

C# - Cross Threading

Scritto da Administrator

Lunedì 21 Marzo 2011 08:44 - Ultimo aggiornamento Martedì 22 Marzo 2011 17:16

Il problema più ricorrente quando si utilizzano più thread contemporaneamente è quello di comunicare con il Thread principale dove tipicamente risiedono i controlli che visualizzano un determinato risultato delle nostre operazioni.

Se queste operazioni vengono eseguite da un thread parallelo e si tenta per esempio di popolare una listbox del thread principale verrà restituito il seguente errore:

**System.InvalidOperationException was unhandled
Cross-thread operation not valid: Control 'listbox1' accessed from a thread other than the thread it was created on.**

Per ovviare a questo inconveniente è necessario delegare una funzione alla "comunicazione" con i due thread la quale si prenderà in carico di utilizzare la proprietà del componente in oggetto.

Ecco un esempio che utilizza una TextBox:

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data; using System.Threading;

public delegate void InsTextDelegato(string text);

namespace CrrThreadTEST
{
    public class Form1 : System.Windows.Forms.Form
    {
        private System.Windows.Forms.TextBox textBox1;

        [STAThread] static void Main()
        {
            Application.Run(new Form1());
        }

        private void Form1_Load(object sender, System.EventArgs e)
        {
            Thread t = new Thread(new ThreadStart(DoWork));
```

C# - Cross Threading

Scritto da Administrator

Lunedì 21 Marzo 2011 08:44 - Ultimo aggiornamento Martedì 22 Marzo 2011 17:16

```
t.Start();
textBox1.Text = string.Empty;
}

private void InsText(string text)
{
textBox1.Text = text;
}

private void DoWork()
{
Thread.Sleep(3000);
textBox1.Invoke( new InsTextDelegato(InsText), new object[]{"Some Text"}); }
}

}
```

quindi riassumendo:

creo un delegato nel thread principale che comunica il thread secondario il quale passa il parametro alla funzione che utilizzerà effettivamente la proprietà dell'oggetto specificato (in questo caso InsText che all'interno esegue la classica assegnazione).