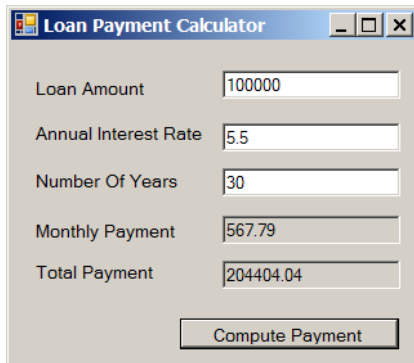


Supplement: Visual C++ GUI Applications (VS2008, VS2010)
For Introduction to C++ Programming
By Y. Daniel Liang

1 Introduction

This supplement introduces how to develop GUI applications using Microsoft Visual C++ 2008 or 2010.

We will create a loan calculator application with the graphical user interface, as shown in Figure 1.



Field	Value
Loan Amount	100000
Annual Interest Rate	5.5
Number Of Years	30
Monthly Payment	567.79
Total Payment	204404.04

Figure 1

The application lets the user enter loan information and compute the monthly payment and total payment.

2 Creating GUI Projects

To create a GUI application using Visual C++, you have to first create a Windows Forms Application project. Here are the steps to create a GUI project named GUIDemo.

1. Choose *File, New, Project* to display the New Project dialog box, as shown in Figure 2.
2. Select CRL in the Project types pane, and Windows Forms Application in the Templates pane.
3. Enter GUIDemo in the Name field and click *OK* to create the project, as shown in Figure 3.

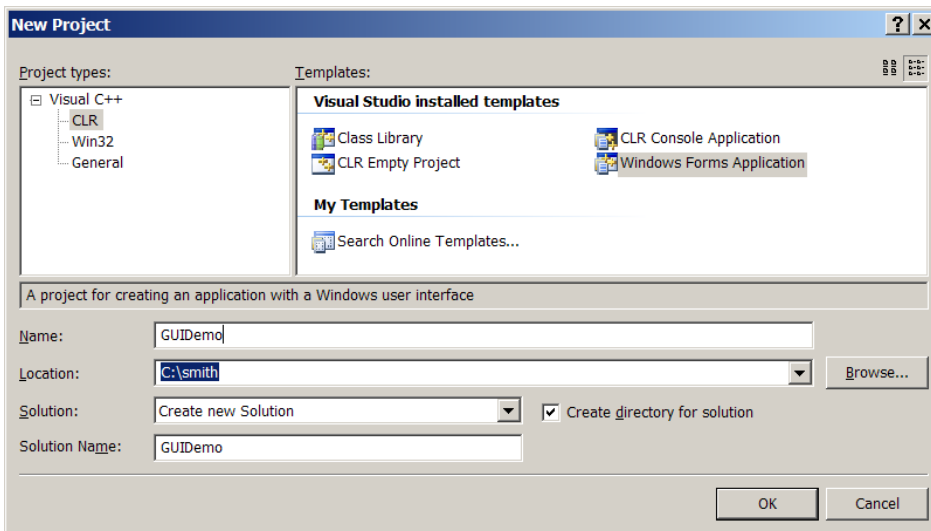


Figure 2

You need to create a Windows Forms Application to develop GUI programs in Visual C++.

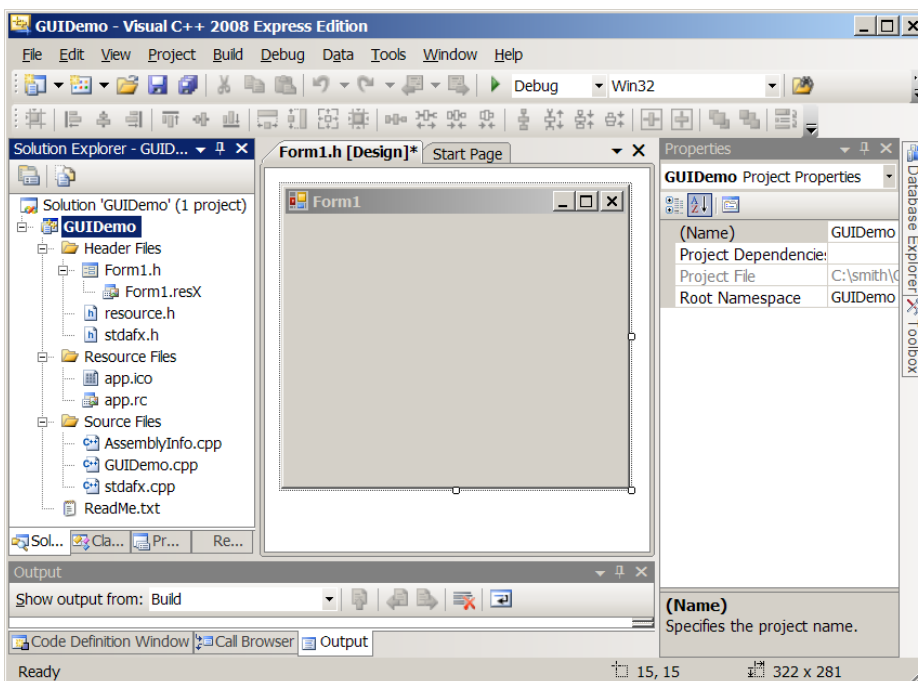


Figure 3

The new project wizard creates a form and other files in the project.

The project is created with a graphical form and several supporting files. The form is for creating the graphical user interface. You can add GUI components such as labels, buttons, and text boxes into the form. The project also

contains several files such as GUIDemo.cpp and Form1.h. The code in these files is automatically generated.

3 Designing GUI

Visual C++ provides an intuitive drag and drop feature for creating GUI user interface. To add a GUI component to the form, simply drag the component from the Toolbox to the form. Here are the steps to create the GUI for the loan calculator project:

1. Click on Label in the Toolbox and drop it in the form, as shown in Figure 4. In the Property window for the label, change the Text property to Loan Amount. Similarly drop four more labels to the form and change their Text properties to Annual Interest Rate, Number Of Years, Monthly Payment, and Total Payment, as shown in Figure 5.

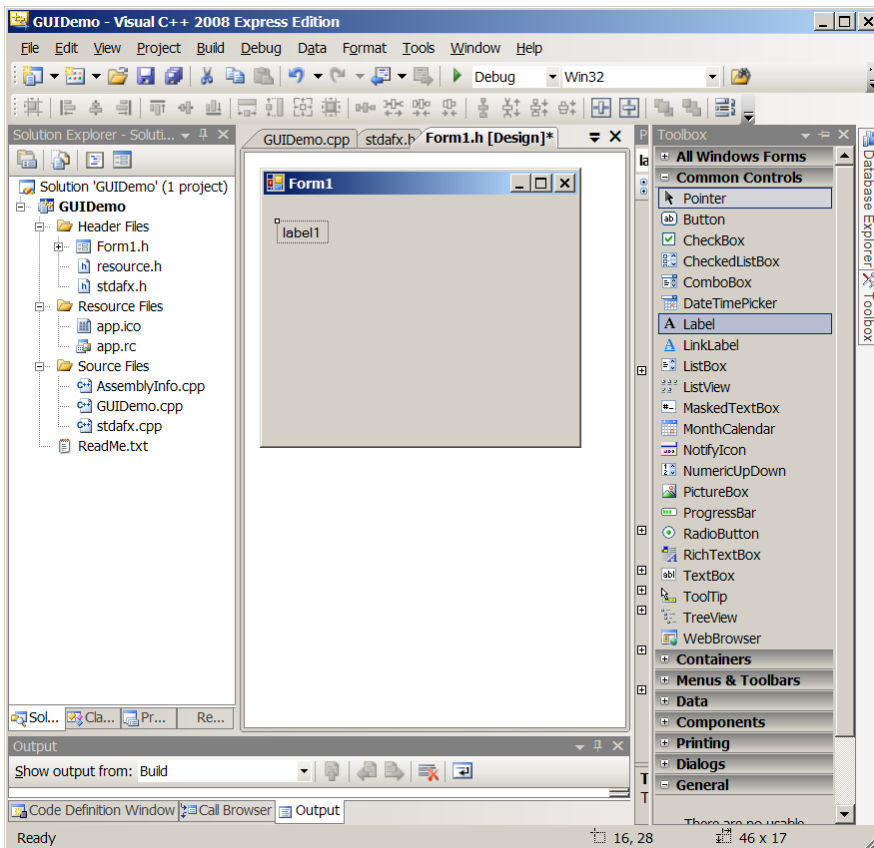


Figure 4

You can add the components from the Toolbox to the form.

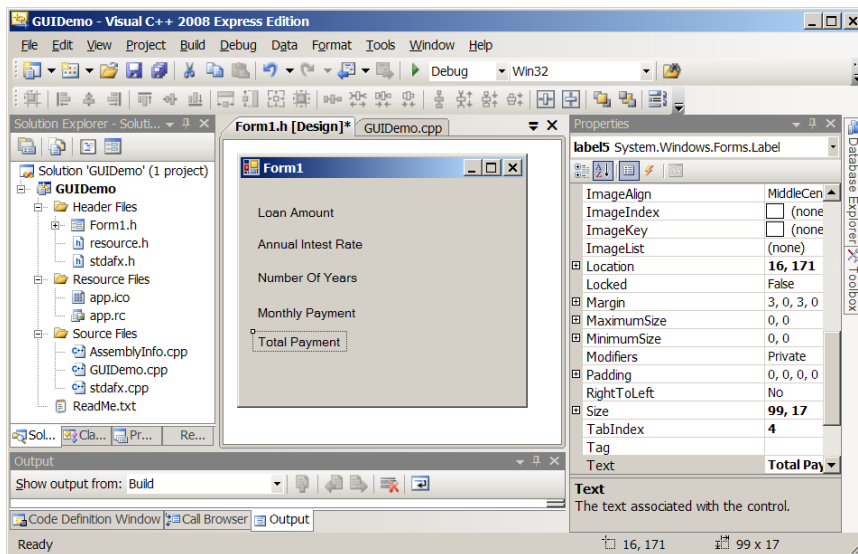


Figure 5

Five labels are added in the form.

2. Click on TextBox in the Toolbox and drop it in the form, as shown in Figure 6. In the Property window for the text box, change the Name property to tbLoanAmount. Similarly drop four more text boxes to the form and change their Name properties to tbAnnualInterestRate, tbNumberOfYears, btMonthlyPayment, and tbTotalPayment, as shown in Figure 7.
3. Click on Button in the Toolbox and drop it in the form, as shown in Figure 8. In the Property window for the button, change the Name property to btComputePayment and the Text property to Compute Payment.

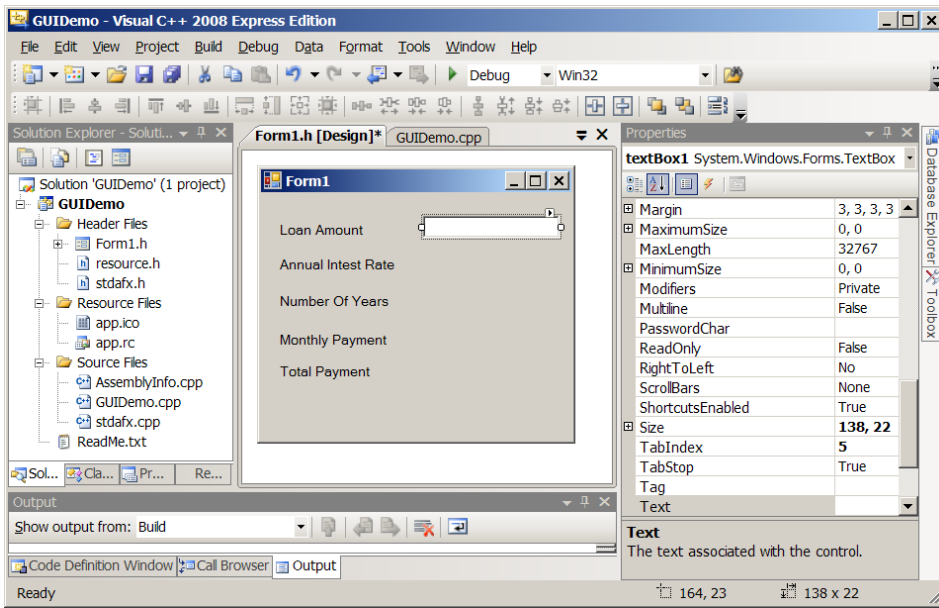


Figure 6
A TextBox is added to the form.

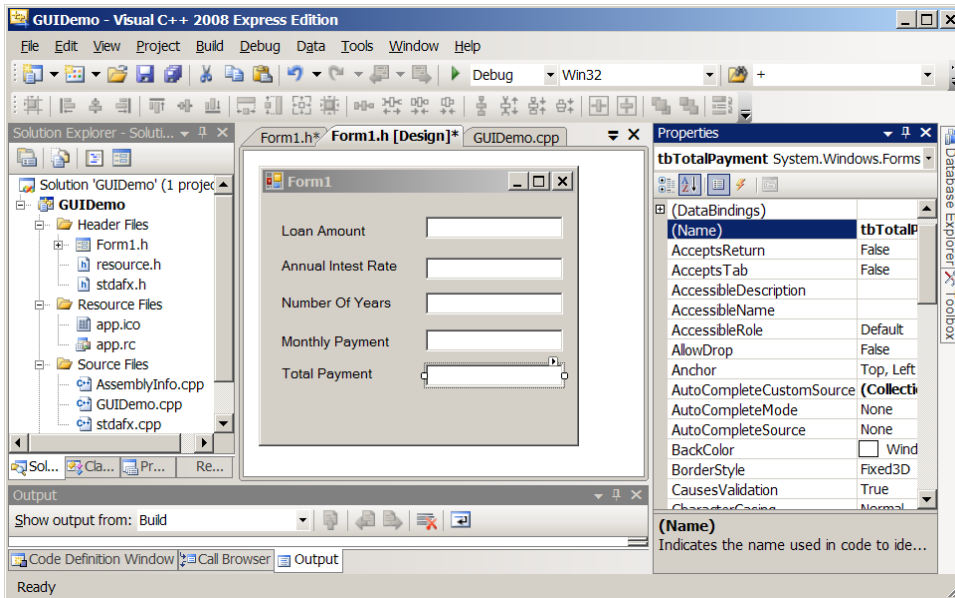


Figure 7
Five text boxes are added to the form.

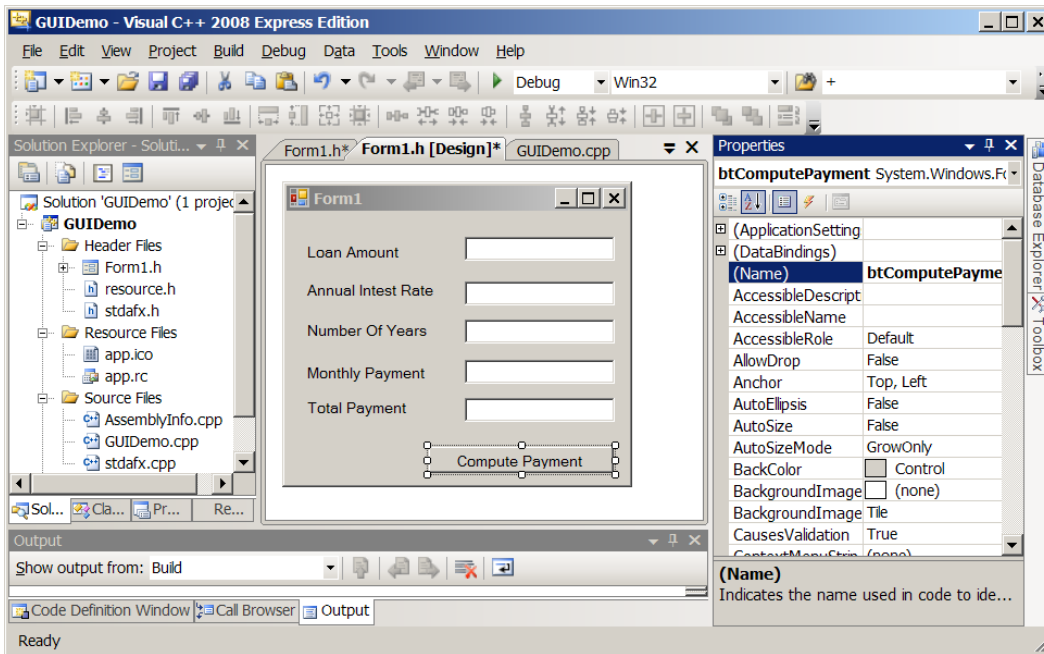


Figure 8
A button is added to the form.

The information about form is stored in Form1.h. Whenever you change the form in the visual designer, the corresponding code in Form1.h is also changed, as shown in Figure 9. For example, the information on the Loan Amount label is shown in lines 76-81.

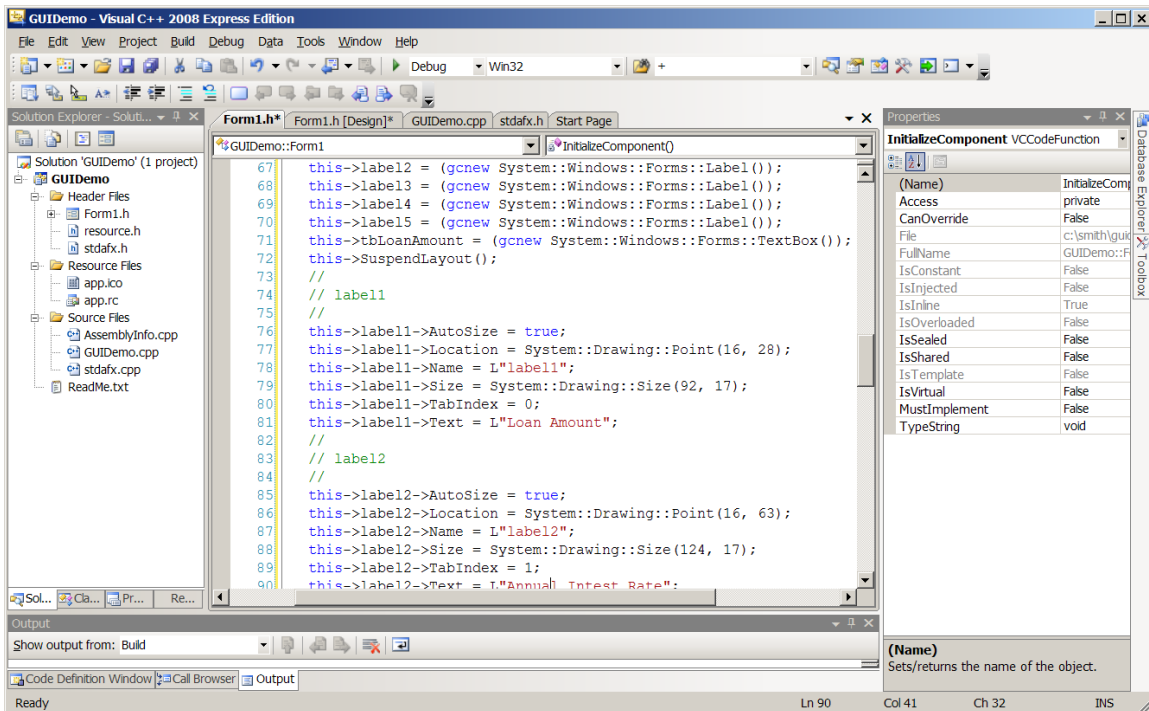


Figure 9

Form1.h contains the code that describes for the form.

4 Handling Events

When you run a GUI program, the program interacts with the user, and the events drive its execution. An event can be defined as a signal to the program that something has happened. Events are triggered either by external user actions, such as mouse movements, button clicks, and keystrokes, or by internal program activities, such as a timer. The program can choose to respond to or ignore an event.

The component for which an event is originated is called the *source component*. For example, a button is the source object for a button-clicking action event. A function that performs a task to respond to the event is called an *event handler*.

For our loan calculator GUI, when the user clicks the *Compute Payment* button, the handler gathers the loan information from the text boxes, computes the monthly payment and total payment, and displays them in the respective text boxes.

Here are the steps to create a handler for the button-click event on the *Compute Payment* button:

1. Choose the Event tab in the Property window for *btComputePayment*, as shown in Figure 10.
2. A button source component can handle many events. The one we need for this application is the click event. In the empty text box on the right of Click, press the Enter key. You will see the following handler is automatically generated in *Form1.h*, as shown in Figure 11.

```
private: System::Void btComputePayment_Click(System::Object^ sender,  
    System::EventArgs^ e)  
{  
  
}
```

3. Write the code to compute and display monthly and total payment as follows:

```
private: System::Void btComputePayment_Click(System::Object^ sender,  
    System::EventArgs^ e)  
{  
    double loanAmount = System::Convert::ToDouble(tbLoanAmount->Text);  
    double annualInterestRate  
        = System::Convert::ToDouble(tbAnnualInterestRate->Text);  
    double numberOfYears  
        = System::Convert::ToDouble(tbNumberOfYears->Text);
```

```

double monthlyInterestRate = annualInterestRate / 1200;

double monthlyPayment = loanAmount * monthlyInterestRate / (1
- 1 / pow(1 + monthlyInterestRate, numberOfYears * 12));
double totalPayment = monthlyPayment * numberOfYears * 12;

monthlyPayment = floor(monthlyPayment * 100 + 0.5) / 100;
totalPayment = floor(totalPayment * 100 + 0.5) / 100;

tbMonthlyPayment->Text = System::Convert::ToString(monthlyPayment);
tbTotalPayment->Text = System::Convert::ToString(totalPayment);
}

```

4. You can also set a title for the form by setting the Text property of the form to Loan Payment Calculator. You should set the ReadOnly property for the monthly payment and total payment text boxes to True to prevent the user from editing these text boxes.
5. You can now run the program. The output is shown in Figure 1.

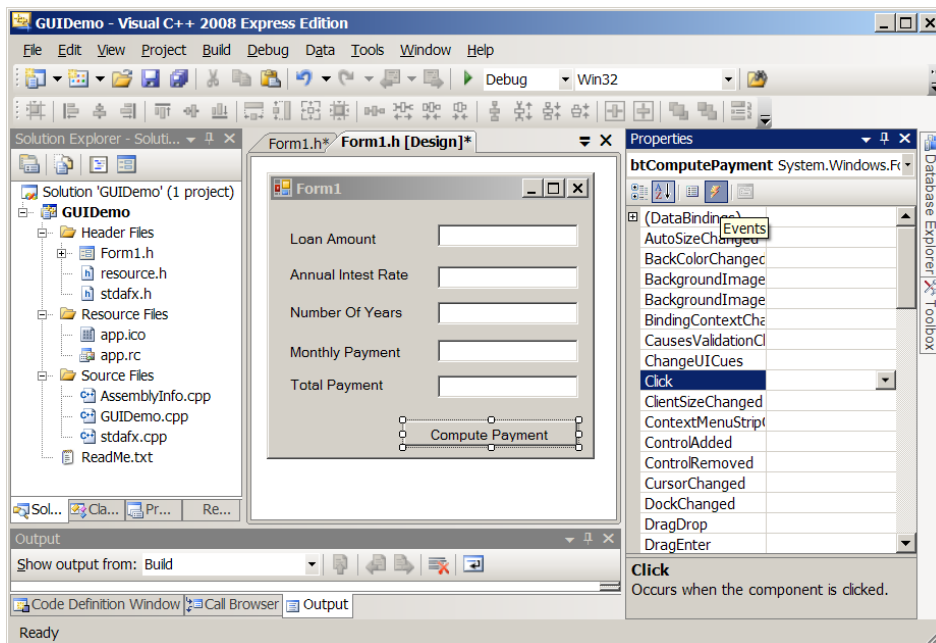


Figure 10

You can generate event handlers from the Event tab of the Properties window.

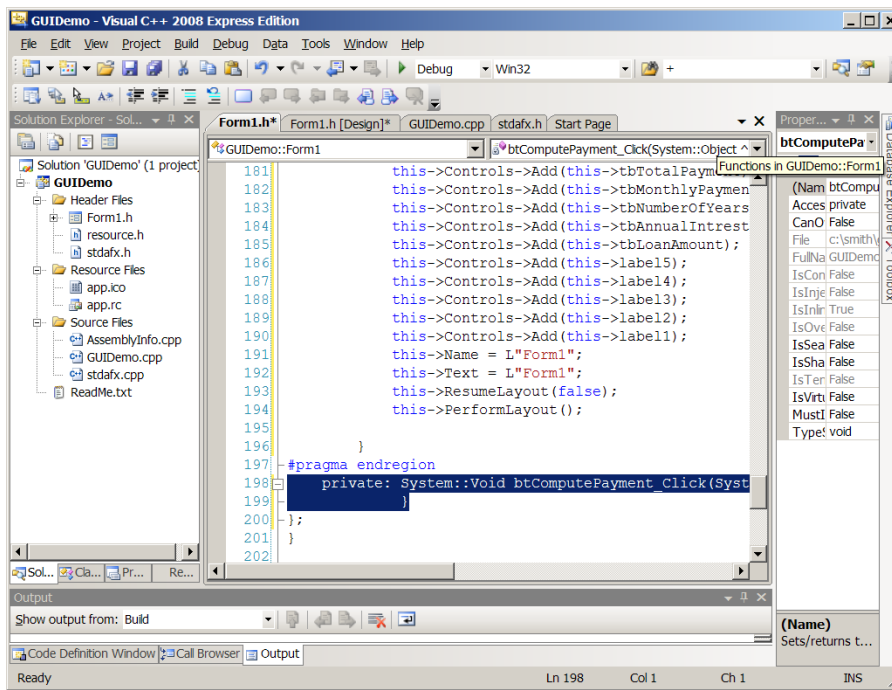


Figure 11

The function for handling the event is automatically generated in Form1.h.