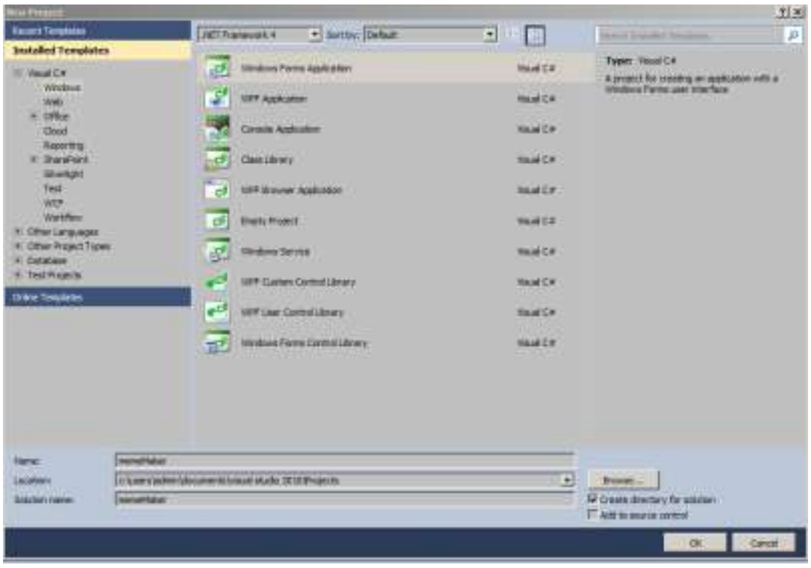
# Create your own Meme Maker in C#

This tutorial will show how to create a meme maker in visual studio 2010 using C#. Now we are using Visual Studio 2010 version you can use any and still get the same result.

Here is the requirement for this project

1. Open an image file
2. Add the top and bottom text
3. Preview the image in real time
4. save the image using the windows dialog box

It's not very complicated however pay extra attention because it's not a basic program this will take extra attention to complete.



Type: Visual C#

A project for creating an application with a Windows Forms user interface

Name: memeMaker

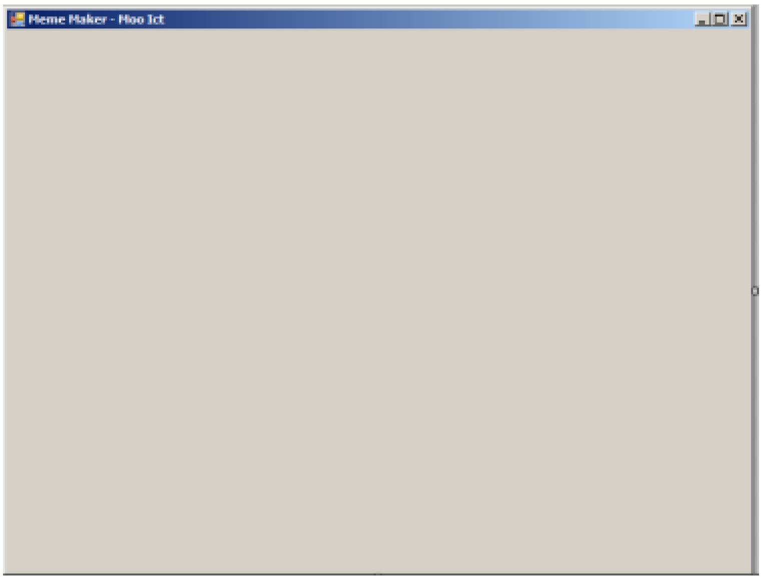
Location: c:\users\admin\documents\visual studio 2010\Projects

Solution name: memeMaker

☑ Create directory for solution

☐ Add to source control

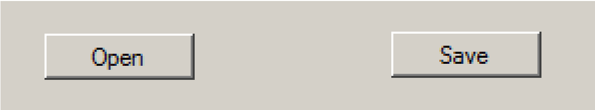
Create a new project called memeMaker. This will help us keep this project organised.

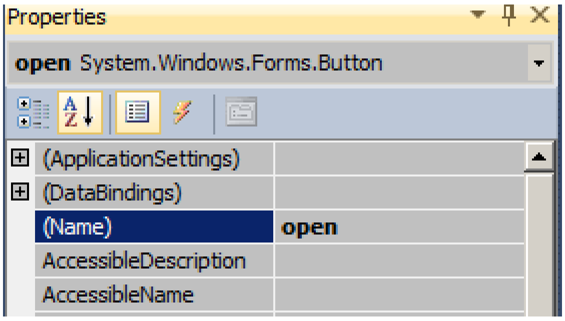


more tutorials on **www.mooict.com**

Resize the form we will need some space in order to see every component in the Application. Also change the form text to Meme Maker - Moo ICT

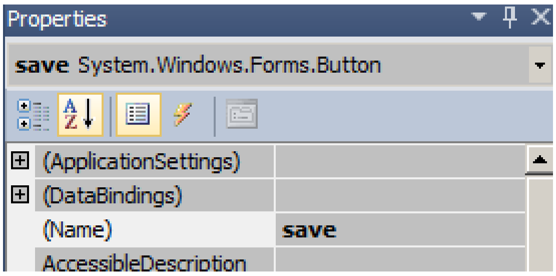
Now let's add two buttons to the form. One will be to open a image file to load in to the program and second button will be used to save the newly created meme.




Properties

open System.Windows.Forms.Button

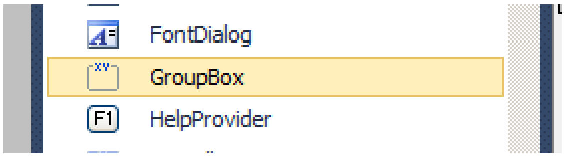
| | |
|---|---|
| ⊞ (ApplicationSettings) | |
| ⊞ (DataBindings) | |
| (Name) | open |
| AccessibleDescription | |
| AccessibleName | |


Properties

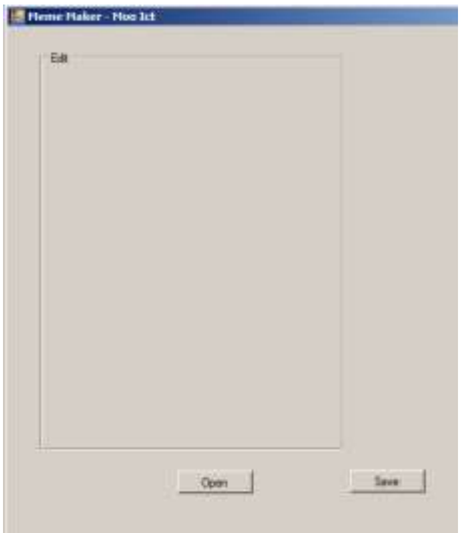
save System.Windows.Forms.Button

| | |
|---|---|
| ⊞ (ApplicationSettings) | |
| ⊞ (DataBindings) | |
| (Name) | save |
| AccessibleDescription | |

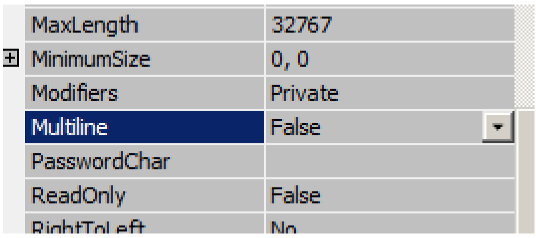
Now change the name of the button from button1/2 to open and save. This way we can call these buttons easily rather than button1 or button2.



Drag and drop a group box to the form. This won't have any special functionality except to group some items together in the form. Make it look professional.



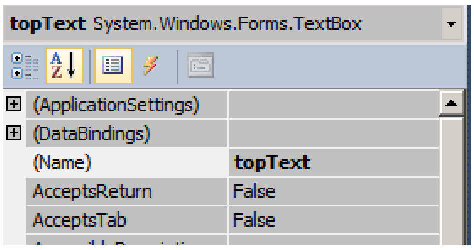
Inside that group we will add 2 labels and 2 text box.



| | |
|---|---|
| MaxLength | 32767 |
| ⊞ MinimumSize | 0, 0 |
| Modifiers | Private |
| Multiline | False |
| PasswordChar | |
| ReadOnly | False |
| RightToLeft | No |

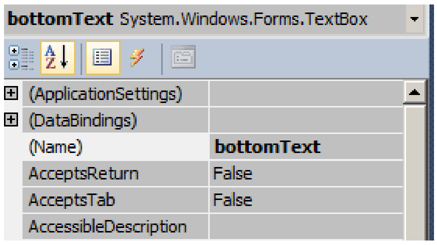
Click on a text box and look for the Multiline option in the properties panel. Change the value from false to true.

Now lets change the name of textBox1 and textBox2 to **topText** and **bottomText**.



**topText** System.Windows.Forms.TextBox

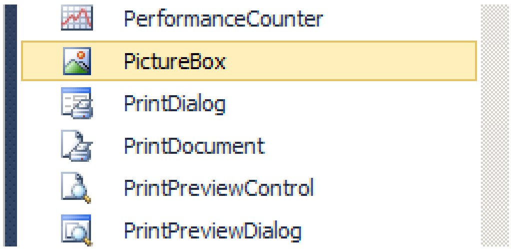
| | |
|---|---|
| ⊞ (ApplicationSettings) | |
| ⊞ (DataBindings) | |
| (Name) | **topText** |
| AcceptsReturn | False |
| AcceptsTab | False |



**bottomText** System.Windows.Forms.TextBox

| | |
|---|---|
| ⊞ (ApplicationSettings) | |
| ⊞ (DataBindings) | |
| (Name) | **bottomText** |
| AcceptsReturn | False |
| AcceptsTab | False |
| AccessibleDescription | |

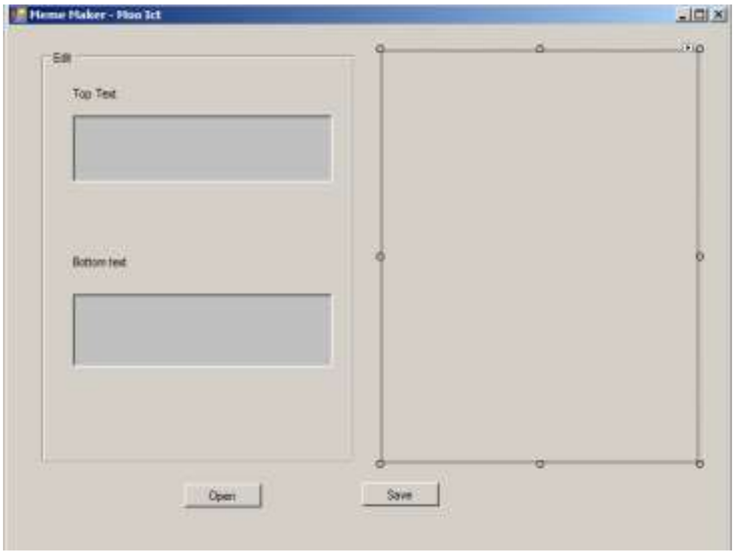
One side of the panel is now done. We need to sort out the other side now.



PerformanceCounter
PictureBox
PrintDialog
PrintDocument
PrintPreviewControl
PrintPreviewDialog

Add a picture box to the form.

name the **picturebox1** to **preview**

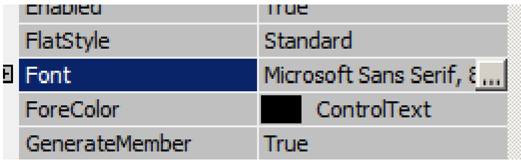


We are really close to start programming however two small things are needed to finish this design part of the tasks.

Now we need to add two Labels to the screen.

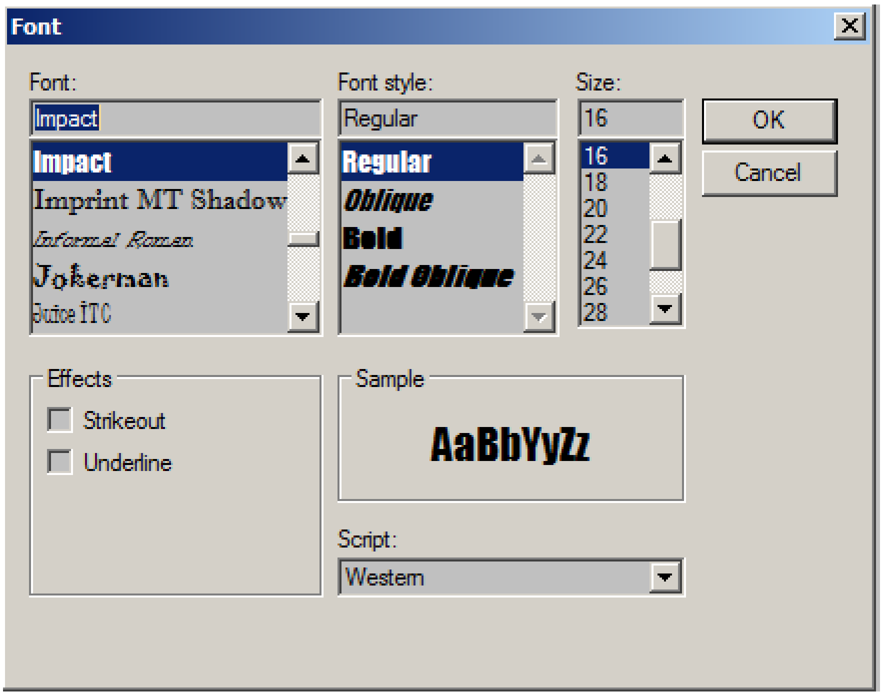
more tutorials on **www.mooict.com**

Name them both to **topLabel** and **bottomLabel**.



| | |
|---|---|
| Enabled | True |
| FlatStyle | Standard |
| Font | Microsoft Sans Serif, 8 ... |
| ForeColor | ControlText |
| GenerateMember | True |

Click on the three dots and change the settings to the following.



Font: Impact

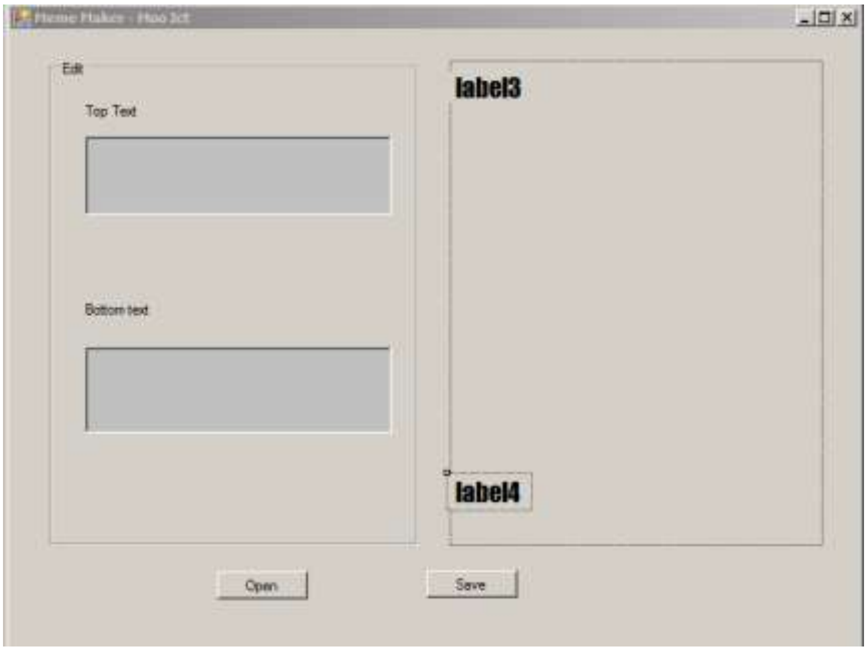
Font style: Regular

Size: 16

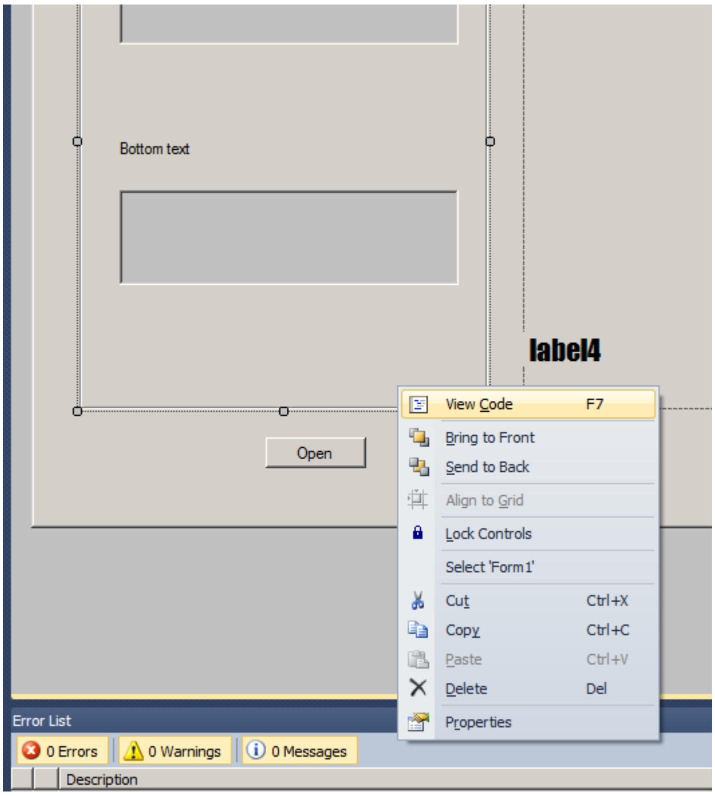
Script: Western

Impact, Regular and 16 font size.

Do this for both labels, now this is the final view of the program design.



more tutorials on **www.mooict.com**



Right click on the form and click on view code.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

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

This is the code in our code view so far.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
```

```
namespace memeMaker
{
    public partial class Form1 : Form
    {
        OpenFileDialog openImage;
        string imageFile;

        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

We have two variables to start with. First one is the open file dialog we are calling it open Image and second is a string we are going to call it image file. The idea is we will use the open file dialog to find an image file and then store its location inside the image file string to be used later.

First thing lets double click on top Text box.

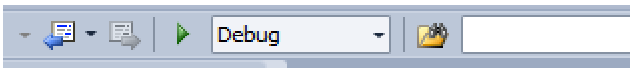
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace memeMaker
{
    public partial class Form1 : Form
    {
        OpenFileDialog openImage;
        string imageFile;

        public Form1()
        {
            InitializeComponent();
        }

        private void topText_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

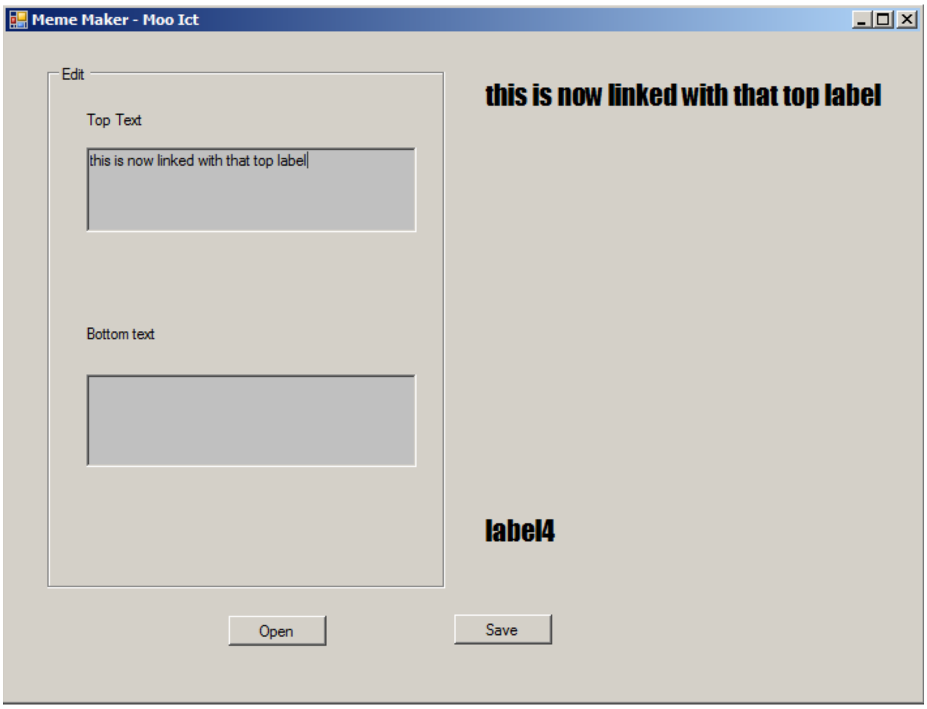
once you have double clicked on it, visual studio will automatically add this function and link it with the text box. We want the top Label to change with the top text box. See how that works with this one line inside the function

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace memeMaker
{
    public partial class Form1 : Form
    {
        OpenFileDialog openImage;
        string imageFile;

        public Form1()
        {
            InitializeComponent();
        }

        private void topText_TextChanged(object sender, EventArgs e)
        {
            topLabel.Text = topText.Text;
        }
    }
}
```

Let's try it out.

run the program by clicking on the debug button on the top. (green play button)





Top Text: this is now linked with that top label

this is now linked with that top label

label4

Any text you enter in the text box will automatically be updated in the top label.

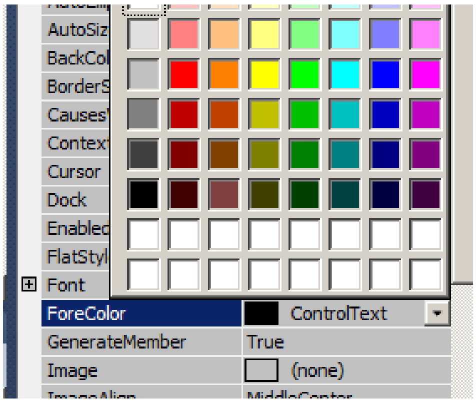
Now lets do the same with bottom text box and bottom label.

more tutorials on **www.mooict.com**

Once again double click on the text box, that will add the function and inside that function add the code

```
bottomLabel.Text = bottomText.Text;
```

Lastly for the labels lets change its colour to white in the properties panel.



ForeColor ControlText
GenerateMember True
Image (none)

Change the foreColor property to white for both labels.

Now for the buttons

Double click on button called open. We will figure out how to open a file from the computer and we only want to accept image files such as jpg, gif, png and bmp.

```
        private void open_Click(object sender, EventArgs e)
        {


        }
```

This code will be now added to your project and it will be linked to your open button.

Now add the following to the open_click function

```
        private void open_Click(object sender, EventArgs e)
        {
            openImage = new OpenFileDialog();
            openImage.InitialDirectory = "c:\\";
            openImage.Filter = "Image files (*.jpg, *.jpeg, *.jpe, *.jfif, *.png) | *.jpg; 
*.jpeg; *.jpe; *.jfif; *.png; *.gif; *.bmp";
            openImage.FilterIndex = 2;
            openImage.RestoreDirectory = true;
        }
```

openImage = new OpenFileDialog(); this line will create a new instance of the open file dialog class which will allow us to select a file from windows index.

openImage.IntialDirectory = "C:\\"; this line will rest the directory to C or local hard drive so we can navigate to where ever.

openImage.Filter = "Image files (*.jpg, *.jpeg, *.jpe, *.jfif, *.png) | *.jpg; *.jpeg; *.jpe; *.jfif; *.png; *.gif; *.bmp"; This will filter the files and only show the image files we want on screen. For this instance it will only show all the image files.

openImage.FilterIndex = 2; This filter represents the index of the filter currently selected in the dialog box. Default is 1 and we have changed it to 2.

Set the FileDialog.RestoreDirectory = true, when you close the FileDialog the original working directory is restored.

Lets add the following to the program now.

```
        private void open_Click(object sender, EventArgs e)
        {
            openImage = new OpenFileDialog();
            openImage.InitialDirectory = "c:\\";
            openImage.Filter = "Image files (*.jpg, *.jpeg, *.jpe, *.jfif, *.png) |
*.jpg; *.jpeg; *.jpe; *.jfif; *.png; *.gif; *.bmp";
            openImage.FilterIndex = 2;
            openImage.RestoreDirectory = true;

            if (openImage.ShowDialog() == DialogResult.OK)
            {
                try
                {
                    preview.Image = Image.FromFile(openImage.FileName);
                    imageFile = openImage.FileName;
                }
                catch (Exception ex)
                {
                    MessageBox.Show("Error: Could not read file from disk. Original
error: " + ex.Message);
                }
            }
        }
```

As you can see we added a if statement to the function. This statement also has a error catching condition in there too. Let us look closely.

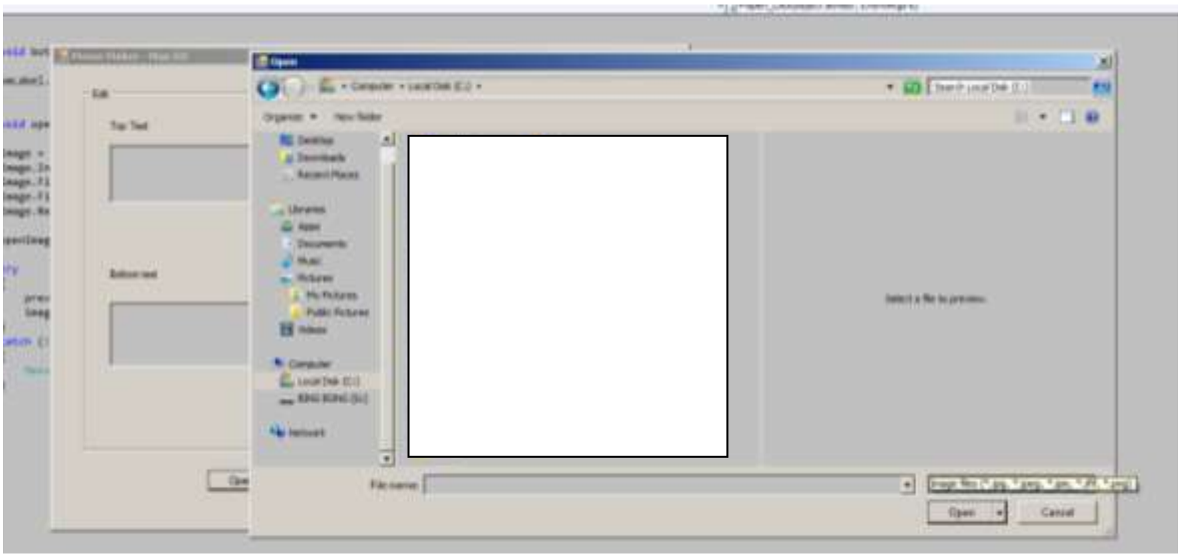
First we declared an if statement to see whether the user has selected a file and clicked OK. If so we will use the try and catch condition.

C# has some inbuilt objects you can use to deal with any potential errors in your code. You tell C# to Try some code, and if can't do anything with it you can Catch the errors.

So if we only used the if statement there will be a certain condition where the program might crash to avoid this tragedy we use the try and catch condition. We will try to open the file user selected and put it inside the preview picture box and if that is not possible we can return an error to the user instead of crashing the whole program.

it's a safety feature and its a good programming practice.

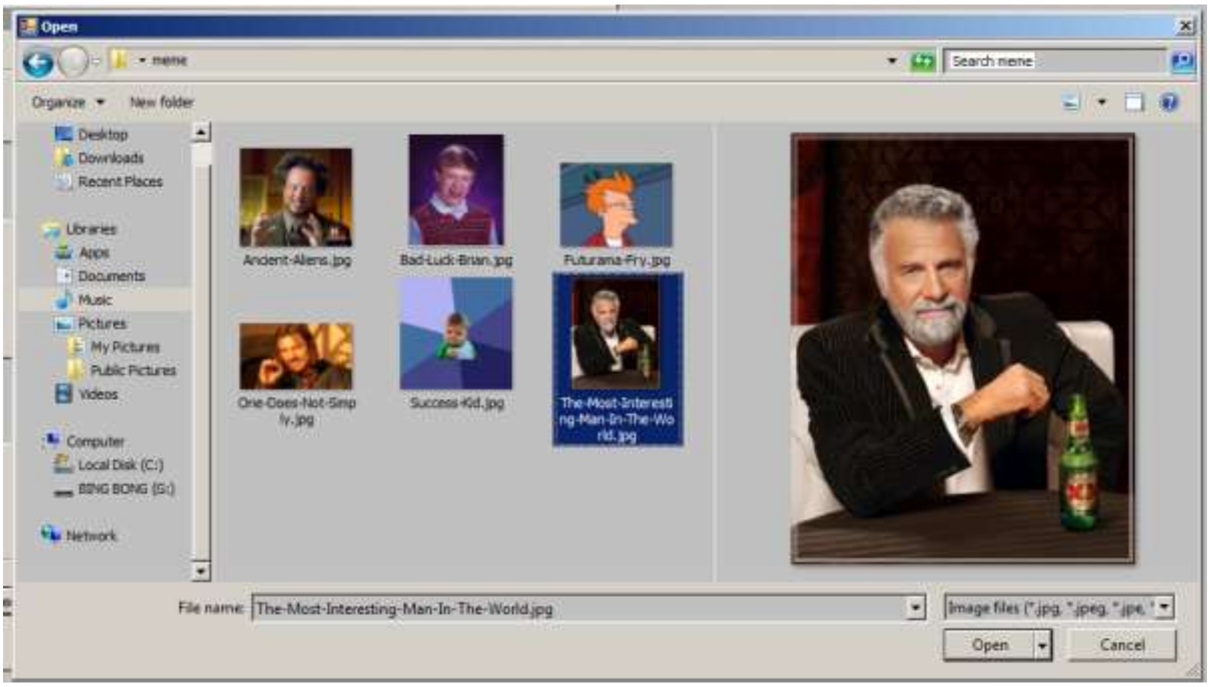
Lets try this out now.



As you can see its opened the dialog box in the C drive and it also filtered the file options.

Now lets load up a meme picture

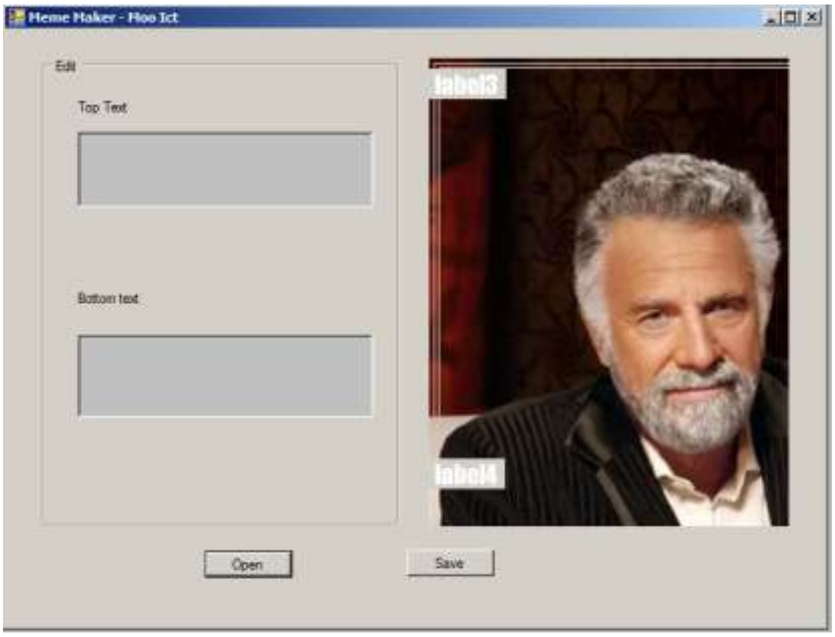
I have saved various meme images inside a meme folder in the desktop so I'll navigate inside that folder now.



File name: The-Most-Interesting-Man-In-The-World.jpg

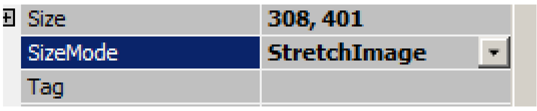
Image files (*.jpg, *.jpeg, *.jpe, *

Ohh yea this is shaping up to be fun now. Click on open



Now as you can see the image seems to be bigger than the picture box. No worries we can stretch it into the picture box.

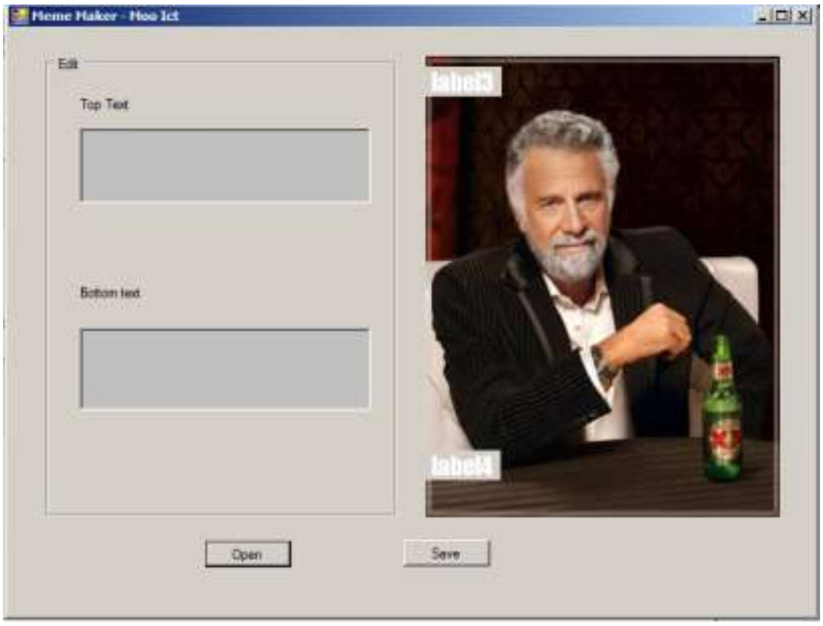
Close the application and look at the properties panel for size mode option.



|  |  |
|---|---|
| Size | 308, 401 |
| SizeMode | StretchImage |
| Tag |  |

Change from NORMAL to STRETCH IMAGE

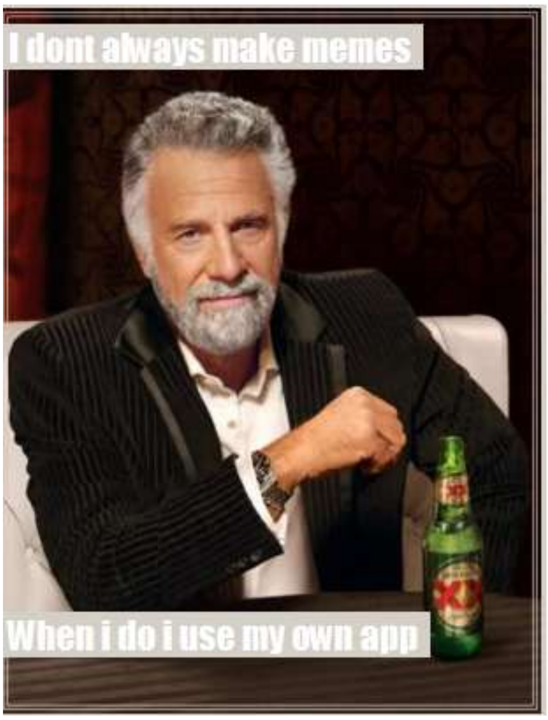
Lets try again.



Problamo solved. Yes I said problamo deal with it.

Now play around with the text and make some meme with it for example

more tutorials on **www.mooict.com**



Nice very classy. Let's move on.

Now that we have the open file and load file functions working we need to solve the save file option. I mean I want to upload my stuff on the internet as soon as possible specially the ones i make using my own meme program.

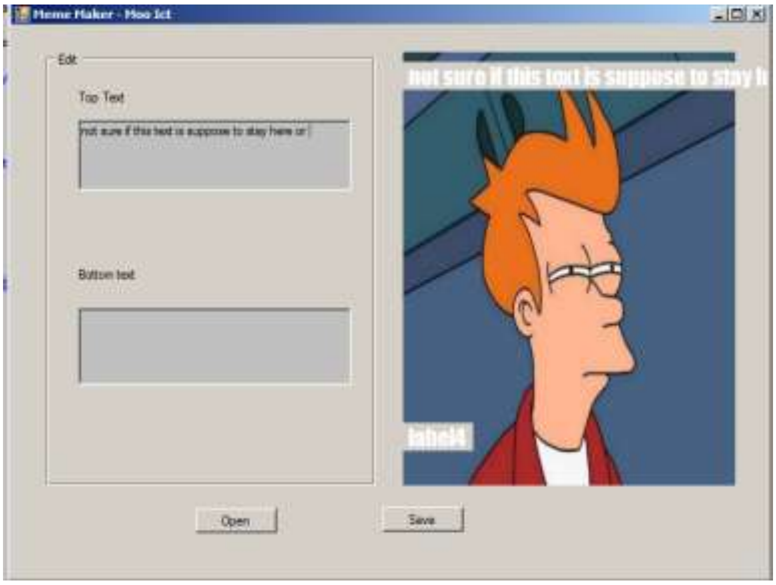
Am i the only one here
who wants to share memes!!!!!

Yep

If you are concerned about the background colour on the images don't be when we save the file it will be transparent.

**Problem**

if you haven't already noticed our text goes over the image and on to the other side.



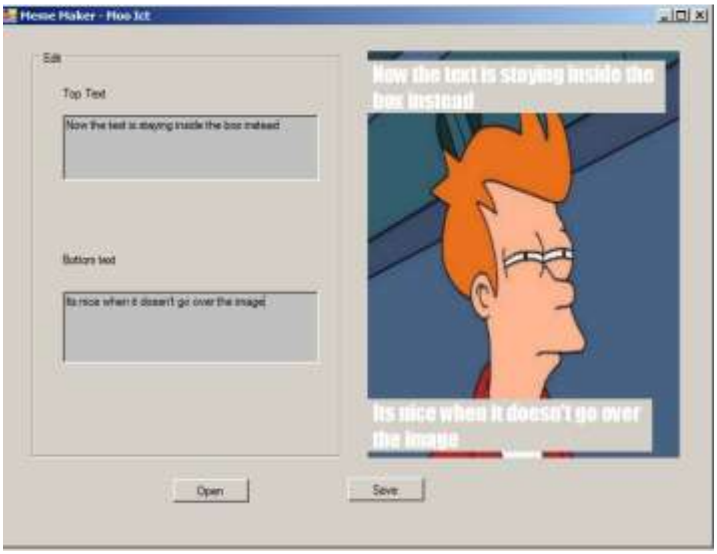
Top Text: not sure if this text is suppose to stay here or

We need to wrap the text to stay inside the picture box width and not surpass it.

add the following inside the top Text and bottom Text functions

```
        private void topText_TextChanged(object sender, EventArgs e)
        {
            topLabel.Text = topText.Text;
            topLabel.MaximumSize = new Size(preview.Width, 200);
        }

        private void bottomText_TextChanged(object sender, EventArgs e)
        {
            bottomLabel.Text = bottomText.Text;
            bottomLabel.MaximumSize = new Size(preview.Width, 200);
        }
```

We are adding a maximum size for the label. inside the new size function we are giving the width of the preview picture and the height of 200px. It should stay within that so problem solved.



Meme Maker - Moo Ict
Edit
Top Text
Bottom text
Now the text is staying inside the box instead
Its nice when it doesn't go over the image
Open
Save

Top Text: Now the text is staying inside the box instead

Bottom text: Its nice when it doesn't go over the image

Ta da!!!!!!. Nice right

more tutorials on **www.mooict.com**

Go back to the design view and double click on the save button.

```
        private void save_Click(object sender, EventArgs e)
        {

        }
```

Inside the function we will add the code which will save our file in the hard drive and give us a usable file to upload where ever.

```
        private void save_Click(object sender, EventArgs e)
        {
            string firstText = topText.Text;
            string secondText = bottomText.Text;

            Bitmap bitmap = (Bitmap)Image.FromFile(imageFile);//load the image file

            RectangleF TopSize = new RectangleF(0, 10, bitmap.Width, 400);
            RectangleF BottomSize = new RectangleF(0, bitmap.Height - 100,
bitmap.Width, 400);


        }
```

First we are declaring two string first text and second text. First text string will contain the contents of top text box and second text will contain the contents of bottom text box.

We need to import the Bitmap class to draw on a image so we call the Bitmap class and name it bitmap with a lower b. then we are telling the bitmap class to locate the image we are using by loading the image file string on to it. This will not only store the image file location it will load it on the memory so we can edit it and save a copy of it.

We are calling two rectangle class for the text boxes. This will help us wrap the text inside the image. The same way we done it before but this time we need to use rectangle to calculate the actual image size rather than the picture box.

`new RectangleF(X, Y, WIDTH, HEIGHT);` inside the rectangle class it accepts 4 different variables. First one is X this is the horizontal position of the text we want so we keep it at 0. second is the Y or vertical position of the text this we can keep it at 10 so it stays a little lower then just the top. third is the width so we give it value of the image width in this case its the bitmap.width and third is the height we will say maximum 400 to be safe for now.

`new RectangleF(0, 10, bitmap.Width, 400);`
For the second text which should appear on the bottom of the meme we need to change the Y position so we are doing this:

`new RectangleF(0, **bitmap.Height - 100**, bitmap.Width, 400);`
as you can see we are dynamically calculating the bitmap.height and deducting 100 pixels so it wont be right at the bottom it be a little higher.

```
        private void save_Click(object sender, EventArgs e)
        {
            string firstText = topText.Text;
            string secondText = bottomText.Text;

            Bitmap bitmap = (Bitmap)Image.FromFile(imageFile);//load the image file

            RectangleF TopSize = new RectangleF(0, 10, bitmap.Width, 400);
            RectangleF BottomSize = new RectangleF(0, bitmap.Height - 100,
bitmap.Width, 400);

            SaveFileDialog saveImage = new SaveFileDialog();
            saveImage.FileName = "*";
            saveImage.DefaultExt = "bmp";
            saveImage.ValidateNames = true;
            saveImage.Filter = "Bitmap Image (.bmp)|*.bmp|Gif Image (.gif)|*.gif |JPEG
Image (.jpg)|*.jpeg |Png Image (.png)|*.png";
        }
```

Here we are calling the save file dialog box. First we are creating a variable called save Image and we are going to store the save file class inside it.

We will reset the file name to * so the user can enter their own.

We will set a default file extension to BMP or bitmap image

We will also validate the image names

lastly we are going to filter the image types and only allow to save in certain file types.

You might of realised the similarities between the open file dialog and save file dialog.

```
        private void save_Click(object sender, EventArgs e)
        {
            string firstText = topText.Text;
            string secondText = bottomText.Text;

            Bitmap bitmap = (Bitmap)Image.FromFile(imageFile);//load the image file

            RectangleF TopSize = new RectangleF(0, 10, bitmap.Width, 400);
            RectangleF BottomSize = new RectangleF(0, bitmap.Height - 100,
bitmap.Width, 400);

            SaveFileDialog saveImage = new SaveFileDialog();
            saveImage.FileName = "*";
            saveImage.DefaultExt = "bmp";
            saveImage.ValidateNames = true;
            saveImage.Filter = "Bitmap Image (.bmp)|*.bmp|Gif Image (.gif)|*.gif |JPEG
Image (.jpg)|*.jpeg |Png Image (.png)|*.png";

            if (saveImage.ShowDialog() == DialogResult.OK)
            {

                using (Graphics graphics = Graphics.FromImage(bitmap))
                {
                    using (Font memeFont = new Font("Impact", 24, FontStyle.Bold,
GraphicsUnit.Point))
                    {
                        graphics.DrawString(firstText, memeFont, Brushes.White,
TopSize);
                        graphics.DrawString(secondText, memeFont, Brushes.White,
BottomSize);
                    }
```

more tutorials on **www.mooict.com**

```
                }

                bitmap.Save(saveImage.FileName);//save the image file

            }
        }
```

Once again we going to use a IF statement inside the function. This will only trigger once the user clicks on Save or OK on the dialog box.

```
 using (Graphics graphics = Graphics.FromImage(bitmap))

{

}
```

Instead of doing the try and catch again and we will use USING condition. This condition will only run once the image has been loaded on to memory.

The using block helps manage resources. Conceptually it protects the whole system's resources by specifying the scope of the usage of the resource.

As you can see we are going to call Graphics class and name it graphics inside the using condition. Then we are going to load the bitmap we created earlier to the graphics class.

Inside the first using statement we will declare another to ensure the text are embedded into the image.

```
using (Font memeFont = new Font("Impact", 24, FontStyle.Bold, GraphicsUnit.Point))
{
}
```

We are specifying the font settings in this one. We will create a Font class called memeFont and specify the font type to Impact, 24 size, BOLD and position.

inside this using statement we are going to declare where to display the two text boxes.

```
graphics.DrawString(firstText, memeFont, Brushes.White, TopSize);
graphics.DrawString(secondText, memeFont, Brushes.White, BottomSize);
```

Remember that graphics class we created in the first using condition this is where it comes to work. We are now going to DRAW STRING or draw the text on the image itself. first draw the firstText string using the memFont settings and a colour white. Also we are going to give it our Top Size location which we created from those rectangles before. We done the same to the second text.

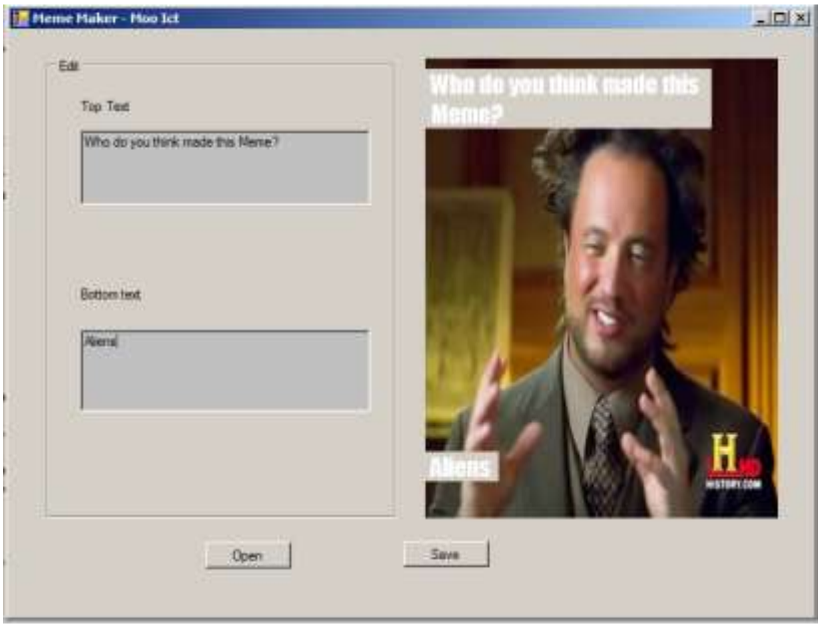
And lastly outside the both using condition we are going to save the file.

```
bitmap.Save(saveImage.FileName);//save the image file
```

The bitmap class has its own save function built in, here we will use that to save a copy of the file inside the system.

more tutorials on **www.mooict.com**

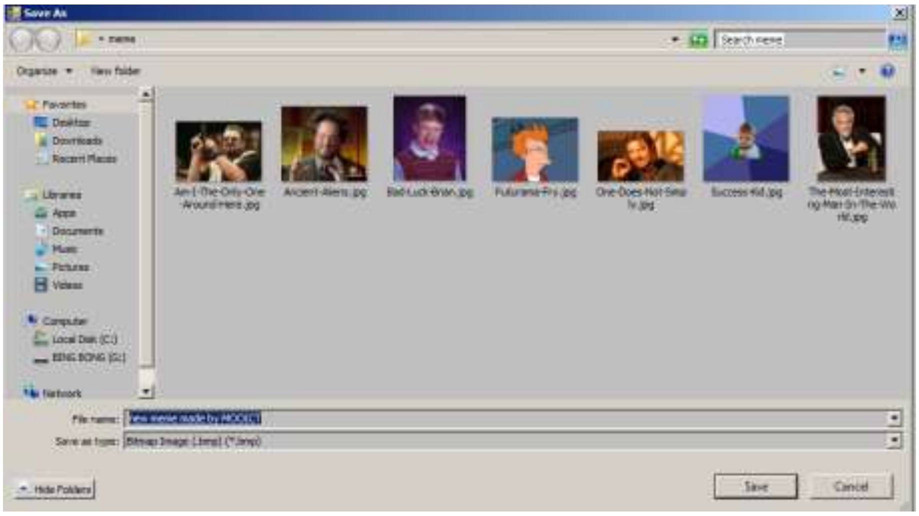
Lets test it out.



Top Text: Who do you think made this Meme?

Bottom text: Aliens

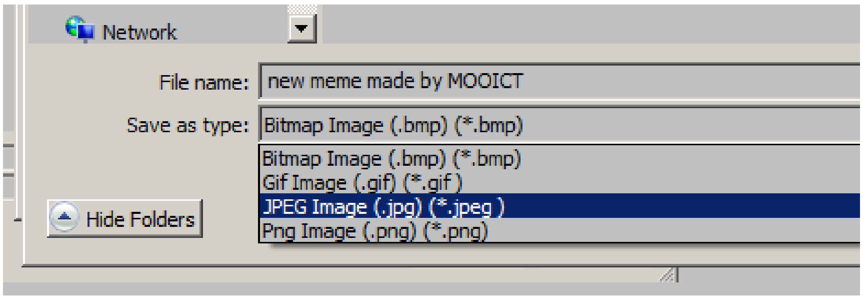
Here is a test meme



File name: new meme made by MOOICT

Save as type: Bitmap Image (.bmp) (*.bmp)

Save dialog box



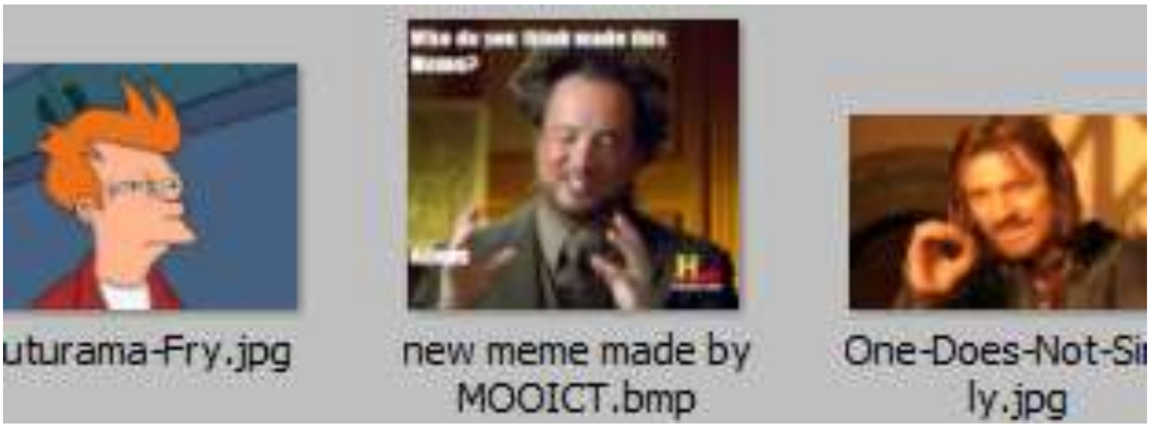
File name: new meme made by MOOICT

Save as type: Bitmap Image (.bmp) (*.bmp)

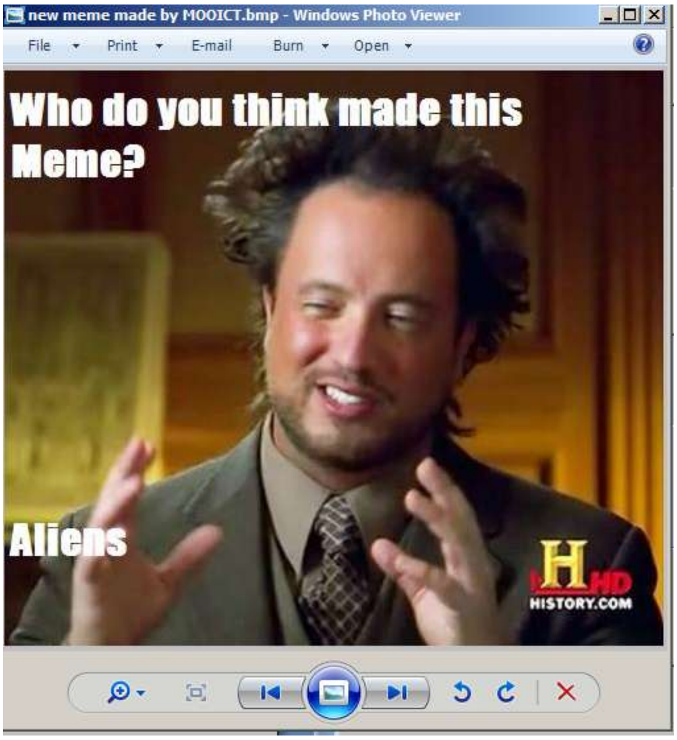
Bitmap Image (.bmp) (*.bmp)
Gif Image (.gif) (*.gif )
JPEG Image (.jpg) (*.jpeg )
Png Image (.png) (*.png)

File types working.

more tutorials on **www.mooict.com**

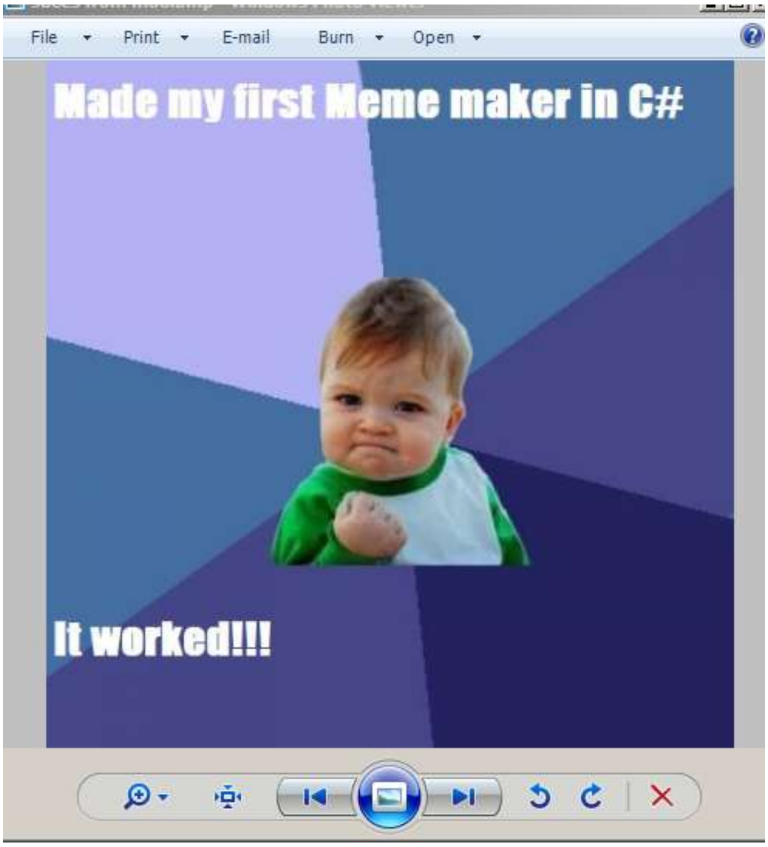


File save worked.


new meme made by MOOICT.bmp - Windows Photo Viewer
File
Print
E-mail
Burn
Open
Who do you think made this Meme?
Aliens
HISTORY.COM

Nice.

more tutorials on **www.mooict.com**



Have fun with it