

## halcon 与 C#混合编程之 Halcon 代码调用

---

### 写在前面

完成 halcon 与 C#混合编程的环境配置后，进行界面布局设计构思每一个按钮所需要实现的功能，将 Halcon 导出的代码复制至相应的 C#模块下即可。

halcon 源程序：

```
dev_open_window(0, 0, 512, 512, 'black', WindowHandle)
read_image (Image, 'C:/Users/Administrator/Desktop/猫.jpg')
dev_display(Image)
get_image_size(Image, Width, Height)
rgb3_to_gray(Image, Image, Image, ImageGray)
dev_display(ImageGray)
```

注意：写 halcon 程序时，如果过程中的图片需要显示出来，则需要在每个过程中都添加 dev\_display ( \*\* )

第一步：导出 C#程序，建立项目，并添加此类

```
//
// File generated by HDevelop for HALCON/DOTNET (C#) Version 10.0
//
// This file is intended to be used with the HDevelopTemplate or
```

```
// HDevelopTemplateWPF projects located
under %HALCONEXAMPLES%\c#
using System;
using HalconDotNet;
public partial class HDevelopExport
{
    public HTuple hv_ExpDefaultWinHandle;

    // Main procedure
    private void action()
    {
        // Local iconic variables
        HObject ho_Image, ho_ImageGray;

        // Local control variables
        HTuple hv_Width, hv_Height;

        // Initialize local and output iconic variables
        HOperatorSet.GenEmptyObj(out ho_Image);
        HOperatorSet.GenEmptyObj(out ho_ImageGray);

        //dev_open_window(...);

        ho_Image.Dispose();

        HOperatorSet.ReadImage(out ho_Image,
"C:/Users/Administrator/Desktop/猫".jpg");

        HOperatorSet.DispObj(ho_Image, hv_ExpDefaultWinHandle);
```

```
    HOperatorSet.GetSize(ho_Image, out hv_Width, out
hv_Height);

    ho_ImageGray.Dispose();

    HOperatorSet.Rgb3ToGray(ho_Image, ho_Image, ho_Image, out
ho_ImageGray);

    HOperatorSet.DispObj(ho_ImageGray, hv_ExpDefaultWinHandle);

    ho_Image.Dispose();

    ho_ImageGray.Dispose();
}

public void InitHalcon()
{
    // Default settings used in HDevelop

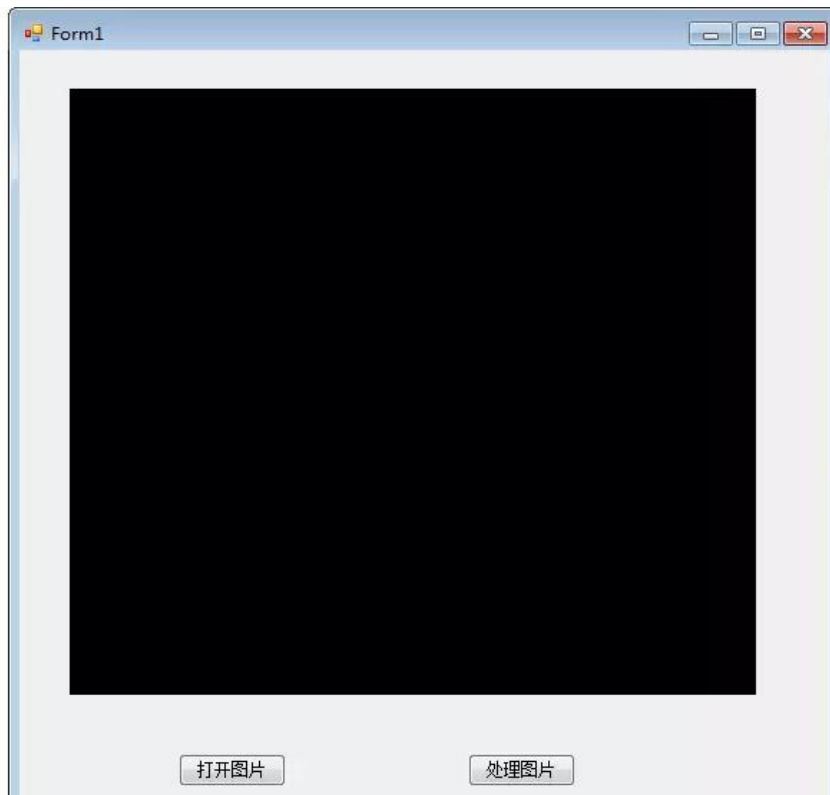
    HOperatorSet.SetSystem("do_low_error", "false");
}

public void RunHalcon(HTuple Window)
{
    hv_ExpDefaultWinHandle = Window;

    action();
}
}
```

第二步：布局设计

本程序需要显示图片的 halconWindow 控件和读图以及处理图片的两个 button 控件；如下图：



第三步：将导出的类进行框架分解

如果想在程序中添加不同的按键，比如一个用来打开图片，一个用来处理图片，则就需要将 halcon 导出的类进行分解；

框架如下：

```
//定义变量
```

```
//初始化 halcon
```

```
public void InitHalcon()
```

```
{
```

```
    //初始化 halcon 程序
```

```
}  
  
//读取图片  
  
public void ReadPicture(HTuple Window)  
  
{  
  
//读图程序  
  
  
  
}  
  
//处理图片  
  
public void Process()  
  
{  
  
  
  
  
  
  
//处理程序  
  
}
```

#### 第四步：填补框架

将对应框架下得程序进行填补操作（下边程序可以代替 public partial class HDevelopExport 下的所有程序）：

```
//定义变量  
  
public HTuple hv_ExpDefaultWinHandle;  
  
HObject ho_Image, ho_ImageGray;  
  
HTuple hv_Width, hv_Height;  
  
//初始化 halcon
```

```
public void InitHalcon()

{
    // Default settings used in HDevelop
    HOperatorSet.SetSystem("do_low_error", "false");
}

public void ReadPicture(HTuple Window)
{
    //读图并显示
    hv_ExpDefaultWinHandle = Window;
    HOperatorSet.GenEmptyObj(out ho_Image);
    ho_Image.Dispose();
    HOperatorSet.ReadImage(out ho_Image,
"C:/Users/Administrator/Desktop/猫".jpg");
    HOperatorSet.DispObj(ho_Image, hv_ExpDefaultWinHandle);
    HOperatorSet.GetImageSize(ho_Image, out hv_Width, out
hv_Height);
}

public void Process()
{
    //处理程序
    HOperatorSet.GenEmptyObj(out ho_ImageGray);
```

```
ho_ImageGray.Dispose();

HOperatorSet.Rgb3ToGray(ho_Image, ho_Image, ho_Image, out
ho_ImageGray);

HOperatorSet.DispObj(ho_ImageGray, hv_ExpDefaultWinHandle);

ho_Image.Dispose();

ho_ImageGray.Dispose();

}
```

第五步：在主程序 Form1.cs 中添加程序：

①在主程序 Form1 的类下对刚才添加的类进行实例化：

```
HDevelopExport hd = new HDevelopExport();
```

②在按键下添加相应程序

```
private void button1_Click(object sender, EventArgs e)
{
    hd.ReadPicture(hWindowControl1.HalconWindow);
}

private void button2_Click(object sender, EventArgs e)
{
    hd.Process();
}
```

此步骤完成之后，程序可以运行。

附加步骤 1：

在以上程序中，已经可以将图片显示出来，并进行处理，但由于窗口与图片的大小不一致，导致在显示窗口中不能适应程序窗口的大小，下面是解决此类问题的方法：

在读图程序中，将获取图片的大小的程序

```
HOperatorSet.GetImageSize(ho_Image, out hv_Width, out hv_Height);放
```

在显示图片之前，并在显示图片程序之前加入下面程序：

```
HOperatorSet.SetPart(hv_ExpDefaultWinHandle, 0, 0, hv_Height - 1,  
hv_Width - 1);然后图片就可以适应窗口大小。
```

此步骤也可以在 halcon 中更改。

附加步骤 2：

如果想在程序中能够打开本地文件夹的图片进行处理，则需要改以下程序：

注意：①②是添加的类中③④是主程序中

①将 `public void ReadPicture(HTuple Window)` 的括号中加入参数，改为  
`public void ReadPicture(HTuple Window, string ImagePath)`

②将 `ReadPicture` 下的程序 `HOperatorSet.ReadImage(out ho_Image,`  
`"C:/Users/Administrator/Desktop/猫".jpg);`改为

```
HOperatorSet.ReadImage(out ho_Image, ImagePath);
```

③主程序中定义变量 `string ImagePath`

④将 `button1` 的程序改成如下：

```
private void button1_Click(object sender, EventArgs e)
```



```
{  
  
    OpenFileDialog openFileDialog1 = new OpenFileDialog();  
    openFileDialog1.Filter = "JPEG 文件|*.jpg|BMP 文件|*.bmp*";  
    openFileDialog1.RestoreDirectory = true;  
    openFileDialog1.FilterIndex = 1;  
    if (openFileDialog1.ShowDialog() == DialogResult.OK)  
    {  
        ImagePath = openFileDialog1.FileName;  
        hd.ReadPicture(hWindowControl1.HalconWindow,  
ImagePath) ;  
    }  
}
```

接下来可以试试运行程序。