

mahapps.metro

2 min read • [original](#)

Table Of Contents

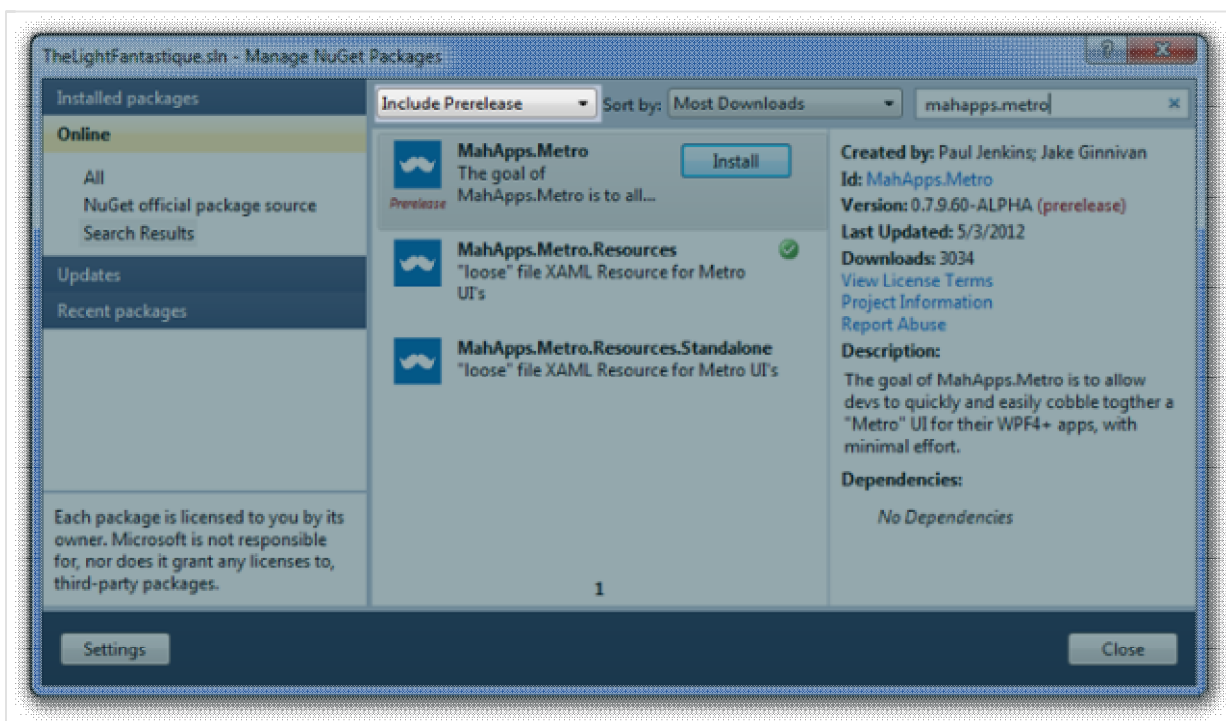
This guide will introduce you to how *MahApps.Metro* works and how to incorporate it into your app.

Installation

You can install MahApps.Metro via the NuGet GUI (right click on your project, click **Manage NuGet Packages**, select **Online** and search for **MahApps.Metro**) or with the Package Manager Console:

```
Install-Package MahApps.Metro
```

If you want to use the pre-release packages of MahApps.Metro (these have the latest code and newest features), you need to enable **Include Prerelease** in the GUI:



or use the Package Manager Console:

```
Install-Package MahApps.Metro -Pre
```

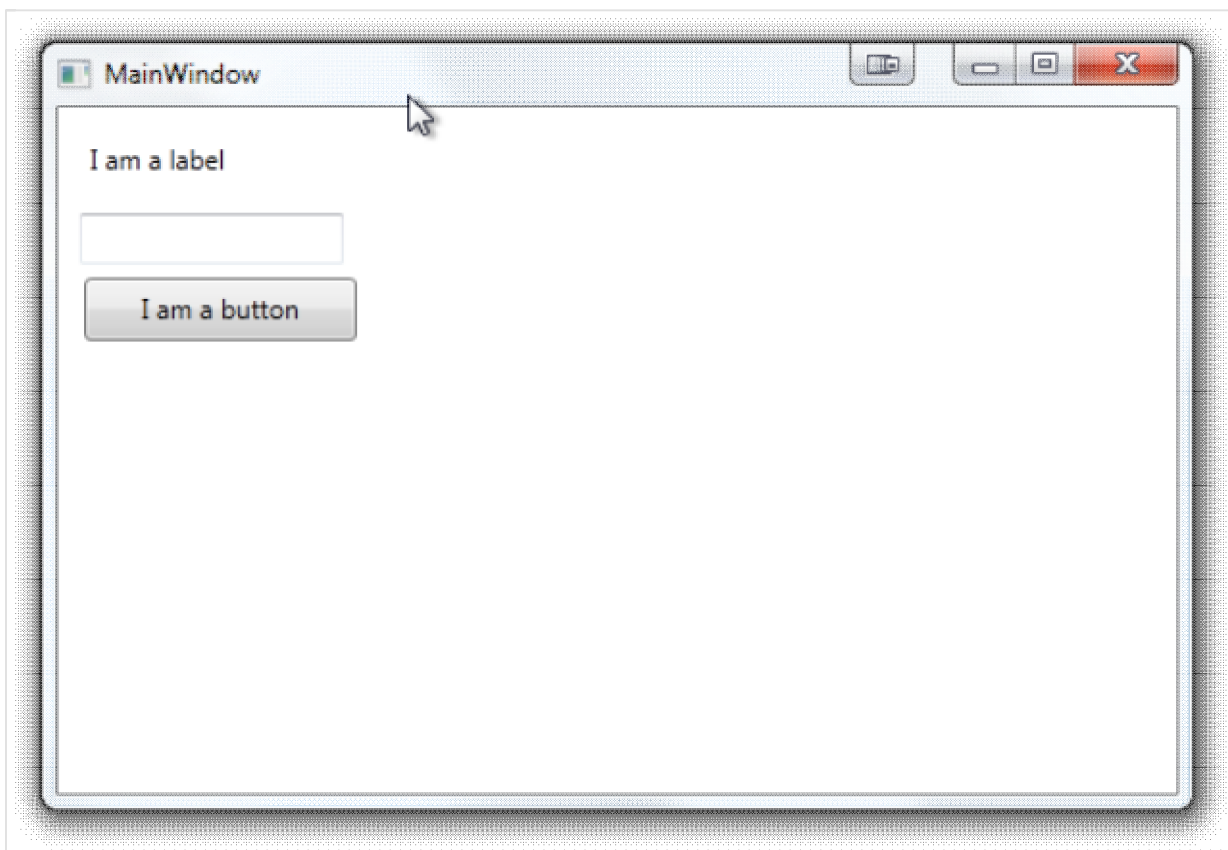
Styling the Window

There's two ways you can style your Window using MahApps.Metro:

- You can use the included `Metrowindow` control or,
- Design your own window

For now we'll use `Metrowindow`, as this approach will work for a good percentage of apps and is the quickest and easiest way to get going. If you want to learn about rolling your own window, check out [the guide](#).

Using the MetroWindow Control



Modifying the XAML file

After installing MahApps.Metro:

- open up `Mainwindow.xaml`
- add this attribute inside the opening Window tag. (It's how you reference other namespaces in XAML):

```
xmlns:controls="clr-namespace:MahApps.Metro.Controls;assembly=MahApps.Metro"
```

- change `<Window ...` tag to `<Controls:Metrowindow ...` (remember to change the closing tag too!)

You should have something like this (don't copy and paste this):

```
<controls:Metrowindow x:Class="wpfApplication2.Mainwindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:controls="clr-namespace:MahApps.Metro.Controls;assembly=MahApps.Metro"
    Title="Mainwindow"
    Height="350"
    Width="525">
<!-- your layout here -->
</controls:Metrowindow>
```

Modifying the CodeBehind File

You'll also need to modify the `Mainwindow.xaml.cs` file so that the base class for `Mainwindow` matches the `Metrowindow` class of the XAML file. To access `Metrowindow`, add the following reference first.

```
// using statements...
using MahApps.Metro.Controls;

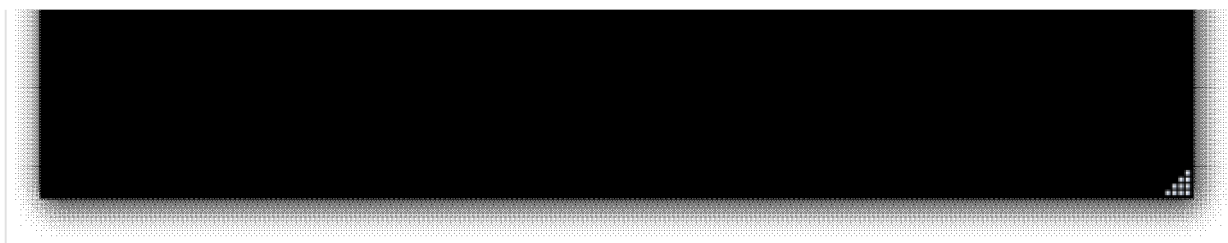
public partial class MainWindow : Metrowindow
{
}
```

But in most cases you can just drop the base class (because this is a `partial` class the XAML should take care of this):

```
public partial class MainWindow
{
}
```

The end result will look something like this:





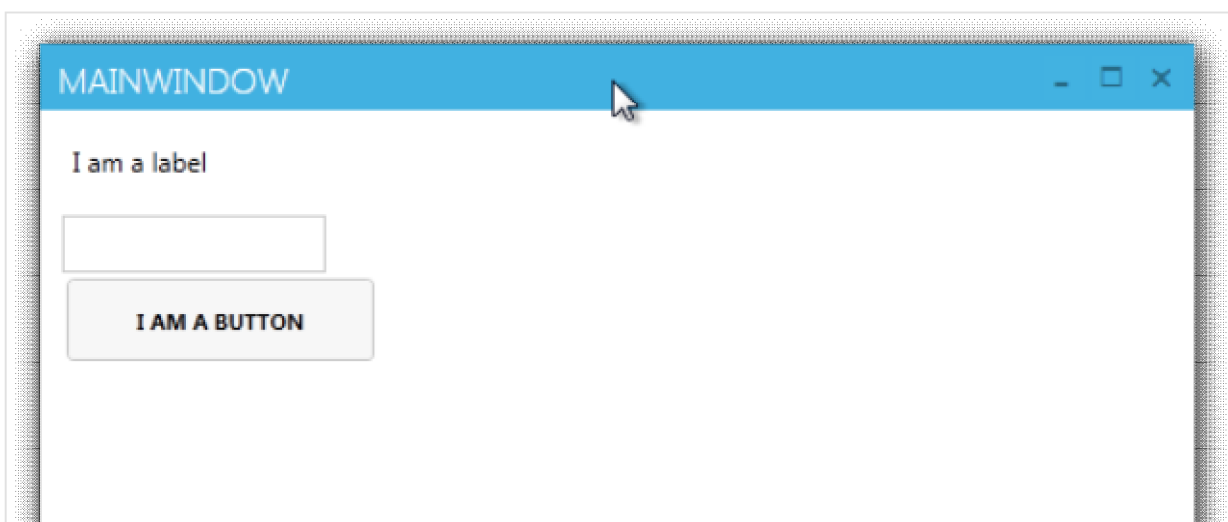
Using Built-In Styles

All of MahApp.Metro's resources are contained within separate resource dictionaries. In order for most of the controls to adopt the MahApps.Metro theme, you will need to add the following ResourceDictionaries to your `App.xaml`

App.xaml

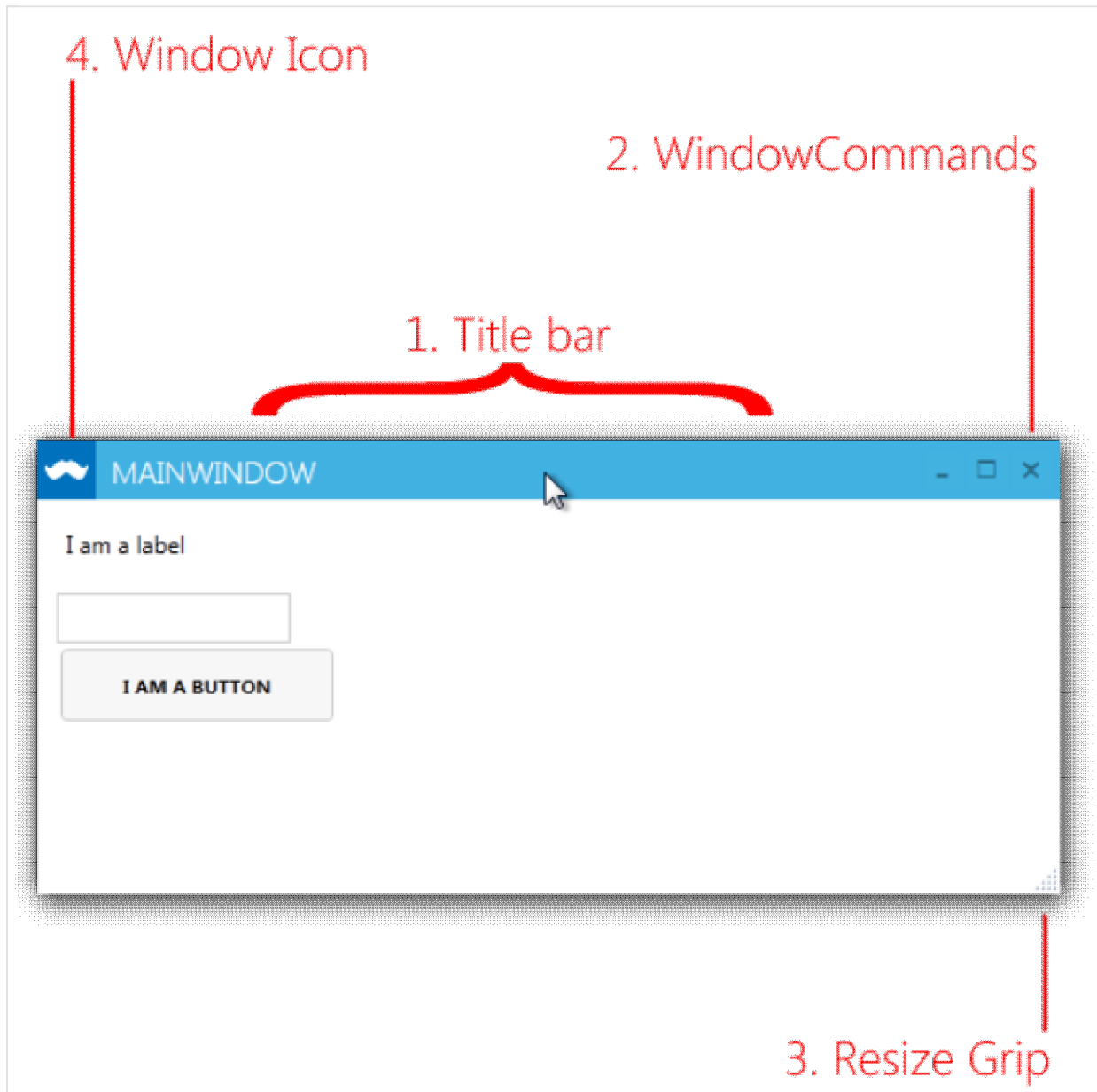
```
<Application.Resources>
  <ResourceDictionary>
    <ResourceDictionary.MergedDictionaries>
      <ResourceDictionary
Source="pack://application:,,,/MahApps.Metro;component/Styles/Controls.xaml" />
      <ResourceDictionary
Source="pack://application:,,,/MahApps.Metro;component/Styles/Fonts.xaml" />
      <ResourceDictionary
Source="pack://application:,,,/MahApps.Metro;component/Styles/Colors.xaml" />
      <ResourceDictionary
Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/Blue.xaml"
/>
      <ResourceDictionary
Source="pack://application:,,,/MahApps.Metro;component/Styles/Accents/BaseLight.x
aml" />
    </ResourceDictionary.MergedDictionaries>
  </ResourceDictionary>
</Application.Resources>
```

The end result will look something like this. If you want to know more about how the control works, more information can be found below.



What's a MetroWindow?

The default `Metrowindow` is made up of a few components:



If you don't like the elements that are labelled, fear not, they're all optional.

- The titlebar is what sets `Metrowindow` apart from rolling your own.
`ShowTitleBar="true|false"`

- The resize grip is not the *only* way to resize a `Metrowindow` - all edges and corners can be gripped, but given a `Metrowindow` doesn't have a noticeable window "chrome" like an aero window, the resize grip can help reassure users.
- Instead of using static images, the icons for minimize/maximize/close are a font called **Marlett**. To explain why this is so requires a walk down memory lane, or at least a visit to [the Wikipedia article](#) about it.
- You can even hide the icons on the title bar by setting the `ShowIconOnTitleBar="true|false"`

Customization

WindowCommands

`windowCommands` are the minimize, maximize/restore, and close buttons. You can add your own controls to `windowsCommands` - by default, buttons have a style automatically applied to them to make them fit in with the rest of the `windowsCommands`. As of 0.9, you are no longer limited to just buttons, but any control. Be aware, you're responsible for styling anything other than buttons.

Including this within the `Metrowindow` tag:

```
<Controls:Metrowindow.RightWindowCommands>
  <Controls:WindowCommands>
    <Button Content="settings" />
    <Button>
      <StackPanel Orientation="Horizontal">
        <Rectangle width="20" Height="20"
          Fill="{Binding RelativeSource={RelativeSource
AncestorType=Button}, Path=Foreground}">
          <Rectangle.OpacityMask>
            <VisualBrush Stretch="Fill"
              Visual="{StaticResource appbar_cupcake}" />
          </Rectangle.OpacityMask>
        </Rectangle>
        <TextBlock Margin="4 0 0 0"
          VerticalAlignment="Center"
          Text="deploy cupcakes" />
      </StackPanel>
    </Button>
  </Controls:WindowCommands>
</Controls:Metrowindow.RightWindowCommands>
```

Make sure to include the [icons](#) to get the cupcake

Produces this window titlebar:





The foreground (link) colour of `windowCommands` will always be white, *unless* the titlebar is disabled, in which case it will be the reverse of whatever theme you have selected. For example, using the White/Light theme, the foreground colour will be black.

There is also the `LeftwindowCommands`, that orientates the commands at the left side of the titlebar.

What Next?

For extended documentation, take a look at the [Controls](#) page

Original URL:

<http://mahapps.com/guides/quick-start.html>