

Forum:[Hack Forums](#)

Thread: [\[C++\] Creating a stub & builder \(video\)](#)

_Frosty - January 11, 2015, 11:20 am

Hey HF,

I released a video tutorial previously on how to make a stub and builder in C#. One of my friends decided to re-code my tutorial in C++ for HF. You can find my original C# tutorial here - hackforums.net/showthread.php?tid=4623639. I've posted his screen-casting on my channel, enjoy!

Stub (video) - youtube.com/watch?v=gb_xkgxGM68

Builder (video) - youtube.com/watch?v=iWo7a6Stg48

Stub - main.cpp

```
Code #include <iostream>
```

```
#include <fstream>
```

```
#include <Windows.h>
```

```
using namespace std;
```

```
struct Settings {
```

```
    int port;
```

```
    char url[256];
```

```
};
```

```
int main() {
```

```
    cout << "Stub\n\n\n";
```

```
    long int filesize = 93184;
```

```
    Settings data;
```

```
    int size = MAX_PATH;
```

```
    char *filename = new char[size];
```

```
    GetModuleFileNameA(NULL, filename, size);
```

```
    ifstream file(filename, ios::binary);
```

```
    file.seekg(filesize);
```

```
    file.read((char*)&data, sizeof(Settings));
```

```
    file.close();
```

```
    cout << data.url << ":" << data.port << endl << endl;
```

```
    system("pause");
```

```
    return 0;
```

```
}
```

Builder - main.cpp

```
Code #include <iostream>
#include <Windows.h>
```

```
#include "MyForm.h"
```

```
using namespace System;
using namespace System::Windows::Forms;
```

```
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInst, char *cmdLine, int
nCmdShow) {
    Application::EnableVisualStyles();
    Application::Run(gcnew Builder::MyForm());
}
```

Builder - form.h

```
Code #include <iostream>
#include <string>
#include <fstream>
#include <msclr\marshal_cppstd.h>
```

```
struct Settings {
    int port;
    char url[256];
};
```

```
namespace Builder {
```

```
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
using namespace std;
```

```
/// <summary>
/// Summary for MyForm
/// </summary>
public ref class MyForm : public System::Windows::Forms::Form
{
public:
    MyForm(void)
    {
        InitializeComponent();
        //
        //TODO: Add the constructor code here
        //
    }
}
```

```
protected:
    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    ~MyForm()
```

```

{
    if (components)
    {
        delete components;
    }
}
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::Button^ button1;
protected:

```

```

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container ^components;

```

#pragma region Windows Form Designer generated code

```

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        this->label1 = (gcnew System::Windows::Forms::Label());
        this->label2 = (gcnew System::Windows::Forms::Label());
        this->textBox1 = (gcnew System::Windows::Forms::TextBox());
        this->textBox2 = (gcnew System::Windows::Forms::TextBox());
        this->button1 = (gcnew System::Windows::Forms::Button());
        this->SuspendLayout();
        //
        // label1
        //
        this->label1->AutoSize = true;
        this->label1->Location = System::Drawing::Point(13, 13);
        this->label1->Name = L"label1";
        this->label1->Size = System::Drawing::Size(32, 13);
        this->label1->TabIndex = 0;
        this->label1->Text = L"URL:";
        this->label1->Click += gcnew System::EventHandler(this, &MyForm::label1_Click);
        //
        // label2
        //
        this->label2->AutoSize = true;
        this->label2->Location = System::Drawing::Point(13, 35);
        this->label2->Name = L"label2";
        this->label2->Size = System::Drawing::Size(29, 13);
        this->label2->TabIndex = 1;
        this->label2->Text = L"Port:";
        //
        // textBox1

```

```

//
this->textBox1->Location = System::Drawing::Point(52, 13);
this->textBox1->Name = L"textBox1";
this->textBox1->Size = System::Drawing::Size(230, 20);
this->textBox1->TabIndex = 2;
//
// textBox2
//
this->textBox2->Location = System::Drawing::Point(52, 35);
this->textBox2->Name = L"textBox2";
this->textBox2->Size = System::Drawing::Size(230, 20);
this->textBox2->TabIndex = 3;
//
// button1
//
this->button1->Location = System::Drawing::Point(16, 62);
this->button1->Name = L"button1";
this->button1->Size = System::Drawing::Size(266, 39);
this->button1->TabIndex = 4;
this->button1->Text = L"Build";
this->button1->UseVisualStyleBackColor = true;
this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1_Click);
//
// MyForm
//
this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);
this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(294, 109);
this->Controls->Add(this->button1);
this->Controls->Add(this->textBox2);
this->Controls->Add(this->textBox1);
this->Controls->Add(this->label2);
this->Controls->Add(this->label1);
this->Name = L"MyForm";
this->Text = L"MyForm";
this->ResumeLayout(false);
this->PerformLayout();

}
#pragma endregion
private: System::Void label1_Click(System::Object^ sender, System::EventArgs^ e) {
}
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    Settings data;

    String ^portStr = textBox2->Text;
    String ^urlStr = textBox1->Text;

    string stdUrlStr = msclr::interop::marshal_as<string>(urlStr);
    string stdPortStr = msclr::interop::marshal_as<string>(portStr);

    data.port = atoi(stdPortStr.c_str());
    strcpy(data.url, stdUrlStr.c_str());
}

```

```

ifstream stub("Stub.exe", ios::binary);
ofstream built("Built.exe", ios::binary);

built << stub.rdbuf();
stub.close();

built.write((char*)&data, sizeof(Settings));
built.close();

MessageBox::Show("Built!", "Success");
}
};
}

```

matty9 - January 12, 2015, 4:05 pm

Nice tutorial, I'm sure this will help out a lot of people.

tqmedia - January 12, 2015, 4:52 pm

Very usefull tutorial. There is lots of good information for me

?lixir - January 12, 2015, 4:57 pm

Is there something equivalent to codedom in c++?

HelloMyNameIsWalter - January 12, 2015, 8:52 pm

(Jan 12, 2015 05:57 PM)?lixir Wrote: [\[CITE\]](#) Is there something equivalent to codedom in c++?
Yes, codedom.

[http://msdn.microsoft.com/en-us/library/y2k85ax6\(v=vs.110\).aspx?cs-save-lang=1&cs-lang=cpp#code-snippet-1](http://msdn.microsoft.com/en-us/library/y2k85ax6(v=vs.110).aspx?cs-save-lang=1&cs-lang=cpp#code-snippet-1)

?lixir - January 13, 2015, 5:18 am

Codedom requires the .net framework. I need something that can compile without dependencies.

_Frosty - January 13, 2015, 1:44 pm

(Jan 13, 2015 06:18 AM)?lixir Wrote: [\[CITE\]](#) Codedom requires the .net framework. I need something that can compile without dependencies. Hi Elixir,

Are you just looking to compile small C code? I like the TinyC compiler, great for malware. Let me know if you need help / thats not what you're looking for :P

HelloMyNameIsWalter - January 13, 2015, 2:25 pm

(Jan 13, 2015 06:18 AM)?lixir Wrote: [\[CITE\]](#) Codedom requires the .net framework. I need something that can compile without dependencies.

Haven't heard of a library like that, however you could always whip one up yourself.

Not2EXceL - January 13, 2015, 3:34 pm

The CLI is real with this one. Seriously take the CLI out.

?lpha - December 27, 2015, 4:33 am

Instead of setting the filesize statically you can also get it dynamically

Replace stub with:

```

Code #include <iostream>
#include <fstream>

#include <Windows.h>

using namespace std;

struct Settings {
int port;
char url[256];
};

int main() {
cout << "Stub\n\n\n";

long int filesize;
Settings data;

int size = MAX_PATH;
char *filename = new char[size];

GetModuleFileNameA(NULL, filename, size);

ifstream file(filename, ifstream::ate | ios::binary);
filesize = file.tellg();
file.seekg(filesize);
file.read((char*)&data, sizeof(Settings));
file.close();

cout << data.url << ":" << data.port << endl << endl;

system("pause");

return 0;
}

```

I changed the following lines

Code long int filesize;

```

ifstream file(filename, ifstream::ate | ios::binary);
filesize = file.tellg();

```

More info about reading files: <http://www.cplusplus.com/doc/tutorial/files/>