

C# Help

[Nevron User Interface](#)

Professional Windows Forms UI Suite Rich,
Extensible and Fully Managed
www.nevron.com

[UML Tool for VS.NET](#)

Visual Modeling Entire Dev. Process VB, C+
ASP, C# code engineering.
www.visual-paradigm.com

Ads by Goooooogle

[Printable Version](#)

C# Mail Checker 1.0

By Matt Watson

Title: Mail Checker 1.0

Author: Matt Watson

Email: watsonm2@myrealbox.com

Environment: C# WinXP

Keywords: mail, [IMAP](#), contextMenu, notifyIcon, Sockets, Timers

Level: Intermediate

Description: An article on creating a [program](#) to check your IMAP mail. The pr
tray and will show the number of new messages.

This program is modeled after MSN Messenger.

- [Download source - 47 Kb](#)

Search Forum
(47199 Postings)



Introduction

I love my MSN Messenger, but I wished that I could use it to check other mail to see in the bottom corner how many unread messages I have. So that is w
you need to get this working is an IMAP mail [server](#), a valid user name, and a
together I had to throw together what I know about [Windows forms](#) and IMAP
a very Messenger like feel. I hope you like it

Using the code

There were a few new things I had to figure out in order to get this app workin

1. Notify Icons
2. Multiple Forms
3. Timers
4. Sockets
5. Context Menus

To Start off, create a new Windows Application using C#. You can start out w
add another one later. The first thing you will want to do is drag a few Windo
form. You will need: a NotifyIcon, a fontDialog, 2 timers, and a contextMenu.
apart:

Notify Icon:

The notify icon is what will display in the system tray when you minimize this
will want to do is go to the properties of the form and set ShowInTaskbar to t
will be displayed in the taskbar. Now for the purpose of this application, I wa
be the number of new mail messages (1,2, etc.). If there is an error in the pr
For this I had to draw a bitmap using the [Graphics](#) objects. The code below sl

[C# Help Board](#)

[Archived Articles](#)

680 Articles

[C# Books](#)

[C# Consultants](#)

[Search Site/Articles](#)

[What Is C#?](#)

[Download Compiler](#)

[Code Archive](#)

[Archived Articles](#)

[Advertise](#)

[Contribute](#)

C# Jobs

[Beginners Tutorial](#)

[C# Contractors](#)

[C# Consulting](#)

[Links](#)

[C# Manual](#)

[Contact Us](#)

[Legal](#)

**Build
interactive
diagrams
easier than
you ever
imagined**



**Powerful
Flexible
Easy-to-use**



**FREE
Download With
Full Support!**

GoDiagram
Diagram Components

Best sites ▼

where the fontDialog comes into play. Rather than messing with getting a pair just added a fontDialog and used the default font from it. Cheating? Yes. Doe

```
//must be used to destroy the icon.
[DllImport("user32.dll", EntryPoint="DestroyIcon")]
static extern bool DestroyIcon(IntPtr hIcon);

//Pass in the number of new mail messages
private void UpdateTaskBar(int i)
{
    String TaskBarLetter;
    // Create a graphics instance that draws to a bitmap
    Bitmap bitmap = new Bitmap(16, 16);
    SolidBrush brush = new SolidBrush(fontDialog1.Color);
    Graphics graphics = Graphics.FromImage(bitmap);

    // Draw then number of Messages to the bitmap using the user sel
    if(i != -1)
        TaskBarLetter= i.ToString();
    else
        TaskBarLetter = "X";
    graphics.DrawString(TaskBarLetter, fontDialog1.Font, brush, 0, 0);

    // Convert the bitmap into an icon and use it for the system tray
    IntPtr hIcon = bitmap.GetHIcon();
    Icon icon = Icon.FromHandle(hIcon);
    notifyIcon1.Icon = icon;

    // unfortunately, GetHIcon creates an unmanaged handle which must
    DestroyIcon(hIcon);
}
```

When the application starts, you just minimize the main form and hide it. No tray.

Now when you want to get the application out of the System Tray you have to

Context Menu:

I added a context menu to this application in order to give the system tray icon needed only two items for the menus: Show Config and Exit. First you must add a Notify Icon with this context Menu. That is easy. Just go into the properties of ContextMenu property to the name of your ContextMenu (in this case ContextMenu1). Then you need to add the code. Double click on the contextMenu and add the popup event.

```
//Add the menu items to the context menu
private void contextMenu1_Popup(object sender, System.EventArgs e)
{
    // Clear the contents of the context menu.
    contextMenu1.MenuItems.Clear();

    // Add a menu item
    contextMenu1.MenuItems.Add("Show Config", new System.EventHandler(contextMenu1.MenuItems.Add("Exit", new System.EventHandler(this.F
```

```
protected void Exit_OnClick(System.Object sender, System.EventArgs
{
    Application.Exit();
}
protected void Config_OnClick(System.Object sender, System.EventAr
{
    showForm2();
}
```

When you right click on the icon in the system tray, it will pop up with a menu config. This would probably be a good time to talk about the second form. The file which stores the values for Username, password, and server name. The file information for loading form 2.

```
private void showForm2()
{
    oForm2.MyParentForm = this;
    oForm2.Show();
    oForm2.WindowState = FormWindowState.Normal;
}
```

This will bring the second form up to the front and allow the user to change the data from an XML file and will be stored there also. After the user changes the info button, the data is written back out to the XML and the values in memory are handled. The Closing event of Form2 or else next time you try and use it you will be removed from memory at that time. To do that just override the method. minimize event so that the form disappears and not just goes down the bottom do this for both forms.

```
private void Form2_Closing(object sender, System.ComponentModel.Ca
{
    // Hide the form...
    this.Hide();

    // Cancel the close...
    e.Cancel = true;
}

protected override void OnResize(EventArgs e)
{
    if (this.WindowState == FormWindowState.Minimized)
        this.Hide();
}
```

Timers:

There are 2 timers need for this application. The first one controls how long the status bar is displayed. Currently this is set to 3 seconds. The second timer controls how often it checks for new messages. This is set to 30 seconds.

Sockets:

The other thing that is extremely important to this app is the Socket connection specification. I did some research on how to communicate through IMAP and a telnet session. I know that this works with the MyRealBox IMAP server, I am not sure about them but I don't know. I haven't tested it. First off you will want to create a new Class Form1 (If you don't do this, you will once again run into the variable loss

it). Here is all of the code for making the connection, Sending data to the ser

```
//connect to the IMAP Server
private bool Connect()
{
    try
    {
        string sResponse="";
        richTextBox1.AppendText("Connecting.....");
        tcpclnt= new TcpClient();
        tcpclnt.Connect(oForm2.Server,143); // use the ipaddress a
        Stream stm = tcpclnt.GetStream();
        byte[] bb=new byte[4096];
        int k=stm.Read(bb,0,4096);

        for (int i=0;i<k;i++)
            sResponse += Convert.ToChar(bb[i]).ToString();
        richTextBox1.AppendText(sResponse);
        return true;
    }
    catch
    {
        return false;
    }
}

//sends data to the server and returns the response
//errString is the error message that we are looking for to se
private string SendToServer(string str, string errString)
{
    string sResponse="";
    Stream stm = tcpclnt.GetStream();
    ASCIIEncoding asen= new ASCIIEncoding();
    byte[] ba=asen.GetBytes(str);
    richTextBox1.AppendText("Transmitting.....");

    stm.Write(ba,0,ba.Length);

    byte[] bb=new byte[4096];
    int k=stm.Read(bb,0,4096);

    for (int i=0;i%lt;k;i++)
        sResponse += Convert.ToChar(bb[i]).ToString();

    richTextBox1.AppendText(sResponse);
    if(sResponse.StartsWith(errString))
        return "";

    return sResponse;
}

//send logout message and close the socket
private bool Disconnect()
{
    SendToServer("? LOGOUT"+ CRLF,"");
    tcpclnt.Close();
    return true;
}
```

The last thing you should probably see is the overridden OnLoad event for Form2 to make the connection to the server.

```
protected override void OnLoad(EventArgs e)
{
    UpdateTaskBar(-1);
    oForm2.MyParentForm = this;

    //rect get data from the function returning the working area
    Rectangle rect=new Rectangle();
    rect=Screen.GetWorkingArea(this);

    //set the window down to the lower right hand corner
    this.Location = new Point(rect.Right - this.Size.Width, rect
    if(Connect())
    {
        if(SendToServer("? LOGIN " + oForm2.UserName + " " + oForm
        {
            if(SendToServer("? Select Inbox"+CRLF,"? NO")!="")
            {
                string sMessages = SendToServer("? SEARCH UNSEEN"+CRLF
                if(sMessages != "")
                {
                    iMessages = NewMessages(sMessages);
                    if(iMessages > 0)
                    {
                        labell1.Text = "You have " + iMessages.ToString() +
                    }
                    else
                    {
                        labell1.Text = "You have no new mail messages";
                    }
                    UpdateTaskBar(iMessages);
                }
            }
            else
            {
                labell1.Text = "Some Sort of Error on getting message
            }
        }
        else
        {
            labell1.Text = "Some Sort of Error on Selecting Inbox";
        }
    }
    else
    {
        labell1.Text = "Some Sort of Error on Login";
    }

    Disconnect();
}
else
{
    labell1.Text = "Error on connect";
}

timer1.Enabled = true;
}
```

That is about it for weird things in the app. I didn't put the full source on this download the zip and get it from there.

Points of Interest

I can't say that this is the best coding job that I have ever done, but hey it killed something in the process. So I am happy with it. There are a few things that with this:

1. I still cannot get the icon in the system tray to go away when the app is closed with your mouse and you will get it.
2. I don't like how I did the loading of the parameters in XML but I didn't have a better way.
3. There really is not much in the way of error handling in this application.

Have fun with it!

History

Version 1.0 Initial Version

Finally, a profiler that's even quicker to find fault than your boss.