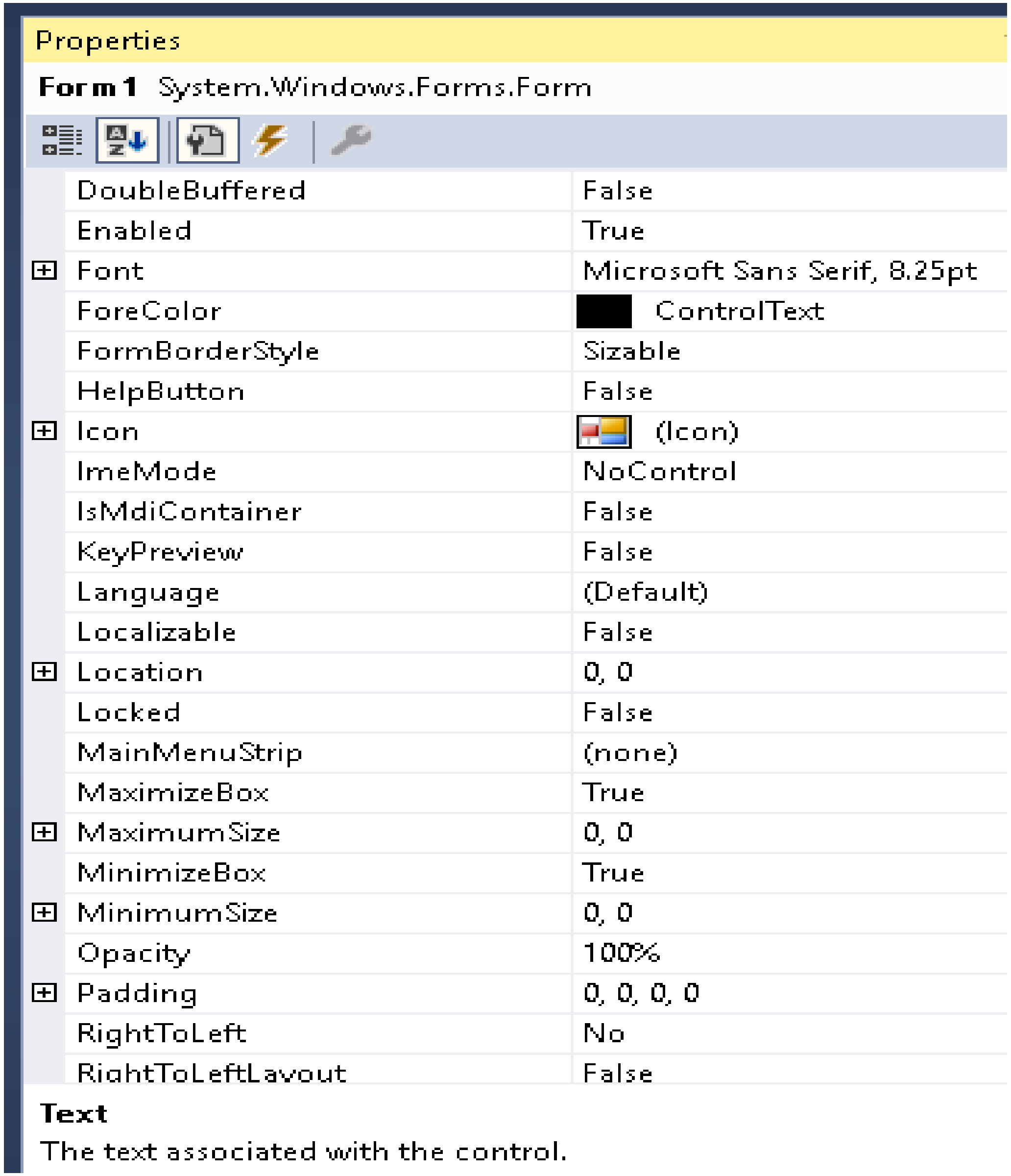
Hello!



Properties

Form1 System.Windows.Forms.Form

⊞	(ApplicationSettings)	
⊞	(DataBindings)	
	(Name)	**Form1**
	AcceptButton	(none)
	AccessibleDescription	
	AccessibleName	
	AccessibleRole	Default
	AllowDrop	False
	AutoScaleMode	**Font**
	AutoScroll	False
⊞	AutoScrollMargin	0, 0
⊞	AutoScrollMinSize	0, 0
	AutoSize	False
	AutoSizeMode	GrowOnly
	AutoValidate	EnablePreventFocusChange
	BackColor	▨ Control
	BackgroundImage	□ (none)
	BackgroundImageLayout	Tile
	CancelButton	(none)
	CausesValidation	True
	ContextMenuStrip	(none)
	ControlBox	True
	Cursor	Default



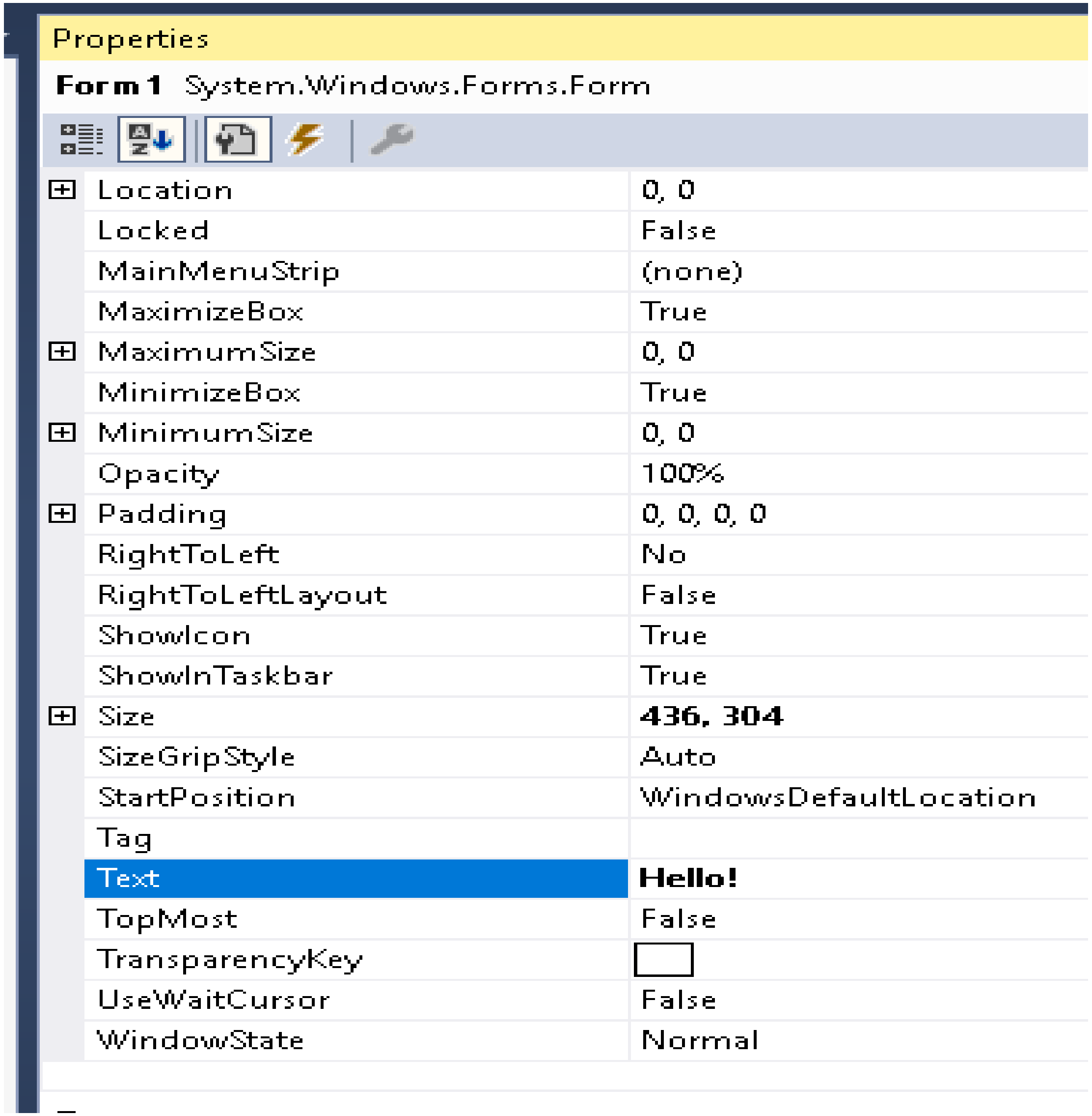
Properties

Form1 System.Windows.Forms.Form

DoubleBuffered	False
Enabled	True
⊞ Font	Microsoft Sans Serif, 8.25pt
ForeColor	■ ControlText
FormBorderStyle	Sizable
HelpButton	False
⊞ Icon	(Icon)
ImeMode	NoControl
IsMdiContainer	False
KeyPreview	False
Language	(Default)
Localizable	False
⊞ Location	0, 0
Locked	False
MainMenuStrip	(none)
MaximizeBox	True
⊞ MaximumSize	0, 0
MinimizeBox	True
⊞ MinimumSize	0, 0
Opacity	100%
⊞ Padding	0, 0, 0, 0
RightToLeft	No
RightToLeftLayout	False

Text

The text associated with the control.



Properties

Form1 System.Windows.Forms.Form

⊞ Location	0, 0
Locked	False
MainMenuStrip	(none)
MaximizeBox	True
⊞ MaximumSize	0, 0
MinimizeBox	True
⊞ MinimumSize	0, 0
Opacity	100%
⊞ Padding	0, 0, 0, 0
RightToLeft	No
RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
⊞ Size	**436, 304**
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	**Hello!**
TopMost	False
TransparencyKey	⬜
UseWaitCursor	False
WindowState	Normal

Cod generat. Clasa Form1

```
// Aceasta este structura pentru clasa fereastra
using System;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

Entry point

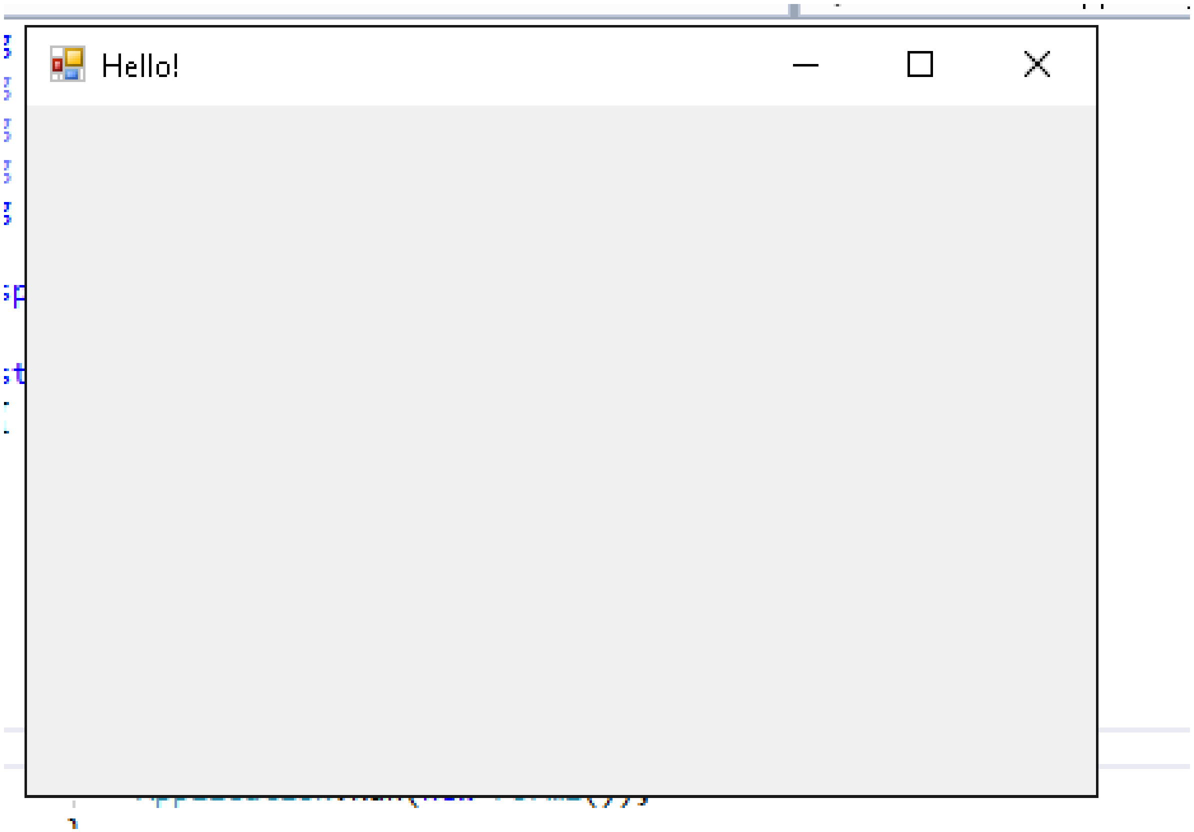
```
using System;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

EnableVisualStyles()

- **Remarks**
- This method enables visual styles for the application. Visual styles are the colors, fonts, and other visual elements that form an operating system theme. Controls will draw with visual styles if the control and the operating system support it. To have an effect, EnableVisualStyles() must be called before creating any controls in the application; typically, EnableVisualStyles() is the first line in the Main function.

Rezultat executie



Desenare in zona client (fereastra)

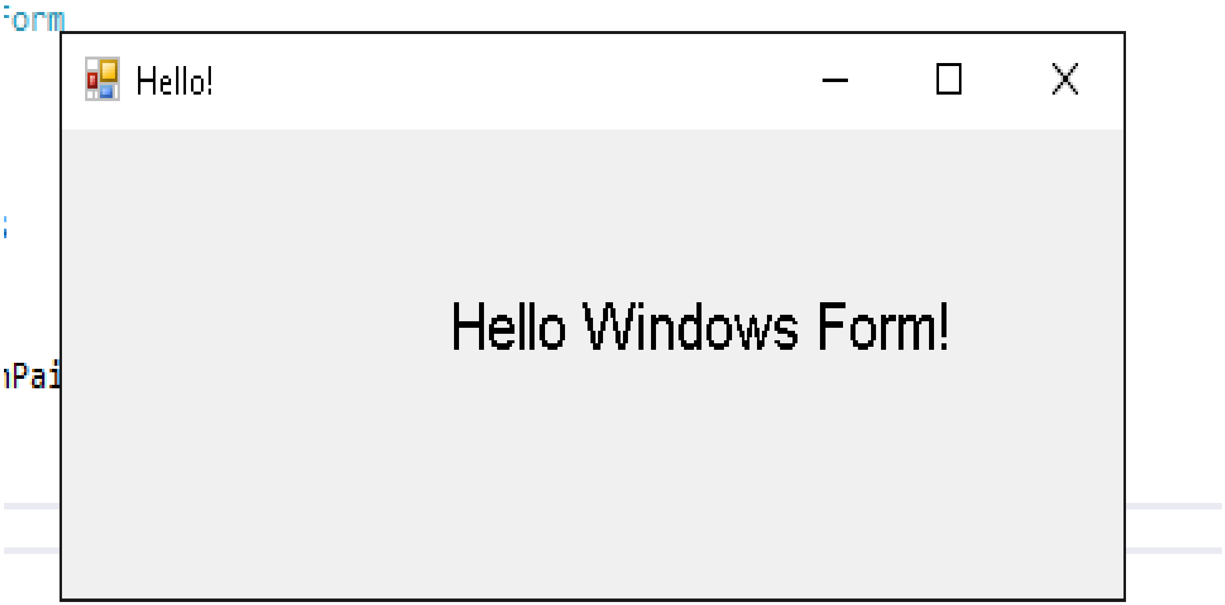
- Trebuie sa suprascriem functia OnPaint().

```
protected override void OnPaint(
                              PaintEventArgs e)
        {
            base.OnPaint(e);
            DrawString();
        }
```

DrawString()
Spatiu de nume: System.Drawing

```
private void DrawString()
    {
      Graphics formGraphics = this.CreateGraphics();
      string drawString = "Hello Windows Form!";
      Font drawFont = new System.Drawing.Font("Arial", 16);
      SolidBrush drawBrush = new   SolidBrush(Color.Black);
      float x = 150.0F;
      float y = 50.0F;
      StringFormat drawFormat = new StringFormat();
      formGraphics.DrawString(drawString, drawFont, drawBrush, x, y, drawFormat);
      drawFont.Dispose();
      drawBrush.Dispose();
      formGraphics.Dispose();
    }
```

Executie



Hello!

Hello Windows Form!

Graphics

- Before you can draw lines and shapes, render text, or display and manipulate images with GDI+, you need to create a Graphics object.
  The Graphics object represents a GDI+ drawing surface, and is the object that is used to create graphical images.
- There are two steps in working with graphics:
  - Creating a Graphics object.
  - Using the Graphics object to draw lines and shapes, render text, or display and manipulate images.

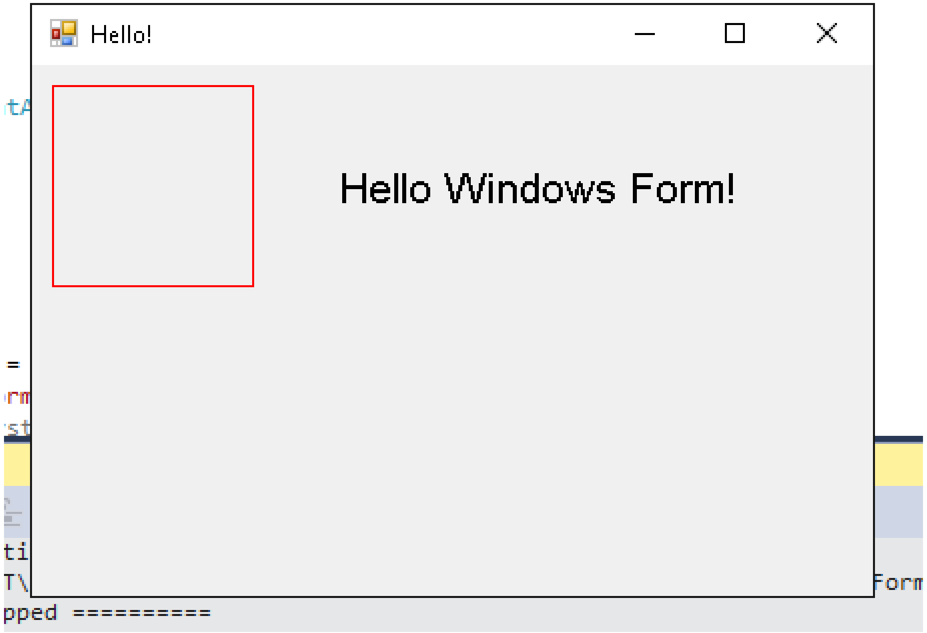
Graphics object

```
// Handler event Paint
private void Form1_Paint(object sender,
   PaintEventArgs pe)
{ // Declares the Graphics object and sets
  // it to the Graphics object supplied in the
  // PaintEventArgs.
Graphics g = pe.Graphics;
// Insert code to paint the form here. }
```

Desenam un patrat

```
// Tratam evenimentul Paint al ferestrei.
private void Form1_Paint(
                    object sender, PaintEventArgs e)
        {
            g = e.Graphics;
            g.DrawRectangle(new Pen(Color.Red),
                     new Rectangle(10, 10, 100, 100));
        }
```

Executie



Hello Windows Form!

Graphics object

```
Graphics g;
 // Sets g to a graphics object representing
 // the drawing surface of the control or form
 //  g is a member of.
g = this.CreateGraphics();
```

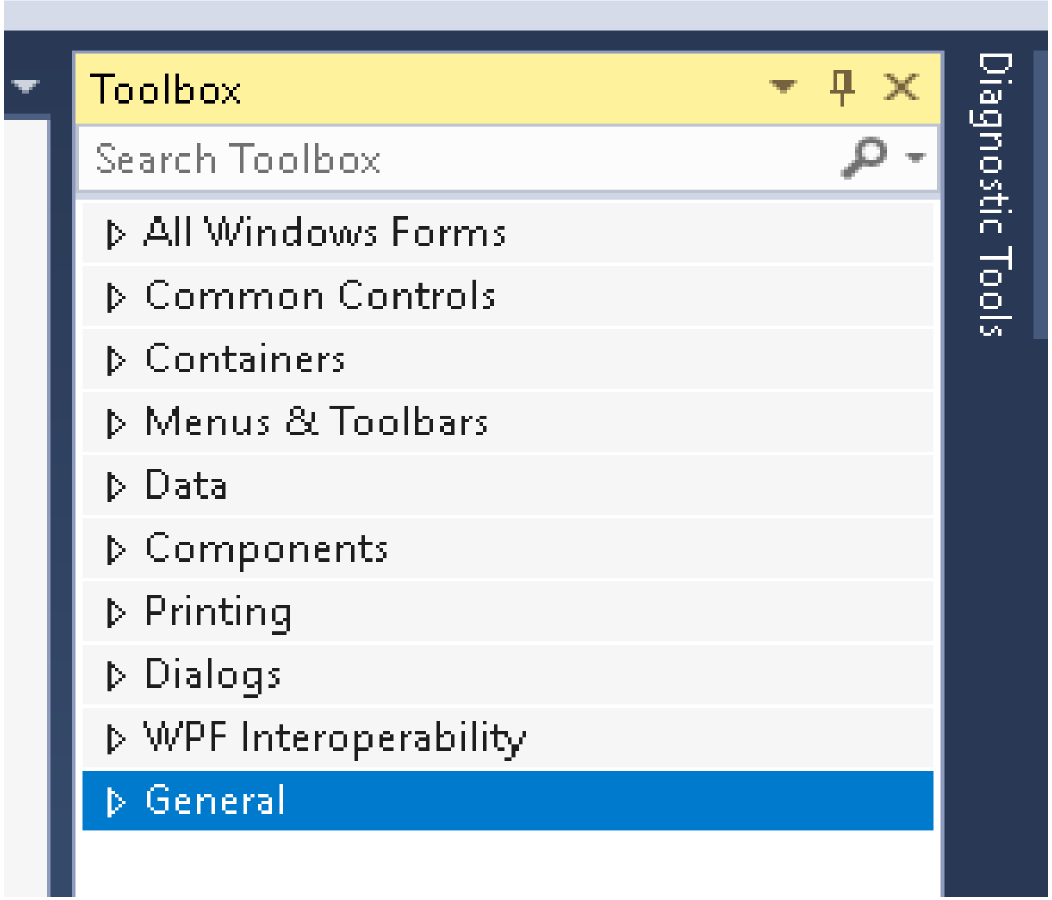
Graphics

- Consultati MSDN pentru a vedea toate facilitatile oferite de Graphics.

Controale

- O lista a controalelor ce pot fi folosite este afisata in fereastra Toolbox din VS.

Controale


Toolbox
Search Toolbox
All Windows Forms
Common Controls
Containers
Menus & Toolbars
Data
Components
Printing
Dialogs
WPF Interoperability
General
Diagnostic Tools

Controale comune

- Des utilizate.
- Afiseaza text sau preiau text.
  - Label, TextBox, RichTextBox.
- Butoane:
  - Button, CheckBox, RadioButton.
- Contin o colectie de articole:
  - ComboBox, ListBox, ListView, TreeView.

Handler evenimente

- private void Metoda(object sender, EventArgs e);
- Al doilea parametru poate fi si o clasa derivata din EventArgs.