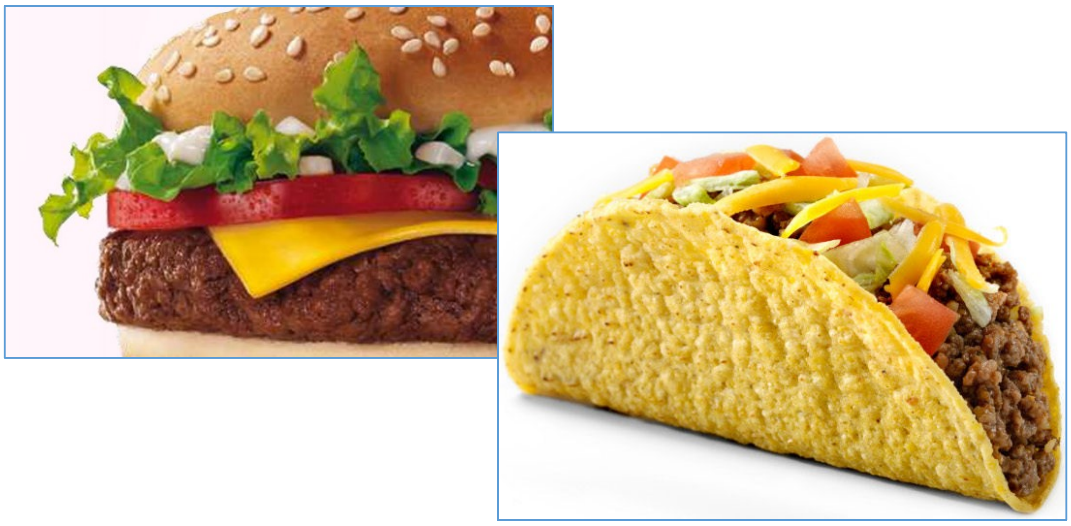
9 Fast Food

This program introduces the technique for uploading picture images to a C# program and storing them in a database table, in a similar way to text or numeric data. This can be very useful, for example, in a shop database where images of the products can accompany the written descriptions.

For this example, we will set up a small program to display a picture menu for a fast food take-away shop.

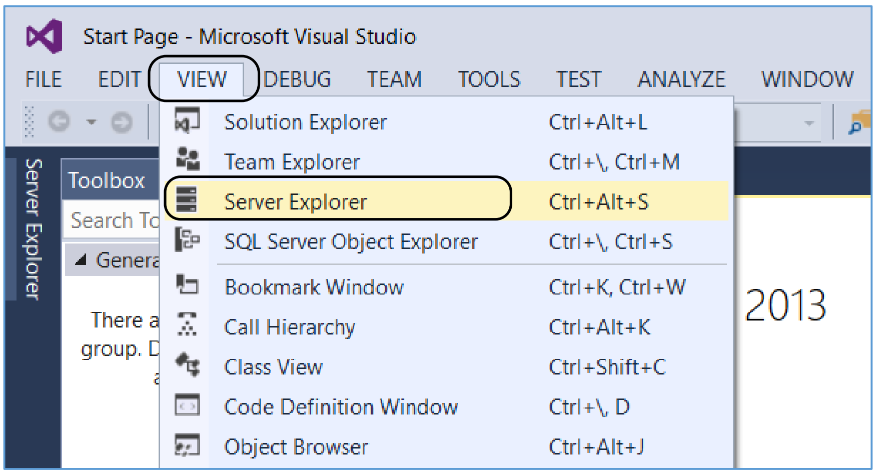
Use the Internet to collect some example photographs of fast food, for example: pizza, burgers, kebabs or tacos. Store these pictures as .JPG images in a folder on your computer.



We begin by setting up a database for the fast food shop, with a table to store details of the food items.

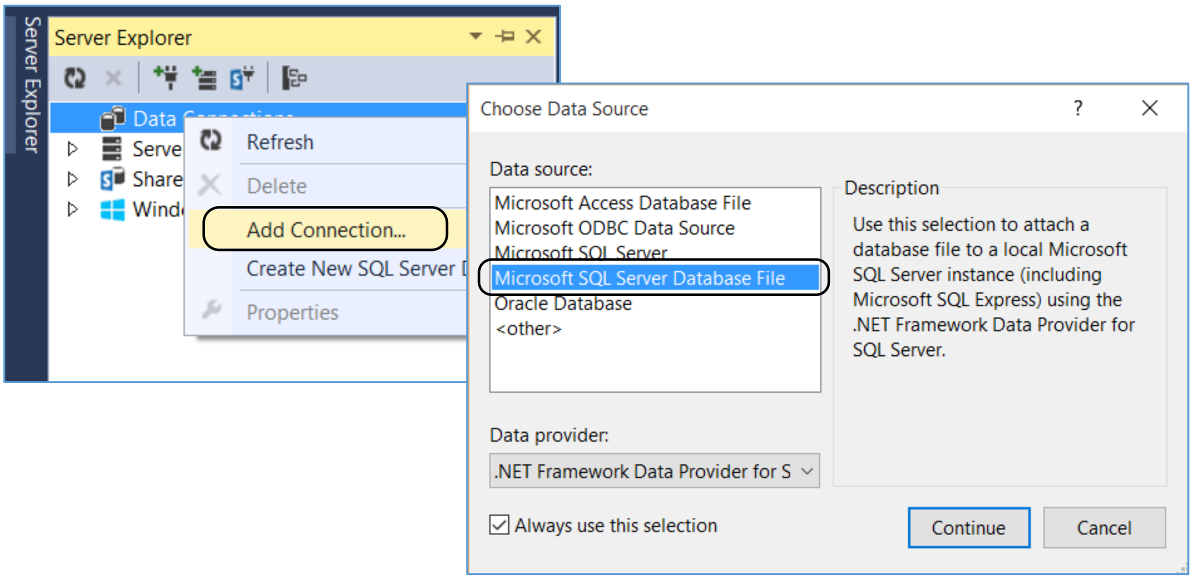
Open *Visual Studio*, but do not start a new project yet.

On the menu line, select '***View / Server Explorer***'


Start Page - Microsoft Visual Studio
FILE EDIT VIEW DEBUG TEAM TOOLS TEST ANALYZE WINDOW
Solution Explorer Ctrl+Alt+L
Team Explorer Ctrl+\, Ctrl+M
Server Explorer Ctrl+Alt+S
SQL Server Object Explorer Ctrl+\, Ctrl+S
Bookmark Window Ctrl+K, Ctrl+W
Call Hierarchy Ctrl+Alt+K
Class View Ctrl+Shift+C
Code Definition Window Ctrl+\, D
Object Browser Ctrl+Alt+J
Server Explorer
Toolbox
Search To
Genera
There a
group. D
2013

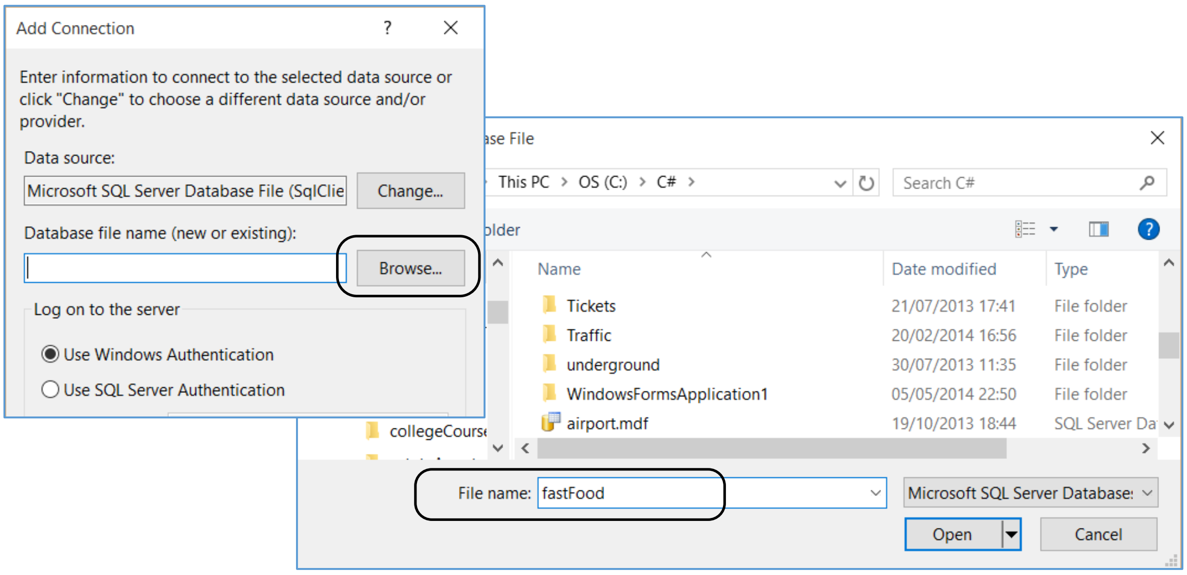
Right-click on '***Data Connections***', then select '***Add Connection***'. If you are asked to choose a Data Source, select:

Microsoft SQL Server Database File



Use this selection to attach a database file to a local Microsoft SQL Server instance (including Microsoft SQL Express) using the .NET Framework Data Provider for SQL Server.

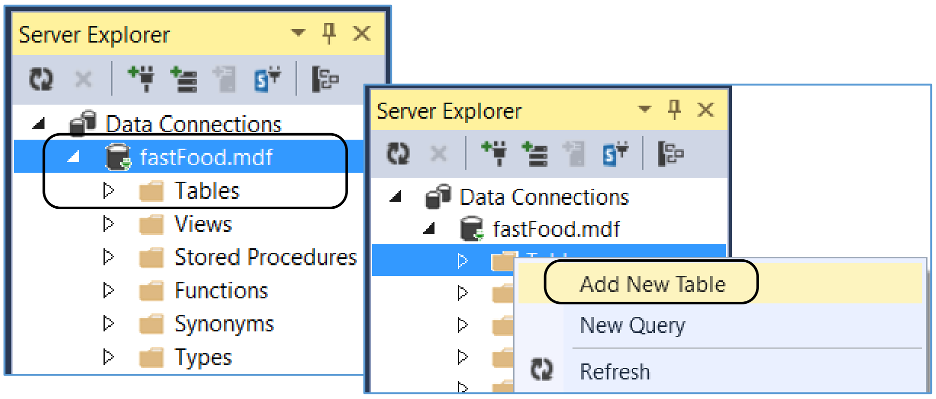
Use the '***Browse***' option to navigate to the location where your C# programs are stored. Give the file name '***fastFood***' for the database which will be created.


Add Connection
Data source:
Microsoft SQL Server Database File (SqlClie
Change...
Database file name (new or existing):
Browse...
Log on to the server
Use Windows Authentication
Use SQL Server Authentication
This PC > OS (C:) > C# >
Search C#
Name
Date modified
Type
File name: fastFood
Microsoft SQL Server Database
Open
Cancel

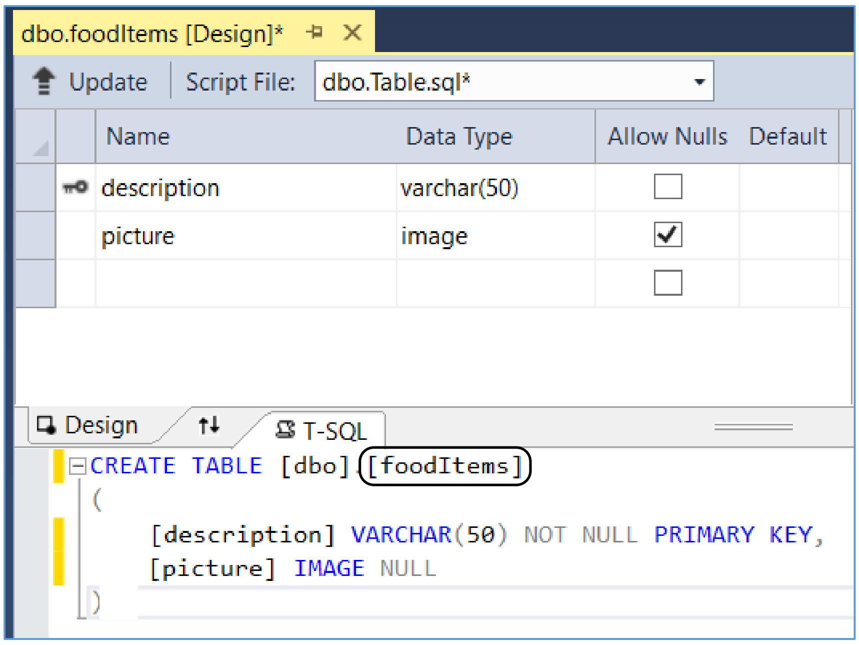
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Name	Date modified	Type
Tickets	21/07/2013 17:41	File folder
Traffic	20/02/2014 16:56	File folder
underground	30/07/2013 11:35	File folder
WindowsFormsApplication1	05/05/2014 22:50	File folder
airport.mdf	19/10/2013 18:44	SQL Server Da

Click '***OK***', then answer '***Yes***' that you wish to create the database file.
Click-right on '***Tables***' and select '***Add New Table***':



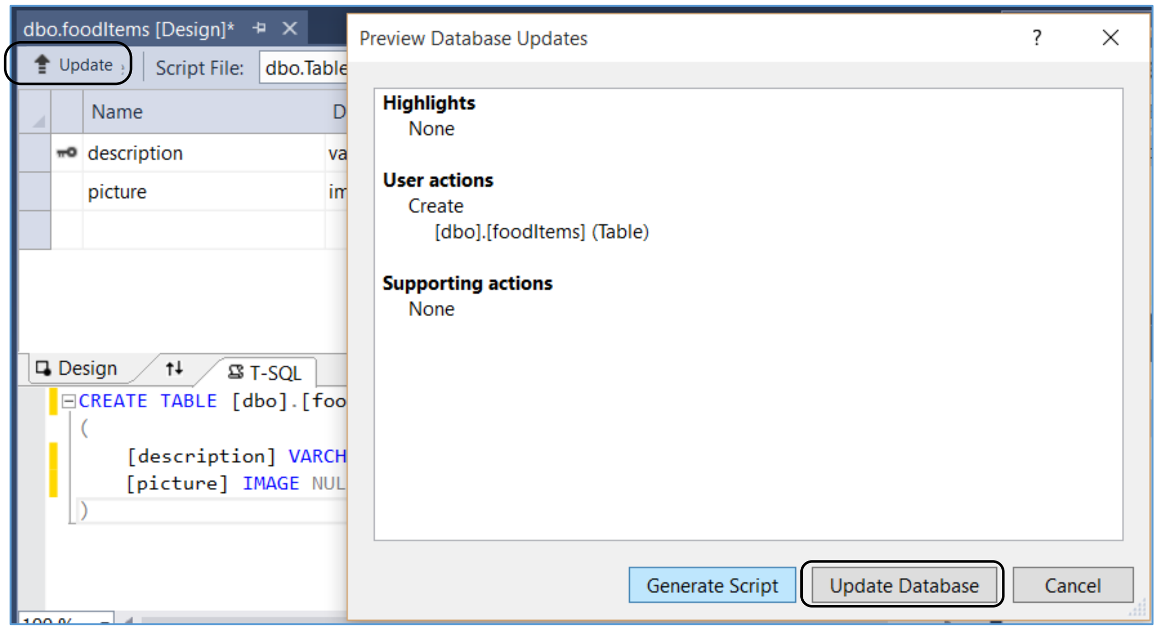
Add two fields to the table: ***description*** which will be a text field, and ***picture*** which has the data type '***image***'.



Name	Data Type	Allow Nulls	Default
description	varchar(50)	☐	
picture	image	✓	
		☐	

```
CREATE TABLE [dbo].[foodItems]
(
    [description] VARCHAR(50) NOT NULL PRIMARY KEY,
    [picture] IMAGE NULL
)
```

Go to the ***CREATE TABLE*** line and change the name of the table to '***foodItems***'. Click the '***Update***' button above the list of fields. When the ***Database Update*** window appears, click the '***Update Database***' button.



Highlights
None

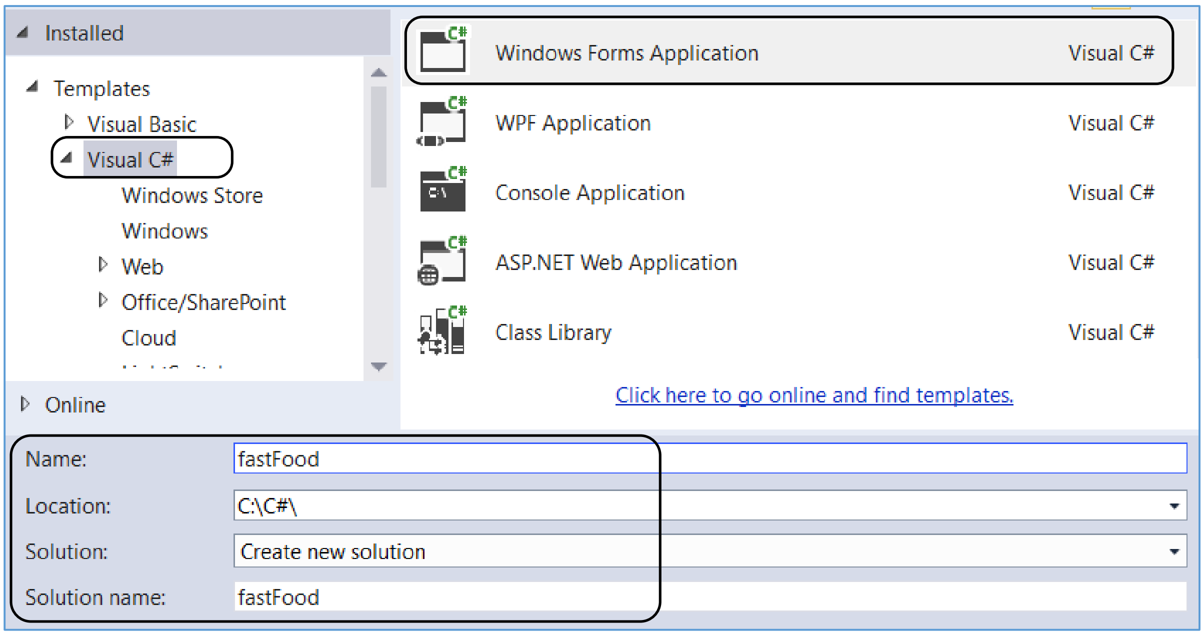
User actions
Create
[dbo].[foodItems] (Table)

Supporting actions
None

Close the ***foodItems*** table design page and re-open the ***Server Explorer*** window. Right-click on ***fastFood.mdf*** and select '***Refresh***'. Click the small arrow to the left of the ***Tables*** icon. The table which you created should now be shown.

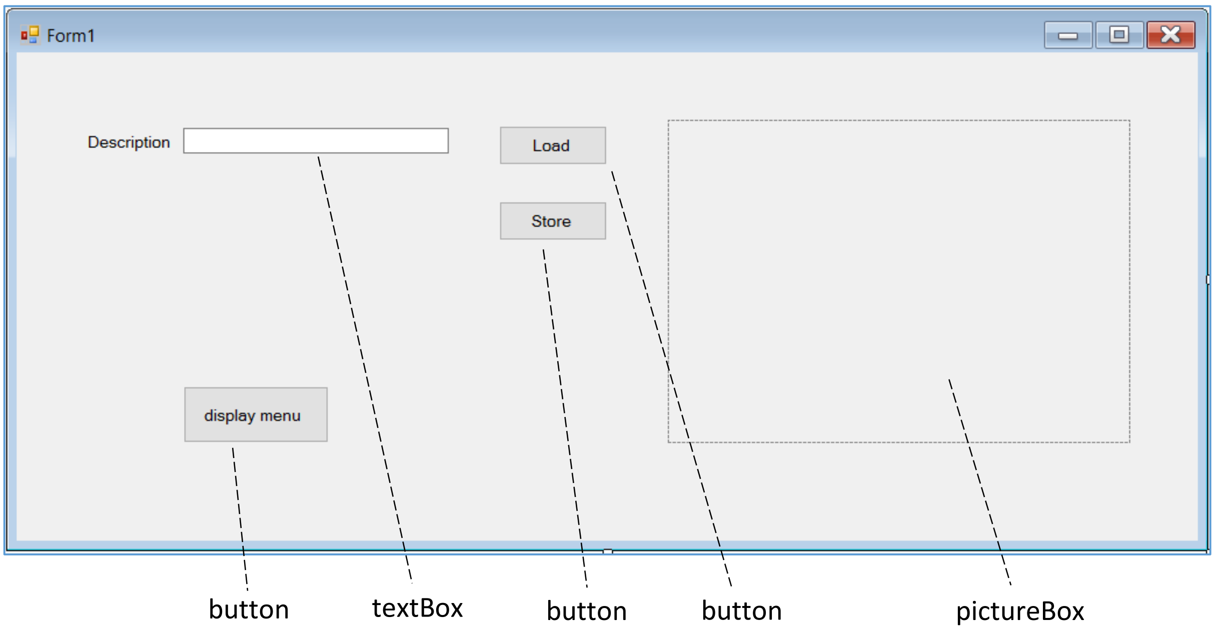
Right-click on ***fastFood.mdf*** and select the '***Delete***' option to close the connection to the database.

We can now begin work on the program to load and display the images. Start a new project, select ***Windows Forms Application***, and give this the name '***fastFood***'.


Installed
Templates
Visual Basic
Visual C#
Windows Store
Windows
Web
Office/SharePoint
Cloud
Online
Windows Forms Application Visual C#
WPF Application Visual C#
Console Application Visual C#
ASP.NET Web Application Visual C#
Class Library Visual C#
Click here to go online and find templates.

Name: fastFood
Location: C:\C#\
Solution: Create new solution
Solution name: fastFood

In the ***Form1*** design window, add the components shown:



Rename the components:

textBox	***txtDescription***
'***Store***' button	***btnStore***
'***Load***' button	***btnLoad***
'***display menu***' button	***btnDisplay***

The first step is to set up a '***Load***' button_click method which will display a file selection window. The user will then be able to select the picture that they wish to upload to the database.

Double click the '***Load***' button and add the necessary code. Notice the filter line, which displays only files which are in graphics format. When the picture is selected, it will be displayed in the pictureBox on the form.

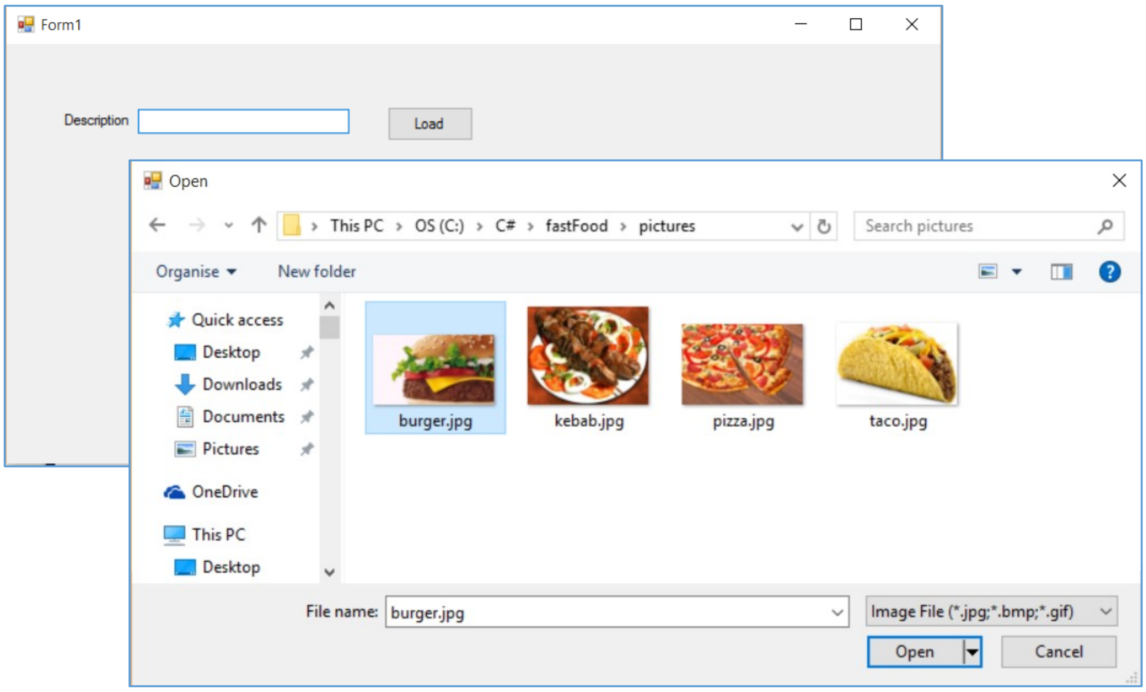
A ***TRY..CATCH*** structure is used, to prevent the program stopping if the selected file cannot be loaded for some reason.

A variable '***imagename***' is also required, to temporarily store the file name and path of the chosen picture.

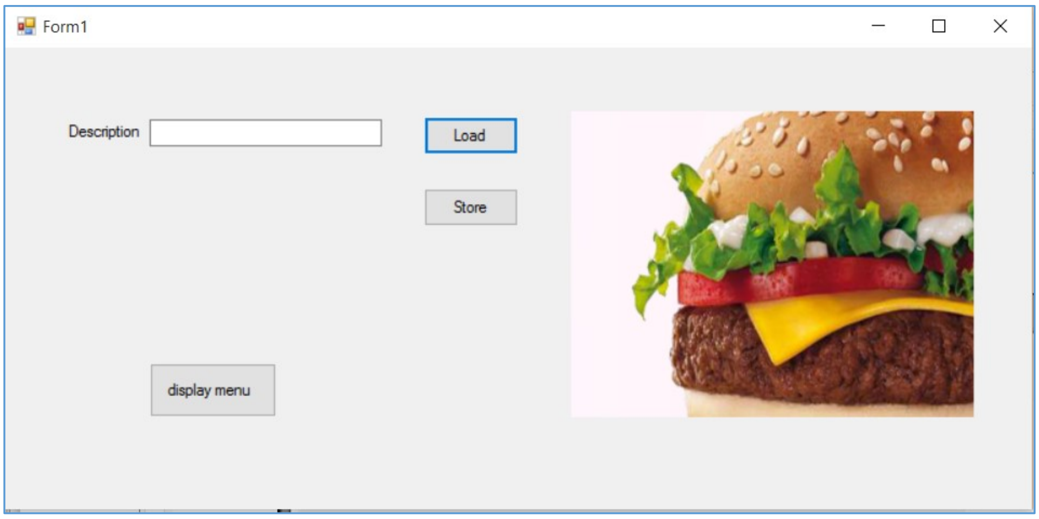
```
public partial class Form1 : Form
{
    string imagename;

    public Form1()
    {
        InitializeComponent();
    }

    private void btnLoad_Click(object sender, EventArgs e)
    {
        try
        {
            FileDialog fileDialog = new OpenFileDialog();
            fileDialog.Filter="Image File (*.jpg;*.bmp;*.gif)|*.jpg;*.bmp;*.gif";

            if (fileDialog.ShowDialog() == DialogResult.OK)
            {
                imagename = fileDialog.FileName;
                Bitmap newimg = new Bitmap(imagename);
                pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
                pictureBox1.Image = (Image)newimg;
            }

            fileDialog = null;
        }
        catch
        {
            MessageBox.Show("Error");
        }
    }

}
```

Run the program to test the image loading. Click the '***Load***' button and navigate to the folder where your images are stored.

Select an image.



Click the '***Open***' button, and the image should appear in the pictureBox on the form:



The user will enter a description of the food item in the textBox.

The next step is to save the image and description into the database table. Begin by double clicking the '***Store***' button and adding a line of code to call an ***addRecord( )*** method:

```
private void btnStore_Click(object sender, EventArgs e)
{
    addRecord();
}
```

Go to the start of the program listing and add '***using IO***' and '***using SqlClient***' directives. We also need to specify the database location.

```
using System.Text;
using System.Windows.Forms;

using System.IO;
using System.Data.SqlClient;

namespace fastFood
{
    public partial class Form1 : Form
    {
        string databaseLocation = "C:\\C#\\fastFood.mdf;";

        string imagename;
```

We will now write the ***addRecord( )*** method. Add this after ***btnStore_Click( )***.

The method uses a ***TRY..CATCH*** structure to prevent the program stopping in the event of an error during the file handling operation.

We include an ***IF***... condition to check that an image has been selected, before going any further.

The program opens the graphics file and transfers it into a storage array as bytes of binary data. This type of data is known as a '***binary large object***', or '***blob***'.

```
private void addRecord()
{
    try
    {
        if (imagename != "")
        {
            FileStream fs;
            fs = new FileStream(@imagename, FileMode.Open, FileAccess.Read);

            byte[] picbyte = new byte[fs.Length];
            fs.Read(picbyte, 0, System.Convert.ToInt32(fs.Length));
            fs.Close();
        }
    }
    catch
    {
        MessageBox.Show("File error");
    }
}
```

We now save the '**description'** field and the '**picture'** data into the database table. This requires a similar SQL command to those used in previous programs, but the '**picture'** data is identifed by a parameter **'@pic**' which tells the program where to find the binary large object array.

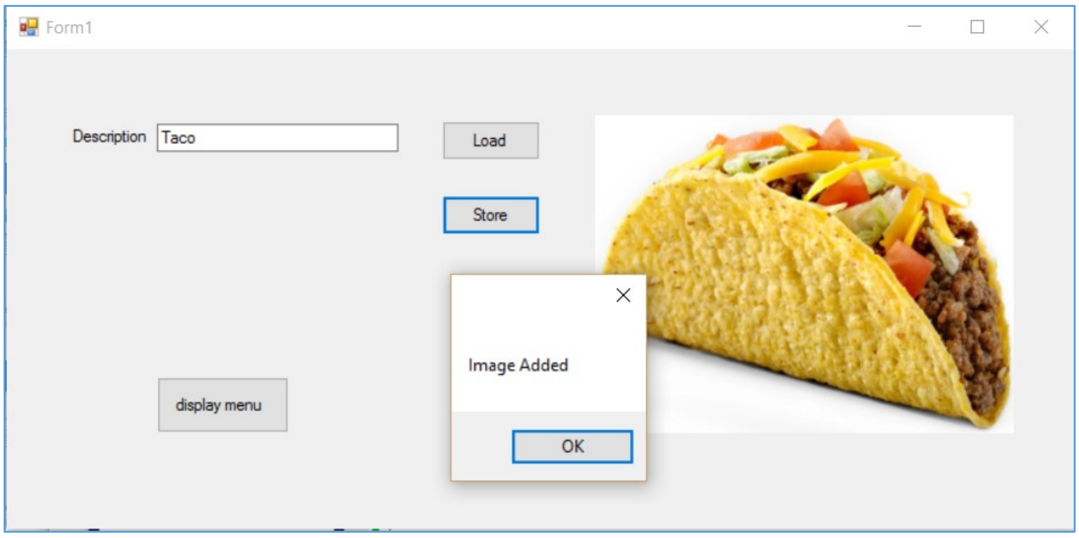
Add code to the ***addRecord( )*** method to save the record. Notice the messageBox line, which gives confirmation that the record has been saved successfully. Please note that the lines beginning:

'SqlConnection con = ' and ' ***string query = "INSERT INTO…*** '

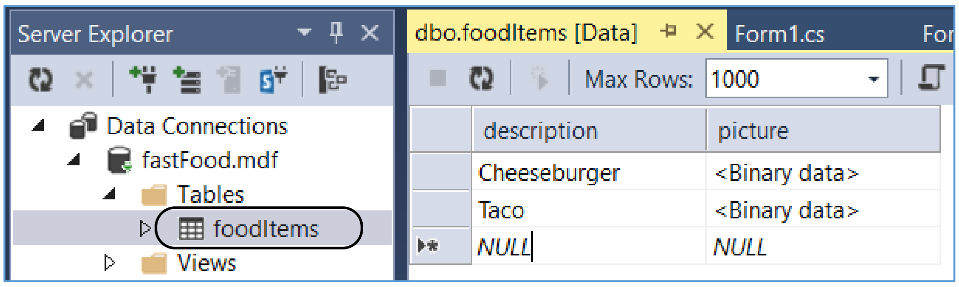
should be entered as single lines of code with no line breaks.

```
try

{
   if (imagename != "")
   {
      FileStream fs;
      fs = new FileStream(@imagename, FileMode.Open, FileAccess.Read);

      byte[] picbyte = new byte[fs.Length];
      fs.Read(picbyte, 0, System.Convert.ToInt32(fs.Length));
      fs.Close();

      SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;
         AttachDbFilename="+ databaseLocation + "Integrated Security=True;
         Connect Timeout=30; User Instance=True");
      con.Open();

      string query = "INSERT INTO foodItems(description,picture)
         VALUES('" + txtDescription.Text + "'," + " @pic)";

      SqlParameter picparameter = new SqlParameter();
      picparameter.SqlDbType = SqlDbType.Image;
      picparameter.ParameterName = "pic";
      picparameter.Value = picbyte;

      SqlCommand cmd = new SqlCommand(query, con);

      cmd.Parameters.Add(picparameter);
      cmd.ExecuteNonQuery();
      MessageBox.Show("Image Added");
      con.Close();
   }
}
catch
{
   MessageBox.Show("File error");
}
```

Run the program. Select a picture, add a description for the food item, then click the '***Store***' button. A message should appear to confirm that the record has been saved.

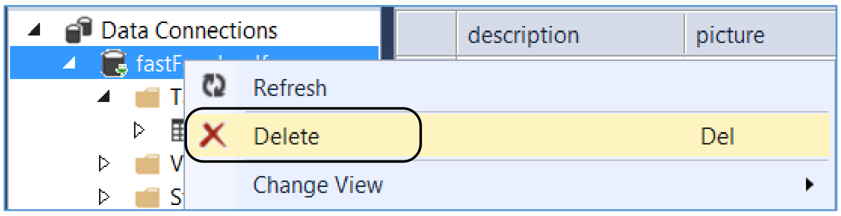


Go to the ***Server Explorer*** and add a connection to the ***fastFood.mdf*** database. Right-click on the ***foodItems*** table and select '***Show Table Data***'. Check that the records have been saved correctly. The picture fields will appear as **<Binary data>**



description	picture
Cheeseburger	<Binary data>
Taco	<Binary data>
NULL	NULL

Right-click on ***fastFood.mdf*** and select '***Delete***' to close the connection to the database.



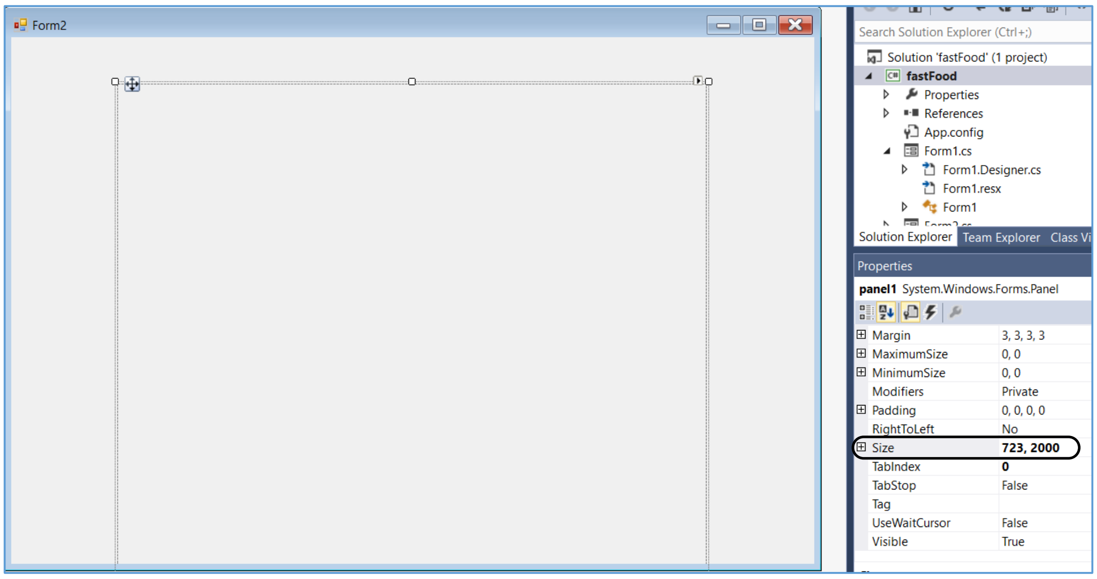
We will now set up another form which will display the images and descriptions of the food items as a Menu for the fast food shop.

Go to the ***Solution Explorer*** and create another ***Windows Form***. You can leave the name as '***Form2***'.

Double click the '***display menu***' button on ***Form1*** and add lines of code to the button_click method to open ***Form2***.

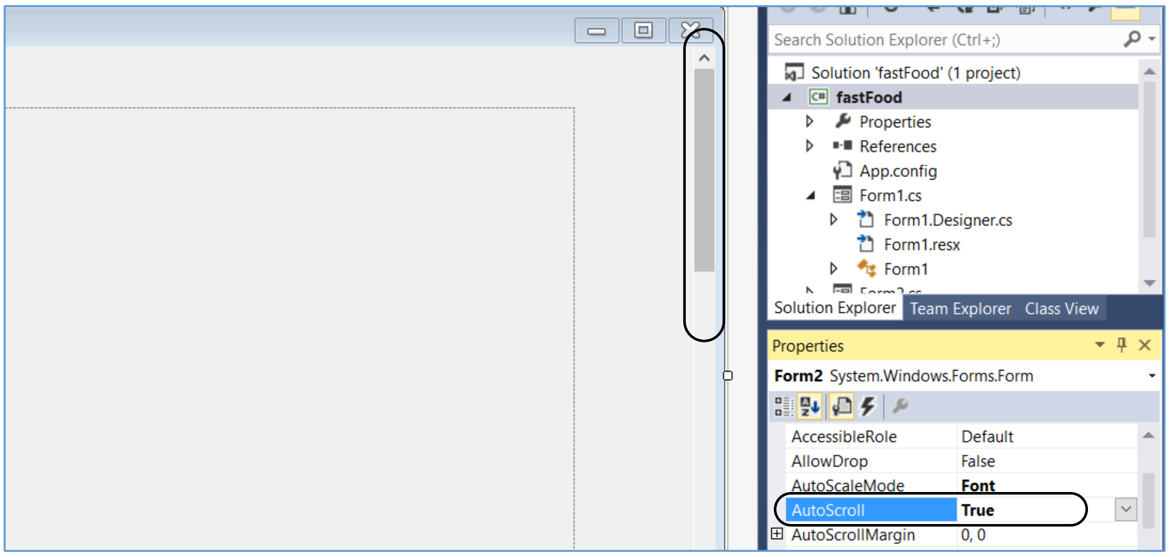
```
private void btnDisplay_Click(object sender, EventArgs e)
{
    Form2 frmForm2 = new Form2();
    frmForm2.ShowDialog();
}
```

Place a ***panel*** on ***Form2***. Click the panel once to select it, then change the height to ***2,000 pixels*** by altering the second number value of the '***Size***' entry in the ***Properties*** window:



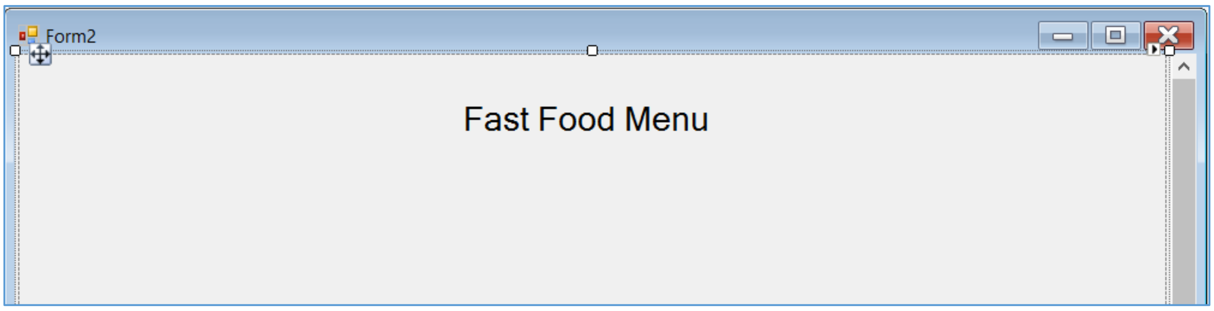
Margin	3, 3, 3, 3
MaximumSize	0, 0
MinimumSize	0, 0
Modifiers	Private
Padding	0, 0, 0, 0
RightToLeft	No
Size	**723, 2000**
TabIndex	**0**
TabStop	False
Tag	
UseWaitCursor	False
Visible	True

Select ***Form2*** by clicking outside the panel. Set the '***AutoScroll***' property to '***True***'. Notice that a vertical scroll bar now appears on the edge of the form.



AccessibleRole	Default
AllowDrop	False
AutoScaleMode	Font
AutoScroll	True
AutoScrollMargin	0, 0

Adjust the width of the panel to just reach the scroll bar. Complete the screen layout by adding a label '***Fast Food Menu***'.



We can now write methods to load and diplay the descriptions and pictures of the food items.

Go to the program code window for ***Form2***. Add '***using IO***' and '***using SqlClient***' directives, give the database location, and add the variables '***dAdapt***' and '***dSet***':

```
using System.Text;
using System.Windows.Forms;

using System.IO;
using System.Data.SqlClient;

namespace fastFood
{
    public partial class Form2 : Form
    {
       string databaseLocation = "C:\\C#\\fastFood.mdf;";
       SqlDataAdapter dAdapt;
       DataSet dSet;
```

Add a ***Connection( )*** method which will access the database table and transfer the records into the storage area called '***dTable***'. Call the ***Connection( )*** method from the ***Form2( )*** method.

```
public Form2()
{
    InitializeComponent();

    Connection();
}

private void Connection()
{
    try
    {
        SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS;
            AttachDbFilename="+ databaseLocation+ "Integrated Security=True;
            Connect Timeout=30; User Instance=True");
        con.Open();
        dAdapt = new SqlDataAdapter();
        dAdapt.SelectCommand = new SqlCommand("SELECT * FROM foodItems", con);
        dSet = new DataSet("dSet");
        dAdapt.Fill(dSet);
        con.Close();
        DataTable dTable;
        dTable = dSet.Tables[0];
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Run the program. Click the '***display menu***' button and check that ***Form2*** opens without any error message. The ***panel*** on Form2 should scroll vertically, but no images are displayed yet. We will deal with the display method next.

Set up a ***displayPictures( )*** method, and call this from the ***Form2( )*** method. We will create arrays for up to eight pictures and labels to be added when the program runs.

```
public Form2()
{
    InitializeComponent();
    Connection();

    displayPictures();
}

private void displayPictures()
{
    DataTable dataTable = dSet.Tables[0];
    int countRecords = dataTable.Rows.Count;
    PictureBox[] pictureBox = new PictureBox[8];
    Label[] label = new Label[8];
}
```

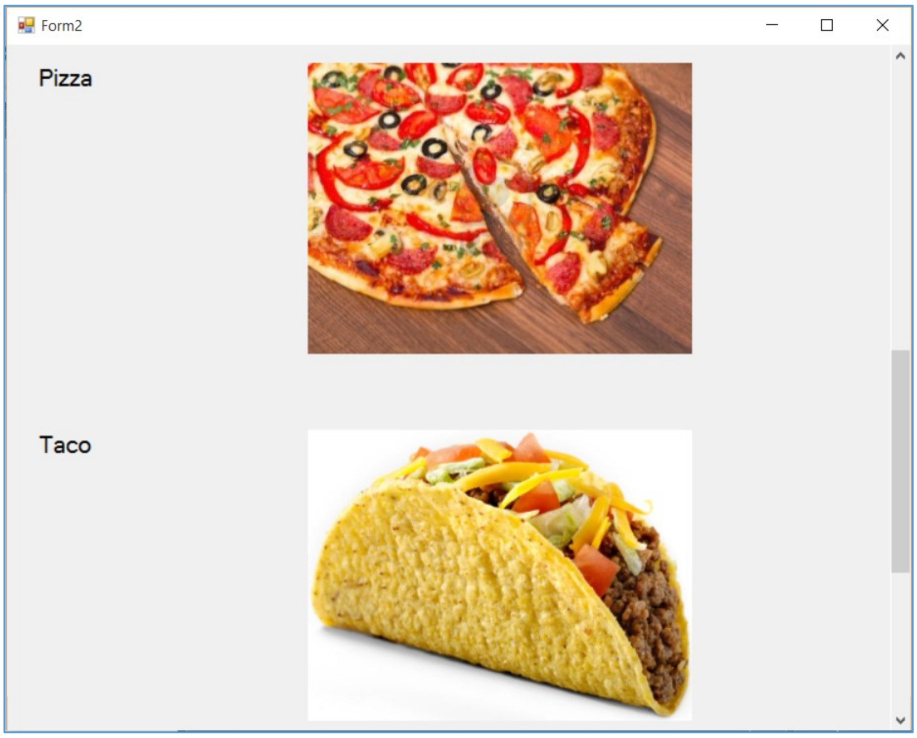
Create a loop which will repeat for each record in the database table. Within the loop, the program collects the ***binary large object*** from the data record and converts it back into an ***image file***.

```
private void displayPictures()
{
    DataTable dataTable = dSet.Tables[0];
    int countRecords = dataTable.Rows.Count;
    PictureBox[] pictureBox = new PictureBox[8];
    Label[] label = new Label[8];

    Random r = new Random();
    for (int i = 0; i < countRecords; i++)
    {
       DataRow dataRow = dataTable.Rows[i];
       int random = r.Next(1,1000);
       string finalString = "pic" + Convert.ToString(random);
       FileStream FS1 = new FileStream(finalString + ".jpg", FileMode.Create);
       byte[] blob = (byte[])dataRow[1];
       FS1.Write(blob, 0, blob.Length);
       FS1.Close();
       FS1 = null;
    }
}
```

The final step is to display the description of the food item and its picture image on the panel. The vertical position is moved down by 300 pixels as each record is displayed.

```
for (int i = 0; i < countRecords; i++)
{
    DataRow dataRow = dataTable.Rows[i];
    int random = r.Next(1,1000);
    string finalString = "pic" + Convert.ToString(random);
    FileStream FS1 = new FileStream(finalString + ".jpg", FileMode.Create);
    byte[] blob = (byte[])dataRow[1];
    FS1.Write(blob, 0, blob.Length);
    FS1.Close();
    FS1 = null;

    pictureBox[i] = new PictureBox();
    pictureBox[i].Image = Image.FromFile(finalString + ".jpg");
    pictureBox[i].Size = new System.Drawing.Size(314, 238);
    pictureBox[i].Location = new System.Drawing.Point(244, 134 + 300 * i);
    pictureBox[i].SizeMode = PictureBoxSizeMode.StretchImage;
    panel1.Controls.Add(pictureBox[i]);
    pictureBox[i].Refresh();

    label[i] = new Label();
    label[i].Size = new System.Drawing.Size(200, 60);
    label[i].Text = Convert.ToString(dataRow[0]);
    label[i].Font = new System.Drawing.Font("Microsoft Sans Serif", 14F,
                         System.Drawing.FontStyle.Regular,
                         System.Drawing.GraphicsUnit.Point, ((byte)(0)));
    label[i].Location = new System.Drawing.Point(20, 134 + 300 * i);

    panel1.Controls.Add(label[i]);
}
```

Run the program and click the ‘***display menu***’ button. Form2 should open and display the food items in a scrolling list.


Form2
Pizza
Taco