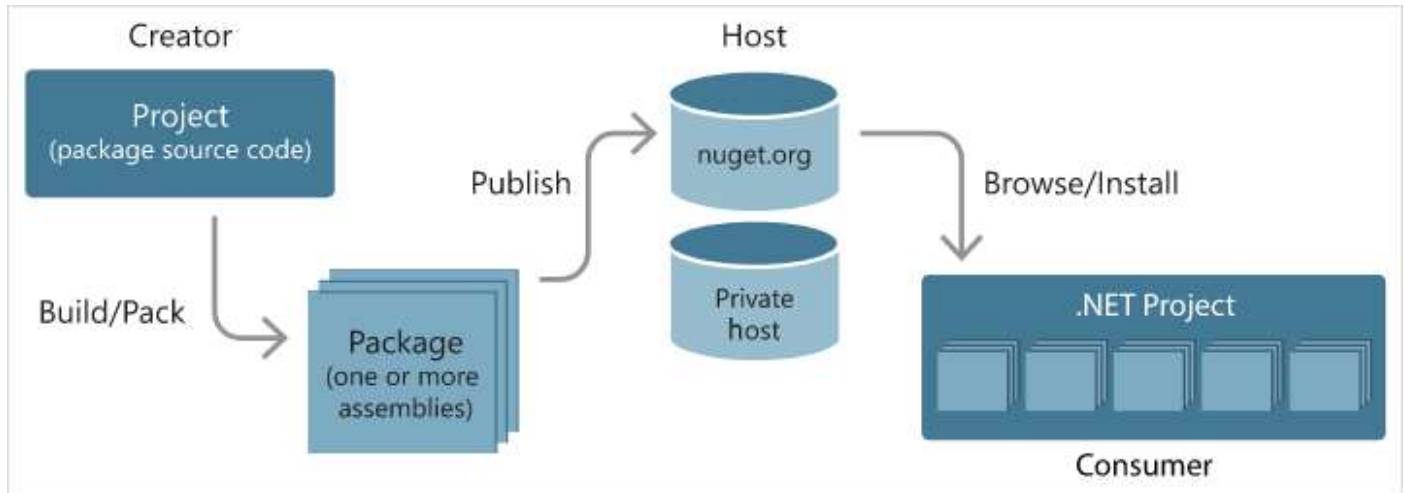
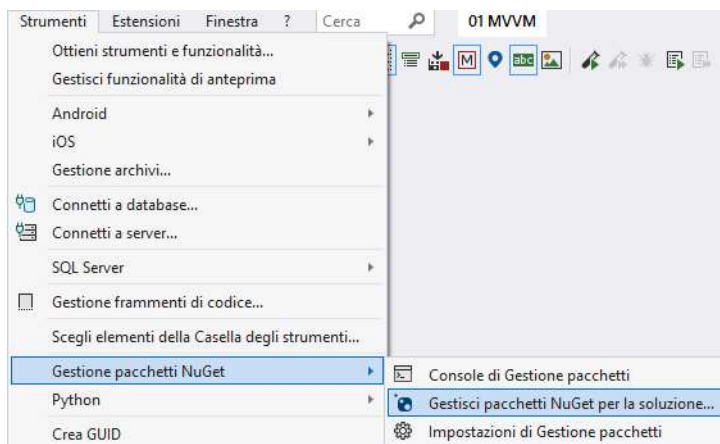


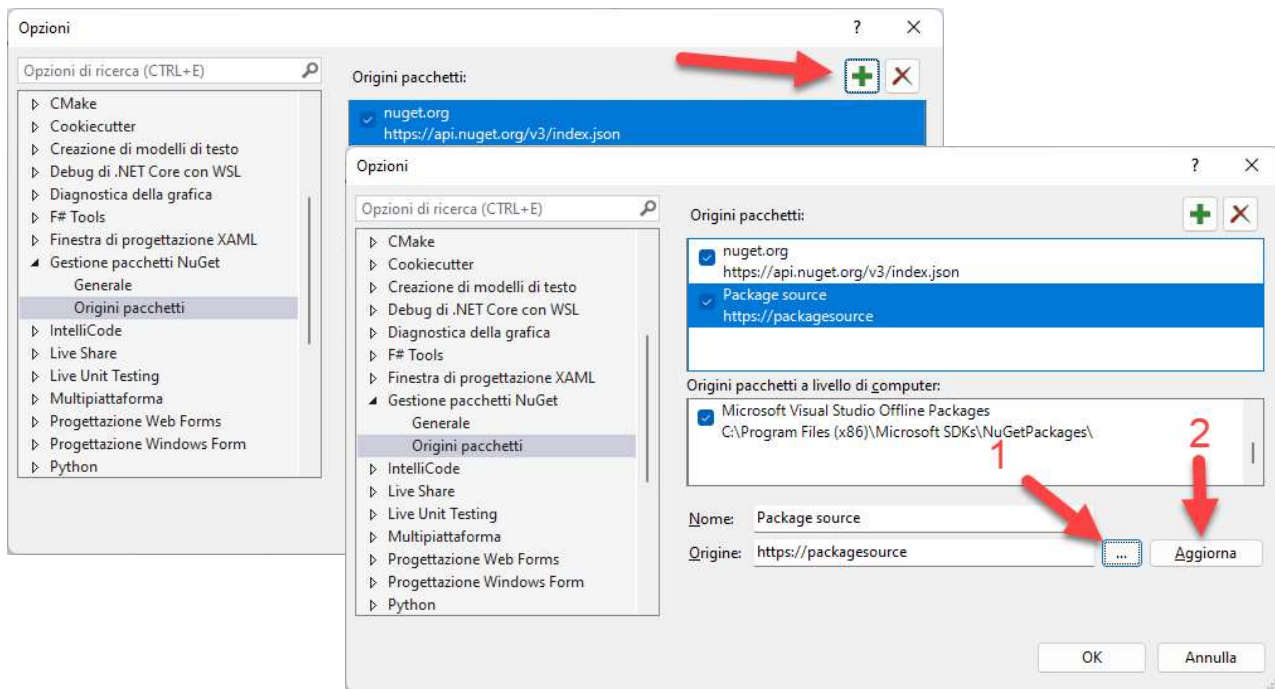
## Aggiunta di NuGet Locale

Uno strumento essenziale per qualsiasi piattaforma di sviluppo moderna è un meccanismo attraverso il quale gli sviluppatori possano creare, condividere e utilizzare codice utile. Spesso questo codice viene incluso in "pacchetti" che contengono codice compilato, ad esempio file DLL, insieme ad altri contenuti necessari nei progetti che utilizzano questi pacchetti. Un host (nuget.org o p.e. una cartella) funge da punto di connessione tra gli autori e i consumer dei pacchetti. Gli autori compilano utili pacchetti NuGet e li pubblicano in un host. I consumer cercano quindi i pacchetti utili e compatibili negli host accessibili, scaricandoli e includendoli nei loro progetti.



Per poter utilizzare questi pacchetti all'interno del CPT (dietro ad un proxy, o quando non è a disposizione la rete) è utile salvare localmente i pacchetti supplementari utilizzati (ovviamente non sono aggiornati).



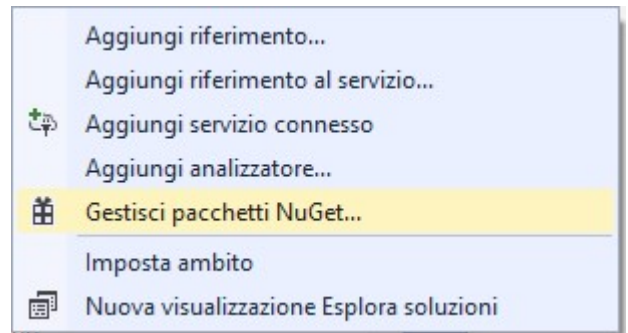


### Aggiunta di MahApps.Metro (2.4.9)

Una delle librerie gratuite che cambia l'aspetto grafico dell'interfaccia è la libreria Mahapps.Metro. Dopo aver copiato i pacchetti nuget (e le dipendenze) nella cartella preposta è possibile scegliere il pacchetto MahApps.Metro. Una volta installato è necessario fare alcune modifiche per ottenere la nuova interfaccia grafica.

#### App.xaml

Qui si deviniscio le risorse da includere e si definisce lo stile (accent) da utilizzare (Light o Dark) e il colore (Theme).



```
<Application.Resources>
  <ResourceDictionary>
    <ResourceDictionary.MergedDictionaries>
      <!-- MahApps.Metro resource dictionaries. Make sure that all file names are Case Sensitive! -->
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Controls.xaml" />
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Fonts.xaml" />
      <!-- Accent and AppTheme setting -->
      <ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Themes/Light.Yellow.xaml" />
    </ResourceDictionary.MergedDictionaries>
  </ResourceDictionary>
</Application.Resources>
```

#### MainWindows.xaml

Prima si deve aggiungere un nuovo namespace.

```
xmlns:controls="http://metro.mahapps.com/winfx/xaml/controls"
```

Poi si modifica il tag iniziale

```
<controls:MetroWindow x:Class="_01_Colore.MainWindow"
```



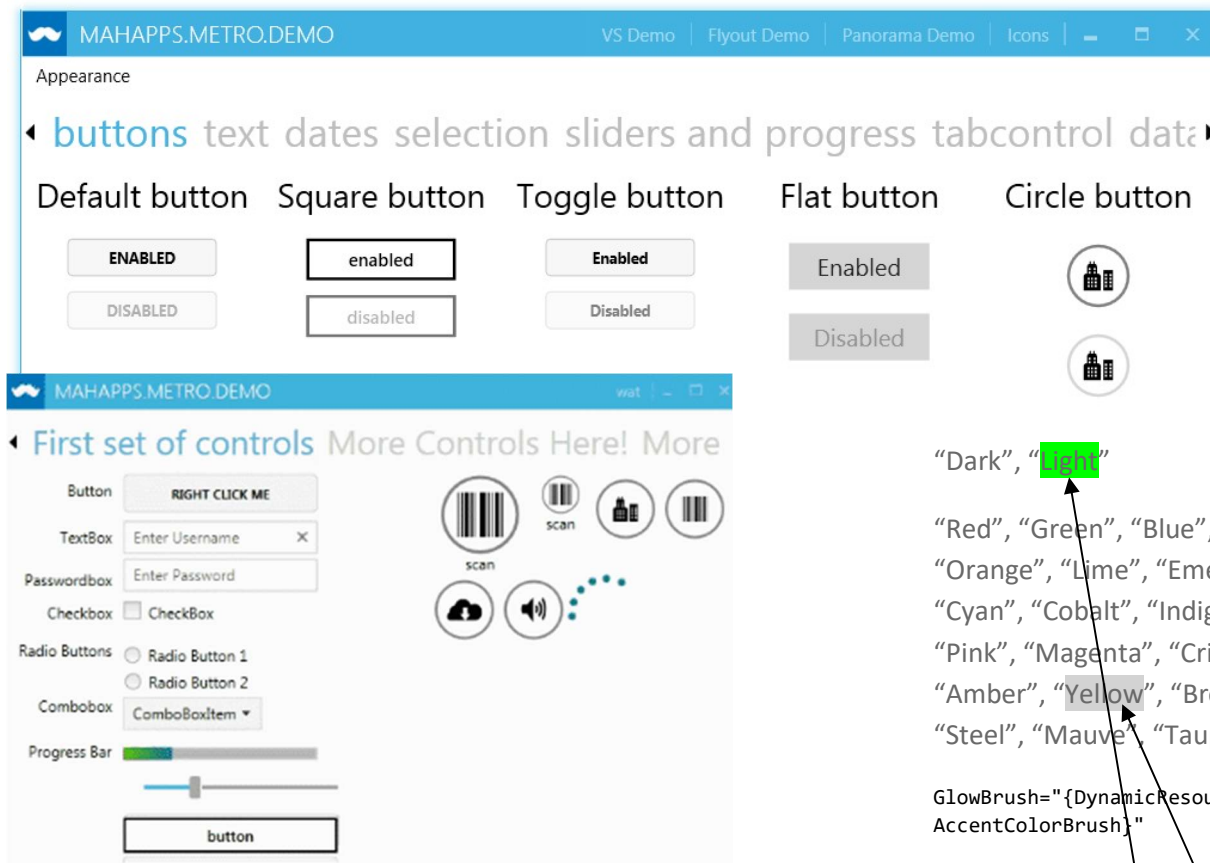
### MainWindows.xaml.cs

Qui si aggiunge uno using e si fa ereditare MainWindow da MetroWindow (Tasto F7)

```
using MahApps.Metro.Controls;

namespace _01_Colore
{
    /// <summary>
    /// Logica di interazione per MainWindow.xaml
    /// </summary>
    public partial class MainWindow : MetroWindow
```

### Esempi



I colori della maschera dipendono da quanto inserito in App.xaml alla riga

```
<ResourceDictionary Source="pack://application:,,,/MahApps.Metro;component/Styles/Themes/Light.Yellow.xaml" />
```

### <Model>View.xaml

Inserire negli UserControl che utilizzeranno i componenti della libreria la riga

```
xmlns:controls="http://metro.mahapps.com/winfx/xaml/controls"
```

### Librerie supplementari

Altre librerie con svariati componenti supplementari, e di più facile utilizzo è quella di ModernWPF (<https://github.com/Kinnara/ModernWpf>) e HandyControls (<https://ghost1372.github.io/handycontrol>).

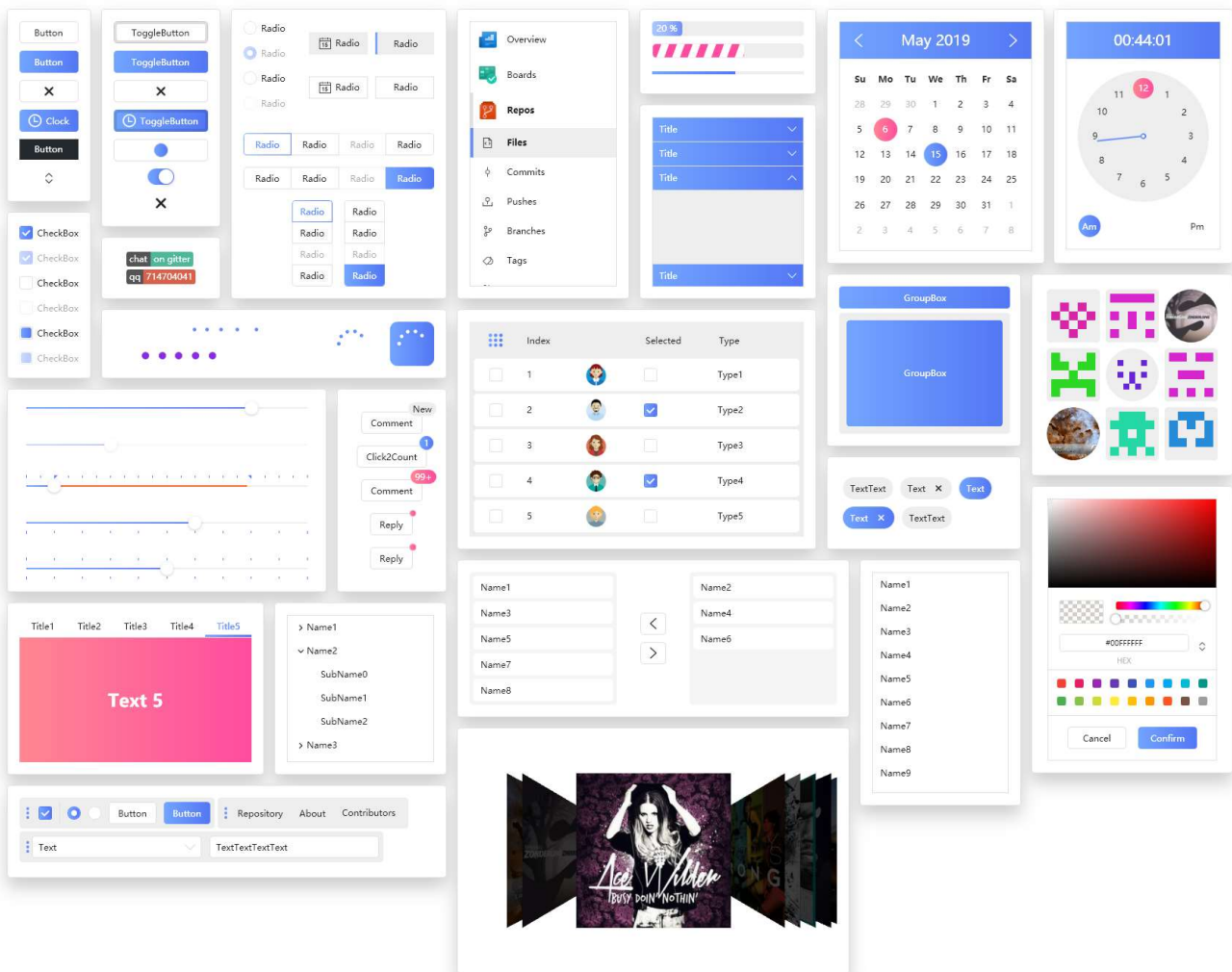
### HandyControls

Questa libreria, dopo la sua installazione presenta un file readme.txt dove viene specificato quali operazioni apportare per il suo utilizzo (App.xaml) . Il namespace da utilizzare è:

```
xmlns:hc="https://handyorg.github.io/handycontrol"
```



## HandyControls



## ModernWPF

