

MaterialMessageBox

MaterialMessageBox works on .Net Core and .Net Framework

How it works?

add it in you project

1. this Library works with MaterialDesign, so you need to install [THIS](#) package from NuGet
2. download the .dll file from the releases ([direct download](#))
3. open your project in VisualStudio and add it as a reference ([how to do it](#))
4. `using CustomMessageBox;`

Documentation

```
//for a custom message
public static MessageBoxResult Show(string messageBoxText);

//for a custom message and custom title
public static MessageBoxResult Show(string messageBoxText, string caption);

//for a custom message, title and MaterialDesign button
public static MessageBoxResult Show(string messageBoxText, string caption, MessageBoxButton button);

//for a custom message, title, MaterialDesign button and a custom icons
public static MessageBoxResult Show(string messageBoxText, string caption, MessageBoxButton button, MessageBoxImage icon);

//for a custom message, title, MaterialDesign button, custom icons and default result when you close the Box
public static MessageBoxResult Show(string messageBoxText, string caption, MessageBoxButton button, MessageBoxImage icon, MessageBoxResult defaultResult);
```

MessageBoxButton

all supported buttons type:

- `MessageBoxButton.OK;`
- `MessageBoxButton.OKCancel;`
- `MessageBoxButton.YesNo;`
- `MessageBoxButton.YesNoCancel;`

MessageBoxIcon

all supported Icons type:

- `MessageBoxImage.Error;`
- `MessageBoxImage.Information;`
- `MessageBoxImage.Question;`
- `MessageBoxImage.Warning;`
- `MessageBoxImage.None;`

MessageBoxResult

all supported Result type:

- `MessageBoxResult.OK;`

- `MessageBoxResult.Cancel;`
- `MessageBoxResult.Yes;`
- `MessageBoxResult.No;`
- `MessageBoxResult.None;`

How use it

`MaterialMessageBox` will use the same Theme you chosed in `App.xaml`

```
<Application.Resources>
  <ResourceDictionary>
    <ResourceDictionary.MergedDictionaries>
      <ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.dark">
      <ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Default">
      <ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Primary/MaterialDesignColor.Primary">
      <ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Accent/MaterialDesignColor.Accent">
    </ResourceDictionary.MergedDictionaries>
  </ResourceDictionary>
</Application.Resources>
```

`MaterialMessageBox` `.Show()` method

some exaples

```
MaterialMessageBox.Show("Some Text", "Costum Title", MessageBoxButton.YesNoCancel, MessageBoxImage.Information);

MaterialMessageBox.Show("Other text", "my title", MessageBoxButton.OKCancel, MessageBoxImage.Warning);

if(MaterialMessageBox.Show(obj.ToString(), "Question", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)
{
    MaterialMessageBox.Show(man.name, "ERROR!", MessageBoxButton.OK, MessageBoxImage.Error);
}

MaterialMessageBox.Show(e.message, "Message", MessageBoxButton.OK, MessageBoxImage.None, MessageBoxResult.OK);
```

costumized Buttons, Box background and Box foreground

color proprieties

```
//for change the Box background color
public static Color backgroundColor;

//for change the Box text color
public static Color foregroundColor;

//for change the background color of any different buttons
public static Color OKButtonColor;
public static Color CancelButtonColor;
public static Color NoButtonColor;
public static Color YesButtonColor;

//for change the text color of any different buttons
public static Color OKButtonForegroundColor;
public static Color CancelButtonForegroundColor;
public static Color NoButtonForegroundColor;
public static Color YesButtonForegroundColor;

//for change the color of the visible icon
public static Color iconColor;
```

color methods

```
//for reset allBox colors
public static void resercolors();

//for reset only one color, this works with all proprieties
MaterialMessageBox.OKButtonColor = new Color();
```

how to set colors

```
//from Hex color
MaterialMessageBox.YesButtonColor = (Color)ColorConverter.ConvertFromString(string HexCode);

//from red green and blue value
MaterialMessageBox.NoButtonColor = Color.FromRgb(byte r, byte g, byte b);
```

example

```
MaterialMessageBox.OKButtonColor = (Color)ColorConverter.ConvertFromString("#00CE52");
MaterialMessageBox.YesButtonColor = (Color)ColorConverter.ConvertFromString("#00CE52");
MaterialMessageBox.NoButtonColor = (Color)ColorConverter.ConvertFromString("#00CE52");
MaterialMessageBox.CancelButtonColor = (Color)ColorConverter.ConvertFromString("#00CE52");
MaterialMessageBox.IconColor = (Color)ColorConverter.ConvertFromString("#00AB44");
MaterialMessageBox.BackgroundColor = (Color)ColorConverter.ConvertFromString("#ffffff");
MaterialMessageBox.ForegroundColor = (Color)ColorConverter.ConvertFromString("#000000");
```

costumize buttons alignment

alignment propertes

```
//set the horizontal alignment of buttons
public static HorizontalAlignment buttonsAlignment;
```

how to use it

```
//set left
MaterialMessageBox.buttonsAlignment = HorizontalAlignment.Left;

//set center
MaterialMessageBox.buttonsAlignment = HorizontalAlignment.Center;

//set right (default value)
MaterialMessageBox.buttonsAlignment = HorizontalAlignment.Right;
```

example