

C# 系统应用之窗体最小化至任务栏及常用操作

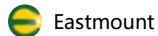
原创 Eastmount 最后发布于2014-01-21 17:50:42 阅读数 11118 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...



¥9.90

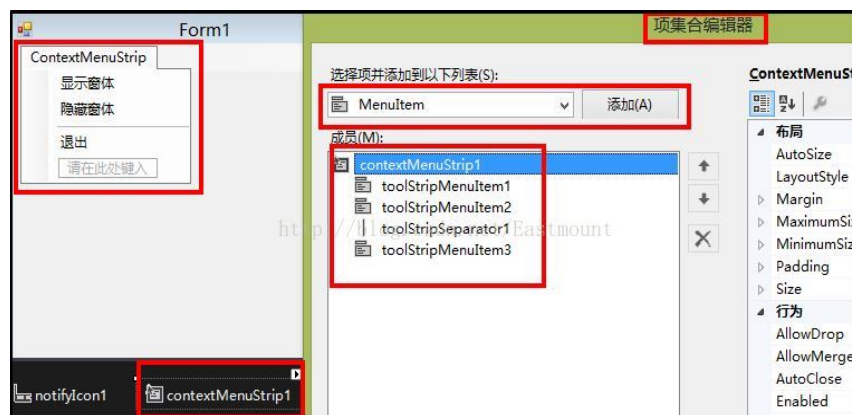
去订阅

现在很多软件如360杀毒软件、酷狗音乐都拥有最小化至任务栏并在系统托盘区留下图标，我的项目同样需要该功能，所以本文主要叙述如何使用C#让窗体最小化至任务栏，同时在系统托盘区的图标点击左键能显示窗体，右键能关联控件显示3个常用功能。主要涉及到的控件是NotifyIcon和ContextMenuStrip，此类文章较多，希望作者以自己的独特视角给大家帮助。

一.界面操作

1.创建"Windows窗体应用程序"项目,从"工具箱"中添加NotifyIcon(运行期间在Windows任务栏右侧的通知区域显示图标).鼠标右击notifyIcon1属性,为控件属性Icon添加图标,Text属性为"CSDN".

2.添加ContextMenuStrip(当用户右击关联控件时显示快捷菜单).鼠标右击contextMenuStrip1属性,进入Items添加或右键"编辑项".添加3个toolStripMenuItem,设置其Text为"显示窗体"、"隐藏窗体"、"退出".如下图所示:

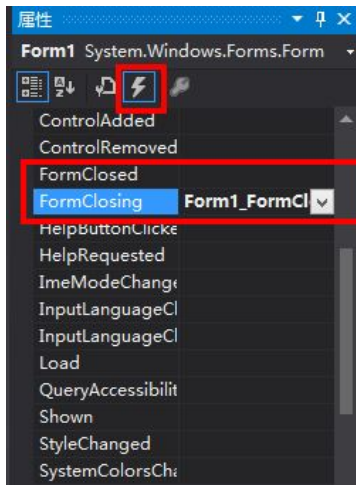


3.关联系统托盘图标与右键菜单.设置notifyIcon1的ContextMenu属性为contextMenuStrip1关联两个控件.运行程序,右下角任务栏的系统托盘处图标点击右键显示如下图所示:



二.窗体设置

窗体设置主要是当窗体点击"退出"按钮时,任务栏仍然显示图标且程序没有退出.设置Form1的MaximizeBox(窗体是否能最大化)属性设置为False,让其不能最大化.并为Form1添加FormClosing(当用户关闭窗口时,在窗体已关闭并制定关闭原因前发生)事件.如下图所示.



添加代码如下,主要实现的功能是当用户点击窗体“关闭”按钮或通过Alt+F4快捷关闭时,取消关闭操作且窗体隐藏,任务栏图标仍然显示:

```
// 窗体关闭前发生事件
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    // 窗体关闭原因为单击"关闭"按钮或Alt+F4
    if (e.CloseReason == CloseReason.UserClosing)
    {
        e.Cancel = true;           // 取消关闭操作 表现为不关闭窗体
        this.Hide();              // 隐藏窗体
    }
}
```

其中:FormClosingEventArgs 类为FormClosing事件提供数据,其属性Cancel获取或设置是否应取消事件的值,CloseReason获取一个值,该值指示关闭窗体的原因.(详见MSDN FormClosingEventArgs 类)

注意:添加的事件是Form_Closing-窗体关闭前发生,而不是Form_Closed窗体已关闭发生.它没有e.Cancel属性,会提示错误 "System.Windows.Forms.FormClosedEventArgs"不包含Cancel的定义.

三.系统托盘功能

常见的窗体最小化至任务栏(系统托盘)图标的功能:

1. 当鼠标左键点击图标时,显示窗体.
2. 当鼠标右键点击图标时,显示“显示窗体”\“隐藏窗体”\“退出”菜单栏,并有相对应的功能.

具体操作是:分别点击“显示窗体”\“隐藏窗体”\“退出”在其属性栏中添加“Click”事件. 添加代码如下:

```
// "显示窗体" 单击事件
private void toolStripMenuItem1_Click(object sender, EventArgs e)
{
    this.Show();                // 窗体显示
    this.WindowState = FormWindowState.Normal; // 窗体状态默认大小
    this.Activate();            // 激活窗体给予焦点
}

// "隐藏窗体" 单击事件
private void toolStripMenuItem2_Click(object sender, EventArgs e)
{
    this.Hide();                // 隐藏窗体
}

// "退出" 单击事件
```

```
private void toolStripMenuItem3_Click(object sender, EventArgs e) {
    // 点击"是(YES)"退出程序
    if (MessageBox.Show("确定要退出程序?", "安全提示",
        System.Windows.Forms.MessageBoxButtons.YesNo,
        System.Windows.Forms.MessageBoxIcon.Warning)
        == System.Windows.Forms.DialogResult.Yes)
    {
        notifyIcon1.Visible = false;    // 设置图标不可见
        this.Close();                  // 关闭窗口体
        this.Dispose();                // 释放资源
        Application.Exit();            // 关闭应用程序窗体
    }
}
```

其中,窗体的状态FormWindowState有Minimized(最小化)、Maximized(最大化)、Normal(默认大小).有的程序设置SizeChanged事件,当用户点击"最小化"按钮窗体尺寸变化时才最小化至任务栏(系统托盘).但我认为打开程序时就有最小化图标更好,同时添加FormClosing事件更符合用户使用.点击"退出"运行结果如下图所示:



最后添加鼠标左键图标显示窗体功能.右键notifyIcon1属性,添加MouseDown(鼠标单击组件时发生)事件.添加代码如下:

```
// 鼠标左键图标事件
private void notifyIcon1_MouseClick(object sender, MouseEventArgs e)
{
    // 点击鼠标"左键"发生
    if (e.Button == MouseButtons.Left)
    {
        this.Visible = true;          // 窗体可见
        this.WindowState = FormWindowState.Normal; // 窗体默认大小
        this.notifyIcon1.Visible = true; // 设置图标可见
    }
}
```

四.完整代码

源代码如下:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
```

```

using System.Text; using System.Threading.Tasks;
using System.Windows.Forms;

namespace WinFormMin
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        // 窗体关闭前发生事件
        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            // 窗体关闭原因为单击"关闭"按钮或Alt+F4
            if (e.CloseReason == CloseReason.UserClosing)
            {
                e.Cancel = true;           //取消关闭操作 表现为不关闭窗体
                this.Hide();               //隐藏窗体
            }
        }

        // "显示窗体" 单击事件
        private void toolStripMenuItem1_Click(object sender, EventArgs e)
        {
            this.Show();                   // 窗体显示
            this.WindowState = FormWindowState.Normal; // 窗体状态默认大小
            this.Activate();                // 激活窗体给予焦点
        }

        // "隐藏窗体" 单击事件
        private void toolStripMenuItem2_Click(object sender, EventArgs e)
        {
            this.Hide();                   // 隐藏窗体
        }

        // "退出" 单击事件
        private void toolStripMenuItem3_Click(object sender, EventArgs e)
        {
            // 点击"是(YES)"退出程序
            if (MessageBox.Show("确定要退出程序?", "安全提示",
                System.Windows.Forms.MessageBoxButtons.YesNo,
                System.Windows.Forms.MessageBoxIcon.Warning)
                == System.Windows.Forms.DialogResult.Yes)
            {
                notifyIcon1.Visible = false; // 设置图标不可见
                this.Close();                 // 关闭窗口体
                this.Dispose();              // 释放资源
                Application.Exit();           // 关闭应用程序窗体
            }
        }

        // 鼠标左键图标事件
        private void notifyIcon1_MouseClick(object sender, MouseEventArgs e)
        {
            // 点击鼠标"左键"发生
            if (e.Button == MouseButtons.Left)
            {
                this.Visible = true;          // 窗体可见
                this.WindowState = FormWindowState.Normal; // 窗体默认大小
            }
        }
    }
}

```

```
        this.notifyIcon1.Visible = true;           // 设置图标可见
    }
}
}
```

五.总结

该文章主要是结合自己的项目完成,主要是窗体最小化至任务栏(系统托盘)同时包括一些常用操作. 同时下面两篇文章涉及到点击“最小化”按钮才实现最小到任务栏的文章, 与该篇文章略有不同, 如果想做这方面的可以阅读, 感谢两位文章作者.

http://blog.sina.com.cn/s/blog_45eaa01a01013u36.html


<http://blog.csdn.net/furturerock/article/details/5687793>

最后,希望文章对大家有所帮助,如果该篇文章中有错误或不足之处,请大家海涵!

(By:Eastmount 2014-1-21 下午6点 <http://blog.csdn.net/eastmount>)

👍 点赞 4 ☆ 收藏 ➦ 分享 ...



Eastmount  博客专家

发布了450 篇原创文章 · 获赞 6227 · 访问量 496万+

他的留言板

关注