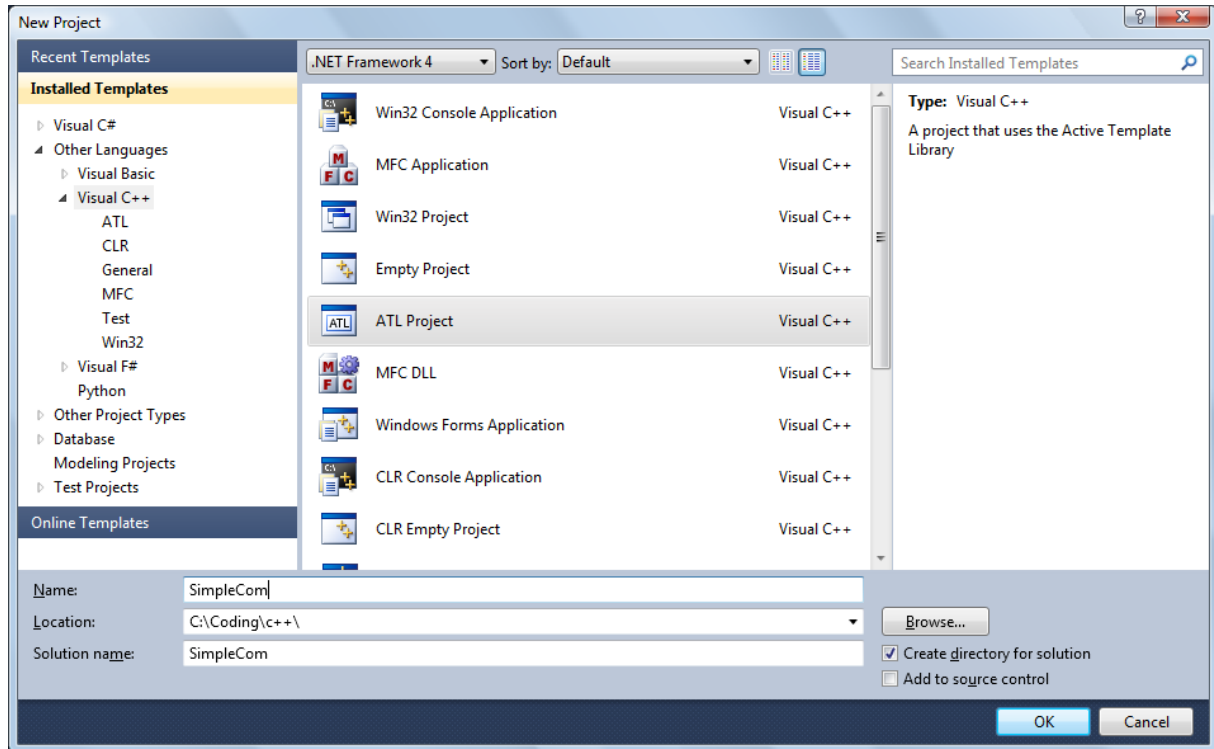
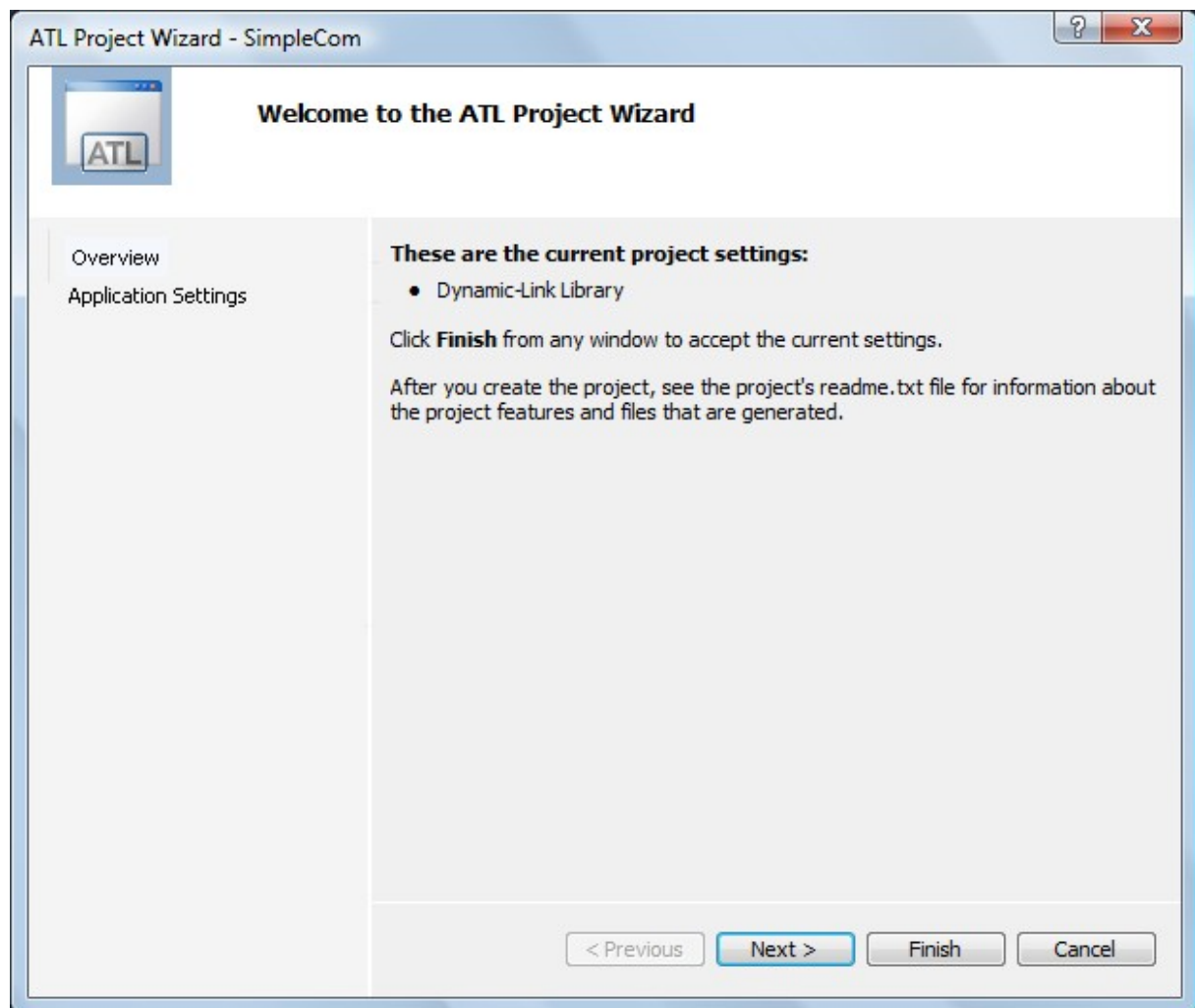


„Programowanie COM”

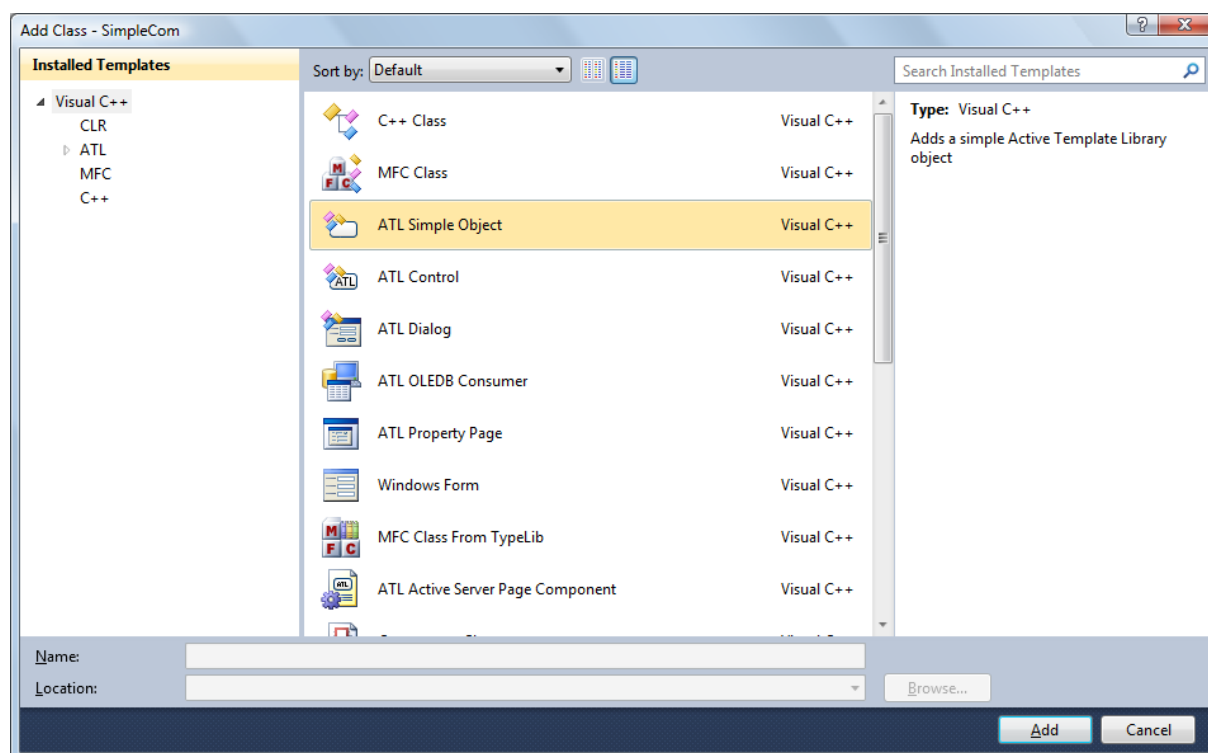
1. Otwieramy nowy projekt.



2. Klikamy Finish.



3. Dodajemy nową klasę (Shift + Alt + C)



4. Wpisujemy nazwę klasy tylko w pierwsze pole, reszta uzupełni się sama. Klikamy Next.

ATL Simple Object Wizard - SimpleCom

Welcome to the ATL Simple Object Wizard

Names
File Type Options
Options

C++

Short name: SimpleChatServer .h file: SimpleChatServer.h ...

Class: CSimpleChatServer .cpp file: SimpleChatServer.cpp ...

COM

Coclass: SimpleChatServer Type: SimpleChatServer Class

Interface: ISimpleChatServer ProgID:

< Previous Next > Finish Cancel

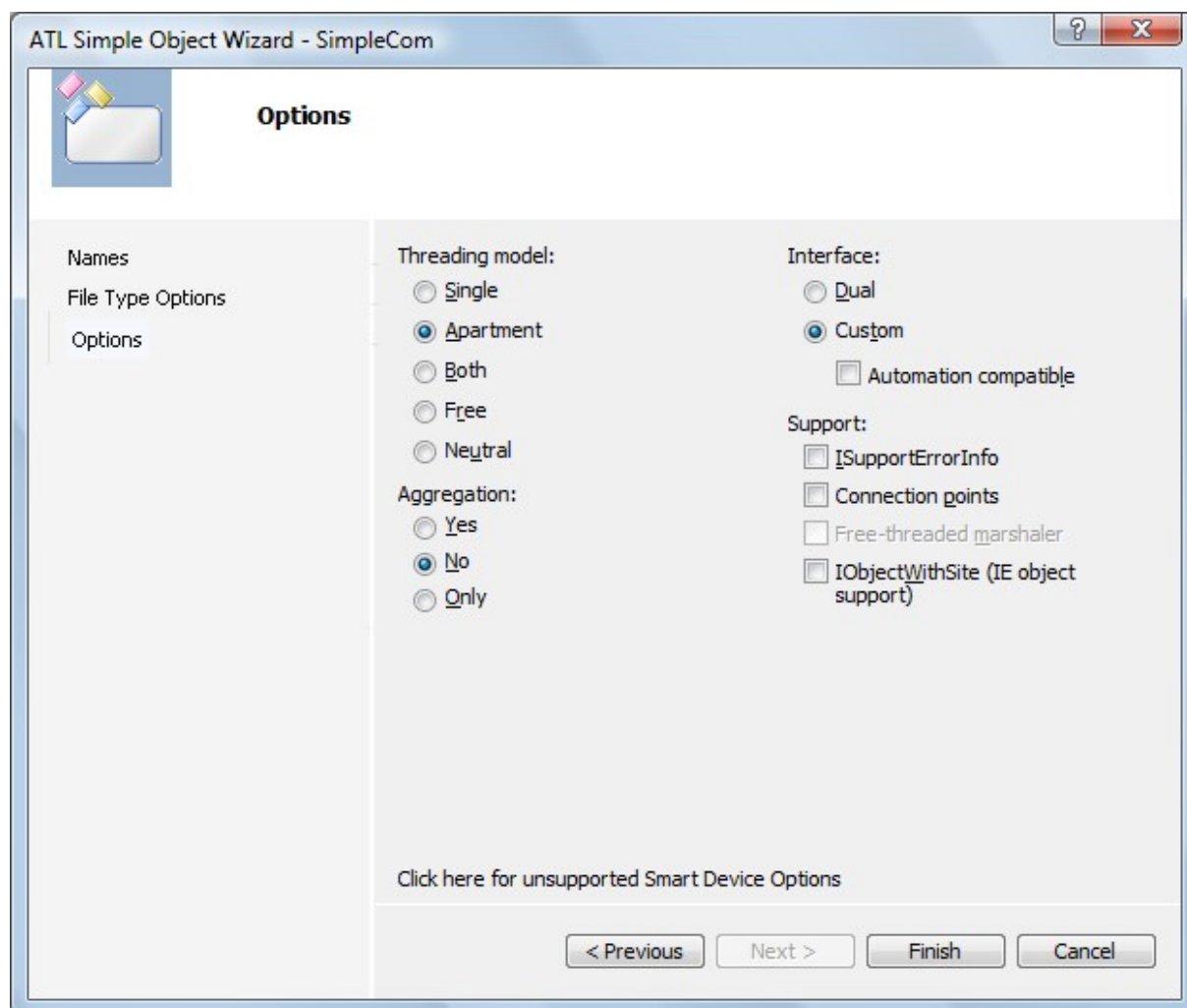
5. Klikamy Next.

The screenshot shows the 'File Type Handler Options' page of the 'ATL Simple Object Wizard - SimpleCom' dialog. The left sidebar has three items: 'Names', 'File Type Options' (which is selected), and 'Options'. The main area contains the following options:

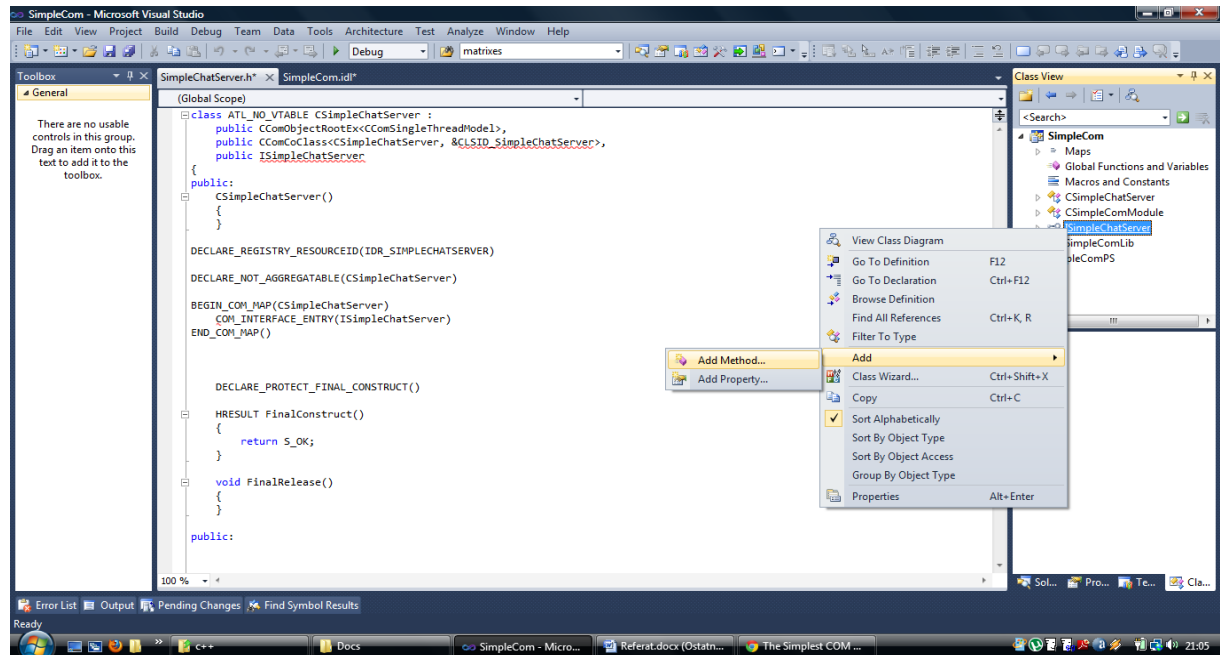
- ☐ Add file type (preview/thumbnail/search handler) support
 - ☒ Preview Handler
 - ☐ Thumbnail Handler
 - ☐ Search Handler
- File Extension:
- ATL-based Document**
 - Generic document class name: .h of generic document: ...
 - ☐ Generate ATL-based document
 - .cpp of generic document:
- MFC-based Document (and View)**
 - Document class name: .h of MFC document: ...
 - View class name: .h of view: ...

At the bottom, there are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

6. Wybieramy następujące opcje. Klikamy Finish.



7. W Class View znajdujemy interface oznaczony kluczykiem, klikamy PPM i wybieramy dodaj metodę.



8. Ustawiamy następujące argumenty i klikamy Finish.

Add Method Wizard - SimpleCom

Welcome to the Add Method Wizard

Names
IDL Attributes

Return type: HRESULT

Method name: CallMe

Parameter attributes:
☒ in ☐ out ☐ retval

Parameter type:

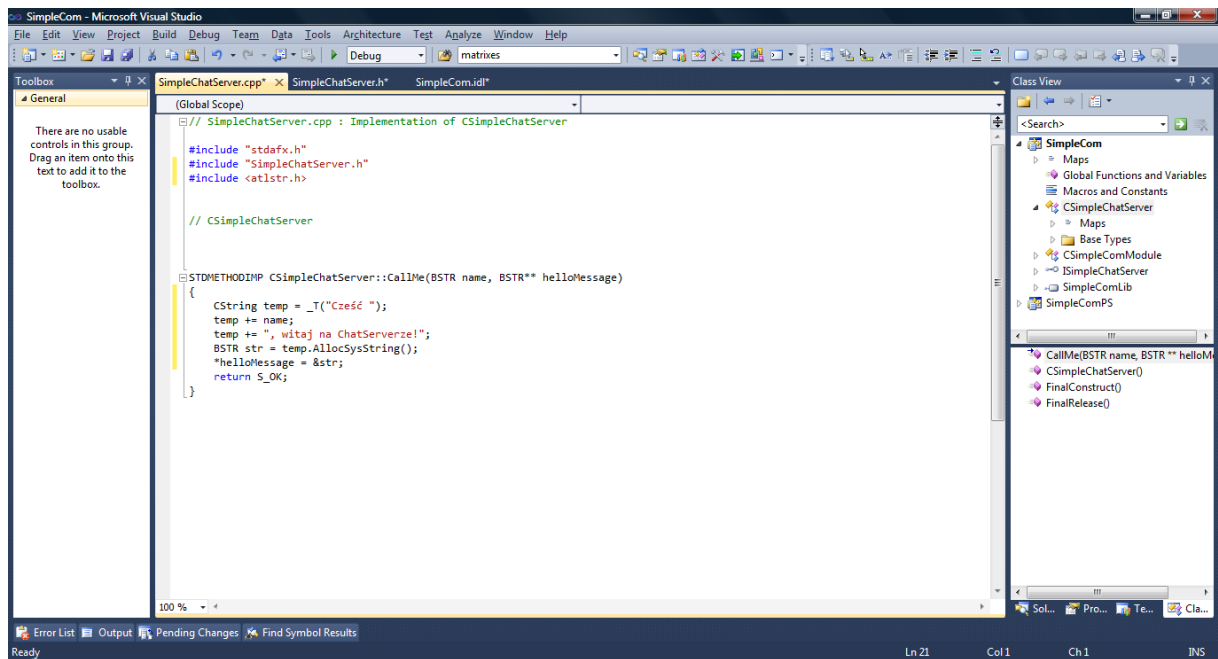
Parameter name:

[in] BSTR name
[in] BSTR** helloMessage

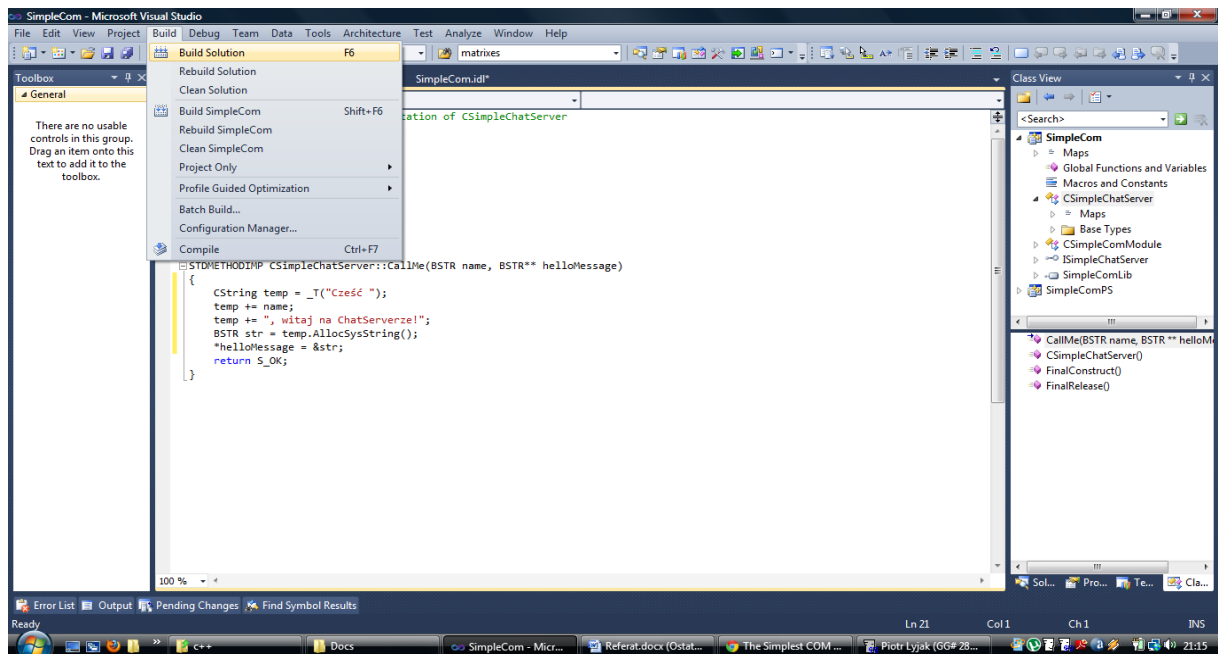
Add Remove

< Previous Next > Finish Cancel

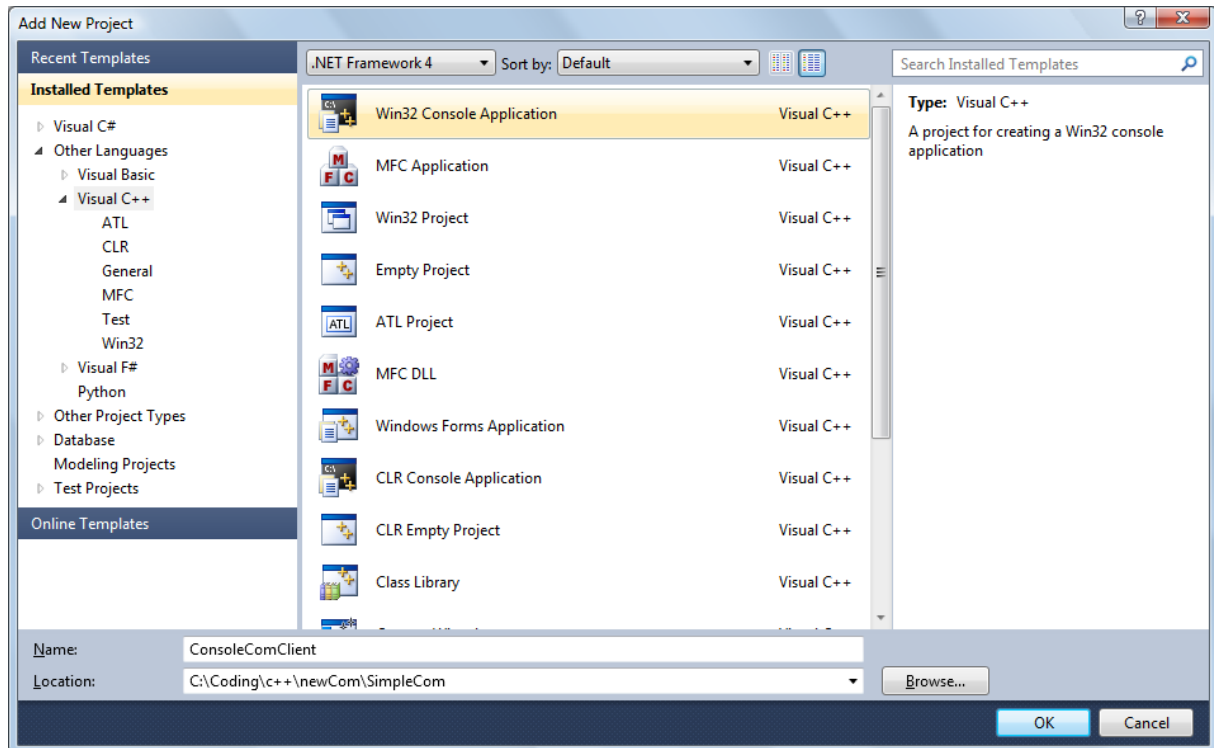
9. Implementujemy metodę interface'u w pliku SimpleChatServer.cpp. Można go łatwo znaleźć poprzez Class View, pod hasłem CSimpleClassServer. (Pamiętamy o zaincludowaniu atlstr.h)



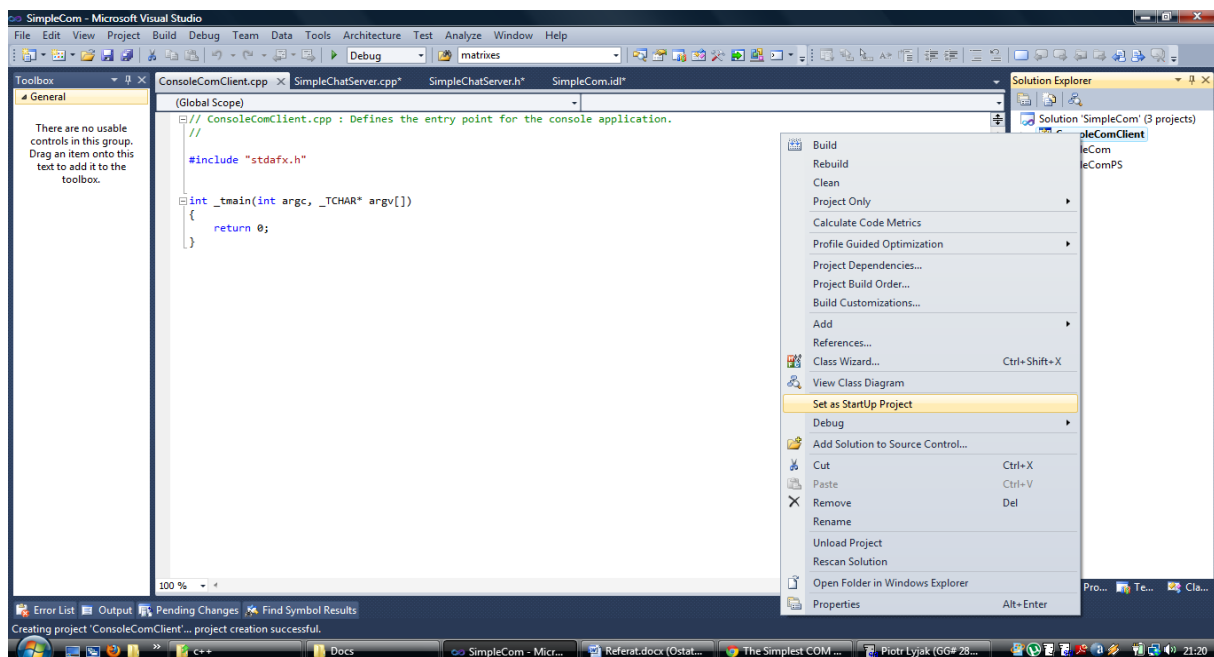
10. Serwer skończony. Teraz wystarczy go zbudować. (Potrzebne prawa administratora)



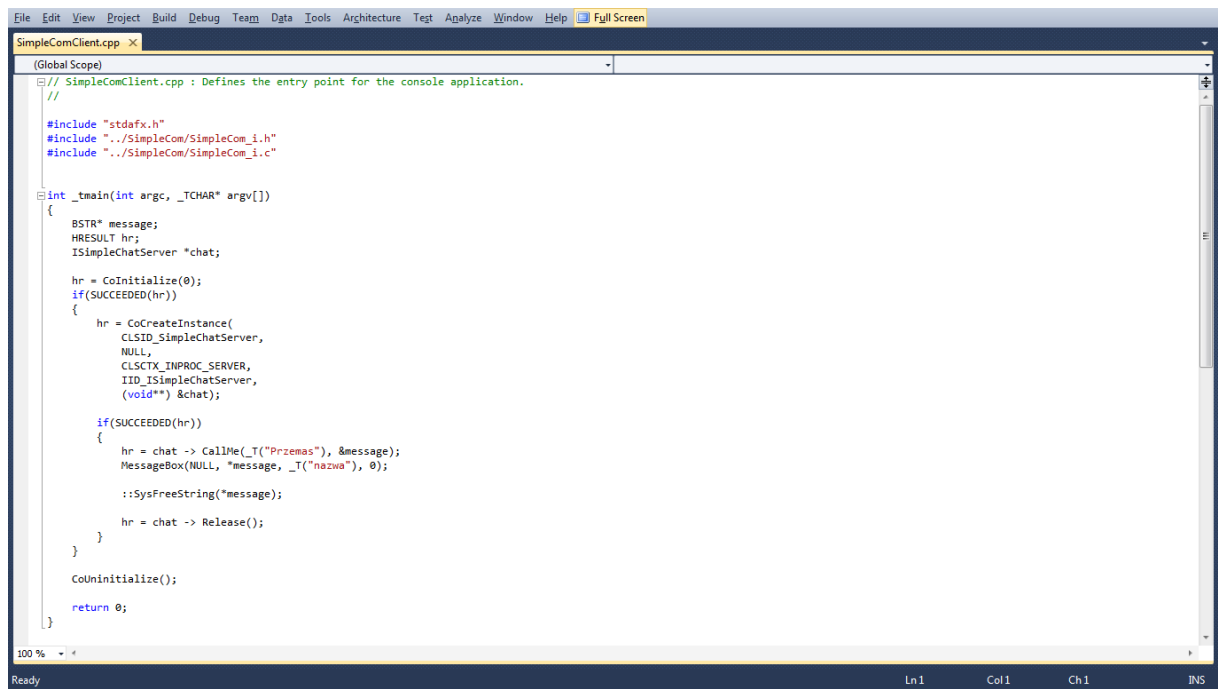
11. Dodajemy nowy projekt na kod klienta. W tym celu klikamy na solution w Solution Explorerze i dodajemy nowy projekt. (aplikacja konsolowa)



13. Sprawiamy by nowododany projekt był projektem startowym. (nazwa zostanie pogrubiona)



14. Implementujemy klienta i buildujemy



```
File Edit View Project Build Debug Team Data Tools Architecture Test Analyze Window Help Full Screen
SimpleComClient.cpp
(Global Scope)
// SimpleComClient.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include "../SimpleCom/SimpleCom_i.h"
#include "../SimpleCom/SimpleCom_i.c"

int _tmain(int argc, _TCHAR* argv[])
{
    BSTR* message;
    HRESULT hr;
    ISimpleChatServer* chat;

    hr = CoInitialize(0);
    if(SUCCEEDED(hr))
    {
        hr = CoCreateInstance(
            CLSID_SimpleChatServer,
            NULL,
            CLSCTX_INPROC_SERVER,
            IID_ISimpleChatServer,
            (void**) &chat);

        if(SUCCEEDED(hr))
        {
            hr = chat -> CallMe(_T("Przemas"), &message);
            MessageBox(NULL, *message, _T("nazwa"), 0);

            ::SysFreeString(*message);

            hr = chat -> Release();
        }
    }

    CoUninitialize();

    return 0;
}
```

100 %
Ready Ln1 Col1 Ch1 INS