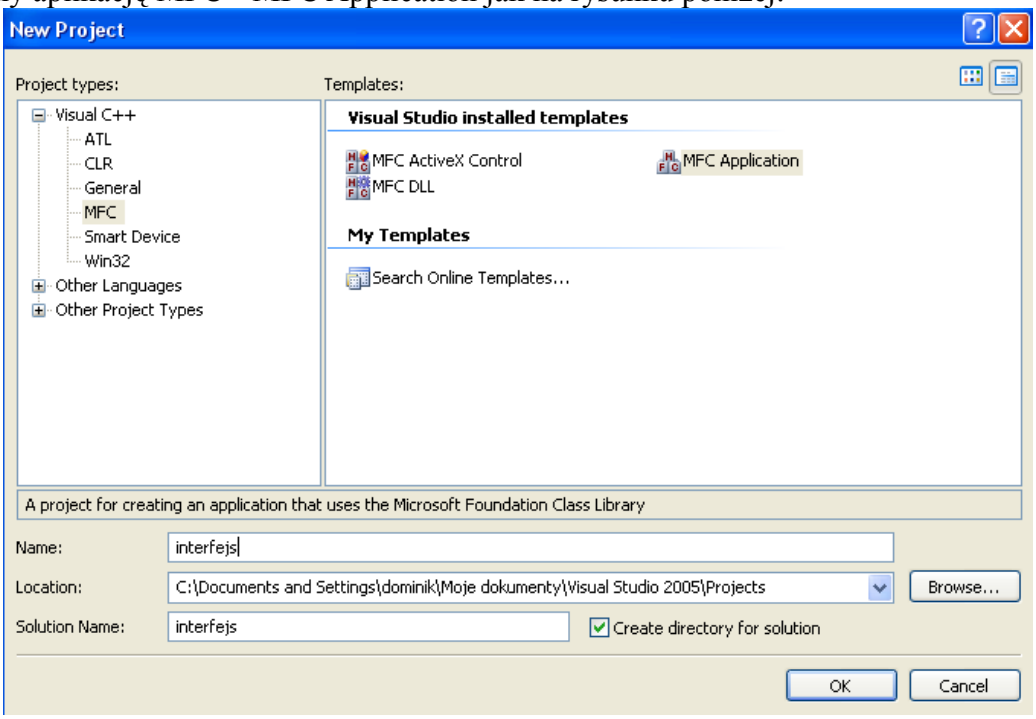
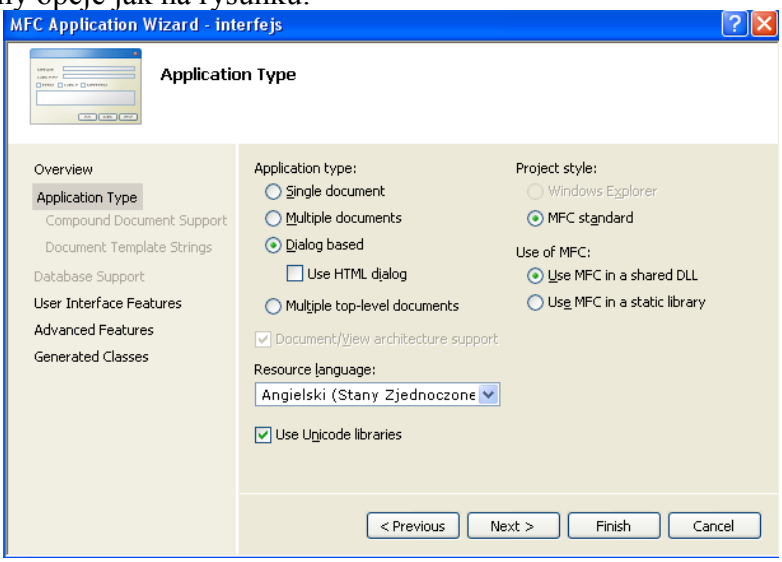
Interfejs w MFC

1. Tworzenie nowego projektu

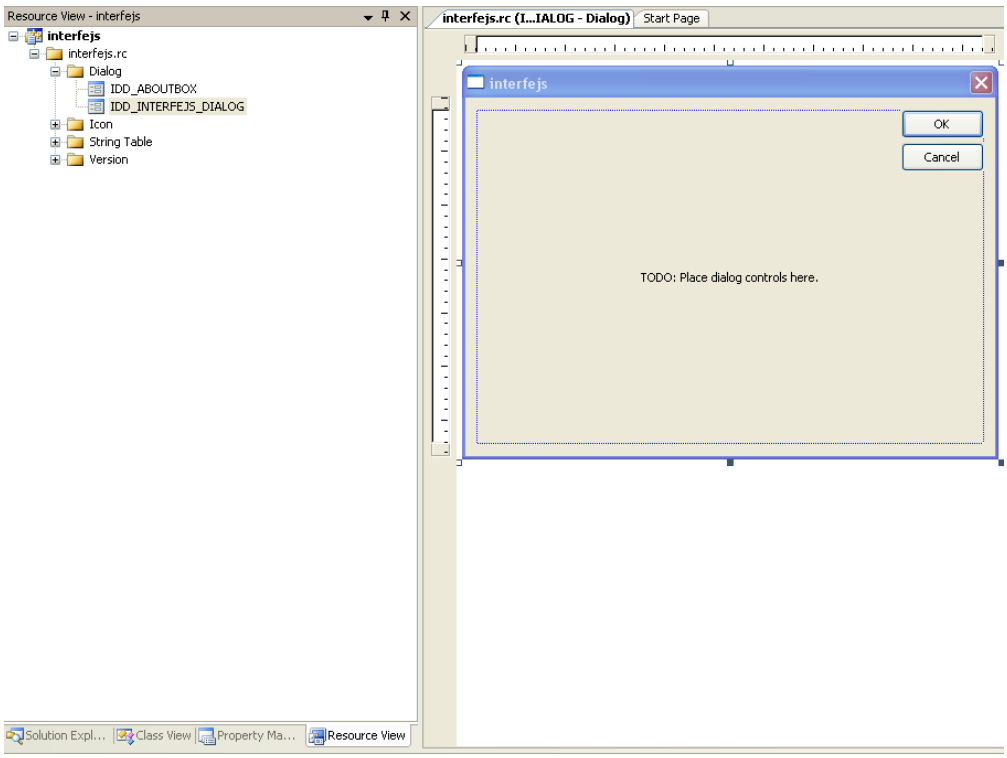
Wybieramy File->New->Project
Wybieramy aplikację MFC->MFC Application jak na rysunku poniżej:



Następnie wybieramy opcje jak na rysunku:



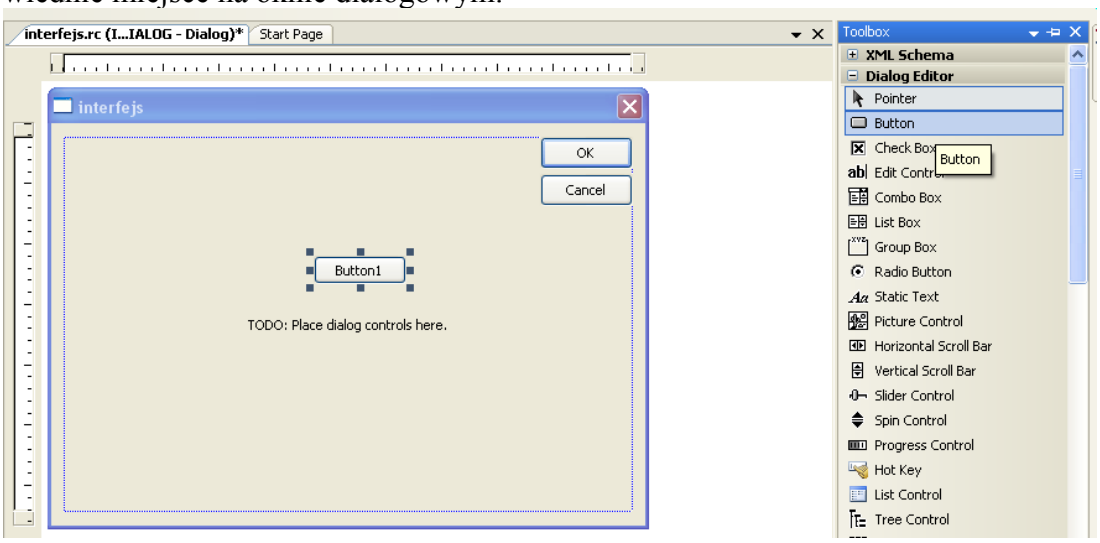
Następnie wybieramy przycisk „Finish”
Zostaje utworzony projekt z zakładką IDD_nazwaprojektu_DIALOG. Aby mieć możliwość zmiany wyglądu należy z zakładki „Resource View” wybrać odpowiednie okno dialogowe „IDD_nazwaprojektu_DIALOG”. Uzyskany efekt będzie następujący:



W zakładce „Resource View” mamy do wyboru elementy zasobów projektu. Po prawej znajduje sie okno dialogowe, które będziemy edytować.
Dodajmy do dialogu przycisk.

2. Przyciski

W celu dodania przycisku należy wybrać odpowiedni element z zakładki „Toolbox” i przenieść go w odpowiednie miejsce na oknie dialogowym:



Aby obsłużyć zdarzenie związane z wciśnięciem przycisku kliknijmy dwukrotnie na przycisk. Zostaniemy przeniesieni do metody:

```
void CinterfejsDlg::OnBnClickedButton1()
{
	//tutaj należy umieścić odpowiednie funkcje i metody
}
```

3. Okna dialogowe

Aby wyświetlić okno z komunikatem należy użyć polecenia:

```
void CinterfejsDlg::OnBnClickedButton1()
```

```
{
        // TODO: Add your control notification handler code here
        MessageBox(_T("Komunikat"),_T("Nazwa okna"),MB_OK|MB_ICONINFORMATION);
}
```

Zamiast ikony “MB_ICONINFORMATION” można użyć: “MB_ICONSTOP”
Aby yswietlić 2 przyciski (OK i Cancel) należy uzyć parametru : “MB_OKCANCEL”. Ine możliwości to:
MB_YESNO for Yes and No.
MB_YESNOCANCEL for Yes, No, and Cancel.
MB_ABORTRETRYIGNORE for Abort, Retry, and Ignore.
MB_RETRYCANCEL for Retry and Cancel.

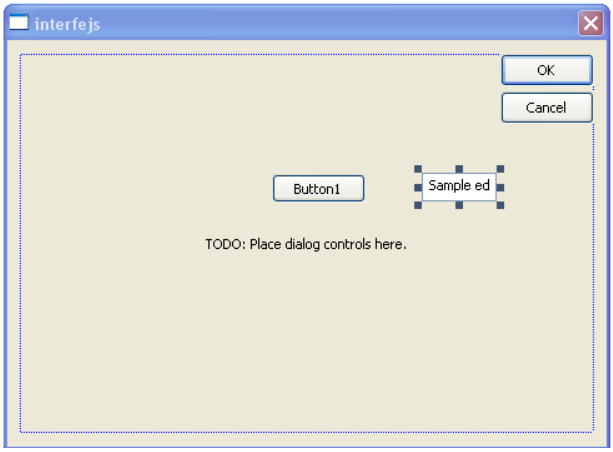
Pozostałe parametry jakie można przekazać do tej funkcji można znaleźć w dokumentacji i kursach w internecie (www.google.com).

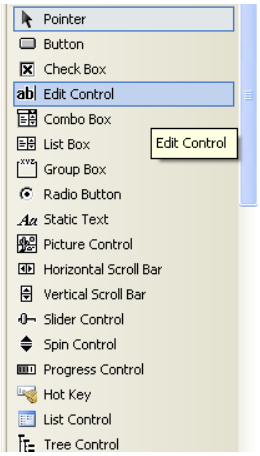
Aby odczytać wybrany przez użytkownika przycisk wystarczy pobrać wartość zwracaną przez funkcję MessageBox, jak na przykładzie poniżej:

```
int wybrano = MessageBox(_T("Komunikat"),_T("Nazwa okna"),MB_OKCANCEL|MB_ICONSTOP);
if (wybrano==1) {
        //przycisk ok
}
else if (wybrano==2){
        // przycisk “cancel”
}
itd...
```

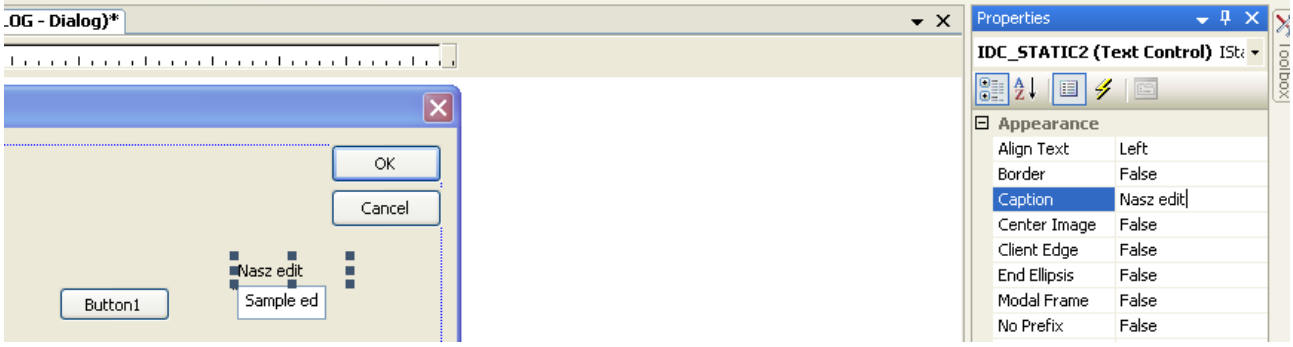
4. Pola edycyjne.

Dodajmy pole edycji do naszej formatki:

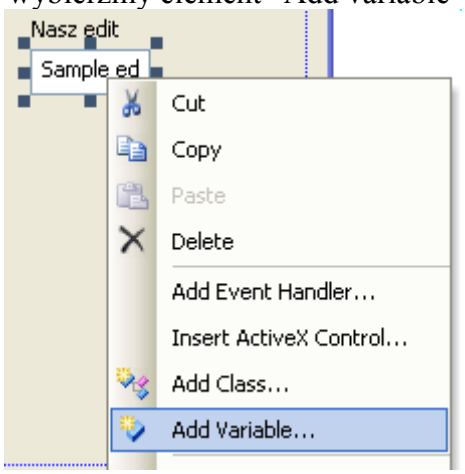




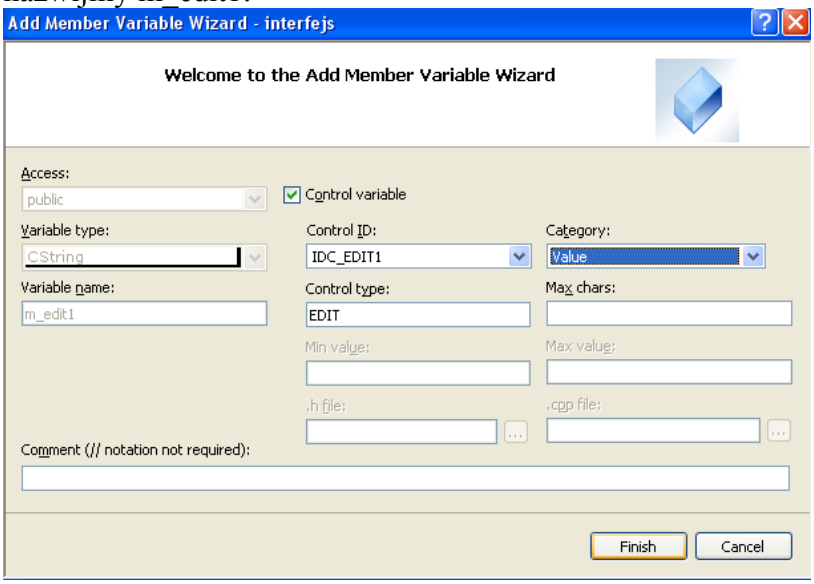
Możemy równocześnie opisać nasze pole przy uzyciu elementu static_text. Właściwości wszystkich elementów znajdujacych sie na ekranie zmieniamy przy użyciu zakładki “Properties”. Zmieńmy zatem nazwę naszego elementu static_text. Do tego celu służy pole “caption”:



Teraz musimy powiązać zmienną z naszym polem edycyjnym. W tym celu kliknijmy na pole edycyjne prawym przyciskiem i wybierzmy element “Add variable”:



W tym miejscu należy wybrać typ zmiennej. W przykładzie zmienna będzie typu CString, kategoria “Value”. Zmienną nazwijmy m_edit1:



Konstruktorze klasy związanej z Dialogiem możemy ustawić wartość początkową zmiennej m_edit1. Ustawmy wartość w polu edycyjnym na “0”:

```
CinterfejsDlg::CinterfejsDlg(CWnd* pParent /*=NULL*/)
	: CDialog(CinterfejsDlg::IDD, pParent)
	, m_edit1(_T("0"))
{
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}
```

Aby zmienić wartość, która jest wpisana w polu edycyjnym należy użyć polecenia (można je umieścić w zdarzeniu od przycisku):

```
	m_edit1="43";
	UpdateData(false);
```

Aby odczytać wartość należy użyć polecenia:

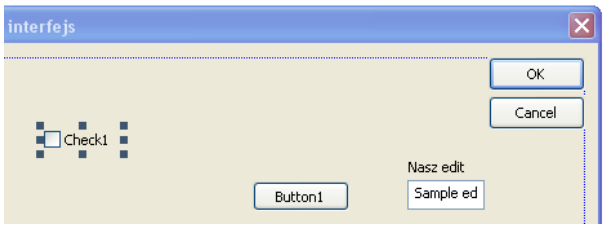
```
	UpdateData(true);
	double wartosc_double = wcstod(m_edit1,NULL);
```

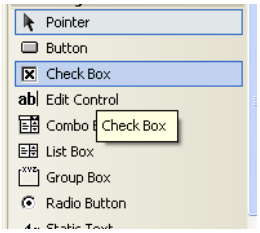
W przykładzie wartość Cstring jest przekształcana na tym double. Na temat konwersji typów można

przeczytać tutaj:
http://www.codeproject.com/KB/cpp/data_conversions.aspx

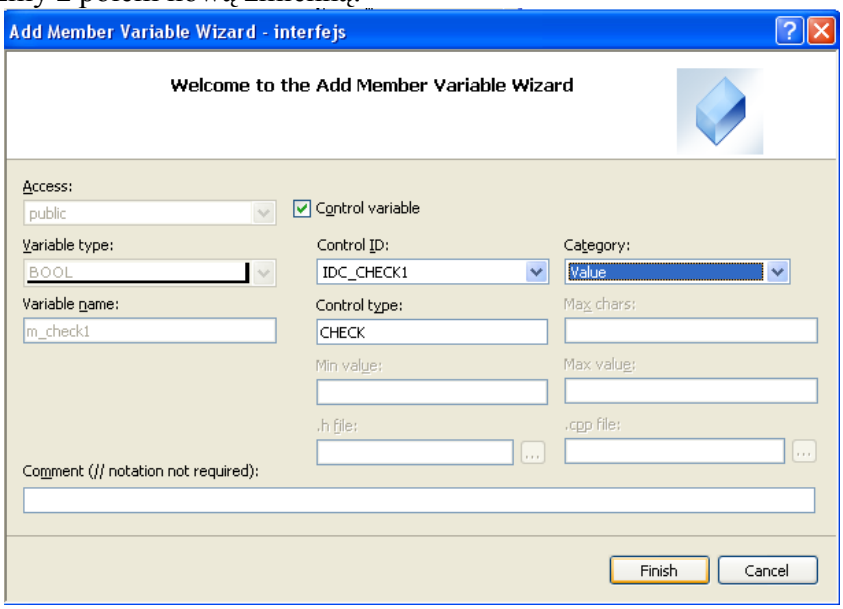
5. CheckBox

Dodajmy pole “checkBox” do Dialogu:





Następnie powiążmy z polem nową zmienną:



Analogicznie jak w przypadku pola “Edit“ aby zmienić wartość początkową należy skorzystać z konstruktora.

Aby od strony kodu zmienić zaznaczenie takiego pola użyjemy poleceń:

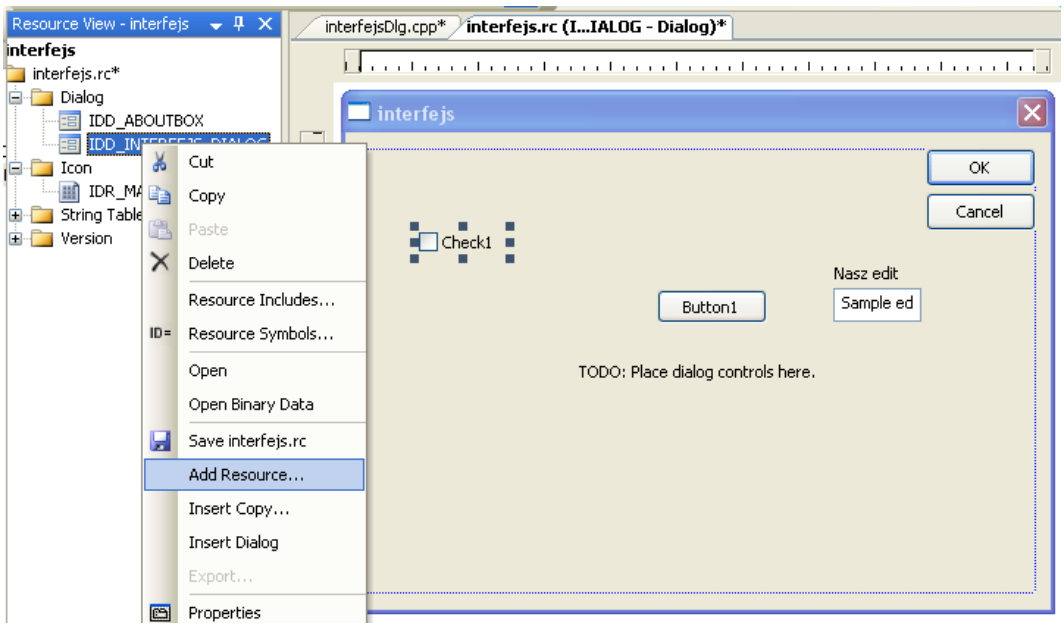
```
m_check1=TRUE;
UpdateData(false);
```

Aby odczytać wartość zaznaczenia:

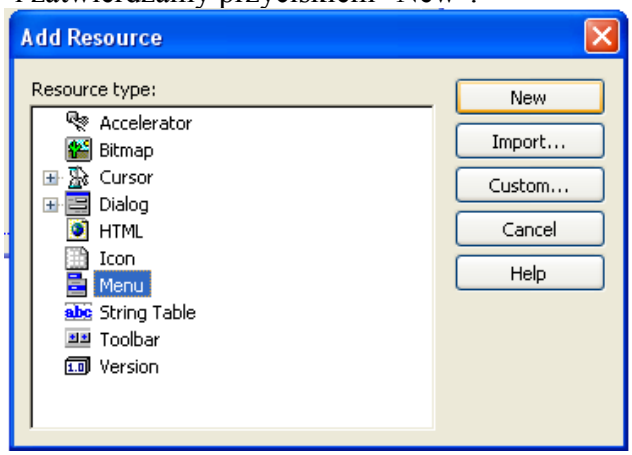
```
UpdateData(true);
if (m_check1==FALSE) {
        //jeżeli checkbox nie jest zaznaczony
}
```

6. Menu

W zakładce “Resources” klikamy na Dialog, do którego chcemy dodać menu:

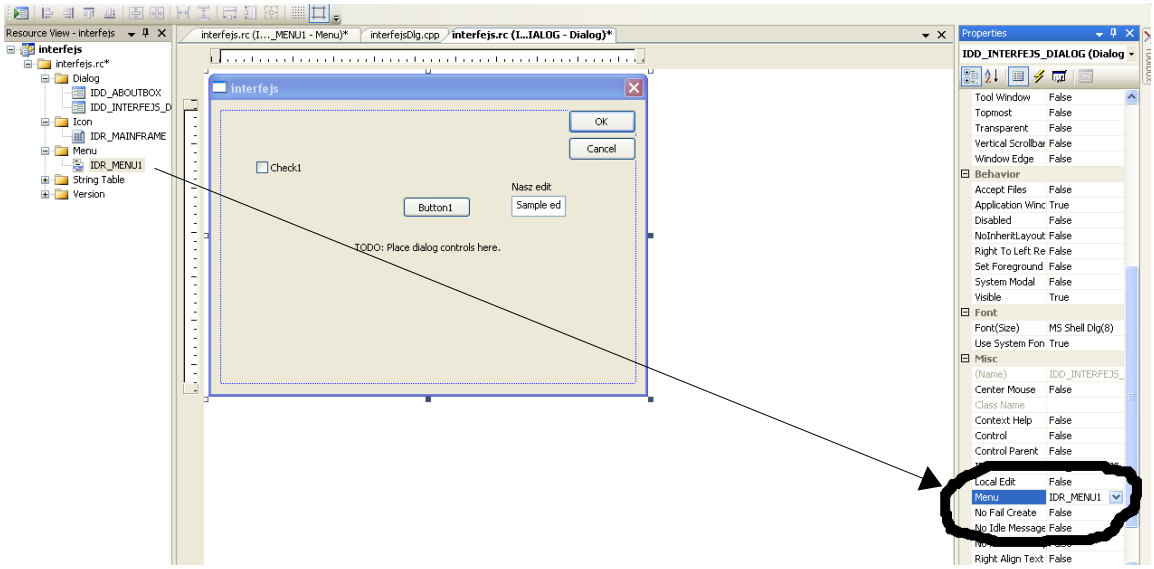


Wybieramy opcję "menu" i zatwierdzamy przyciskiem "New":

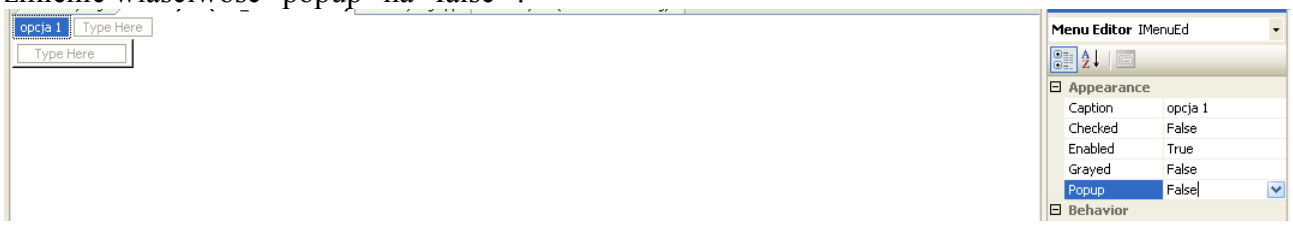


Pojawi się puste menu. Dodajmy zatem element "opcja1":

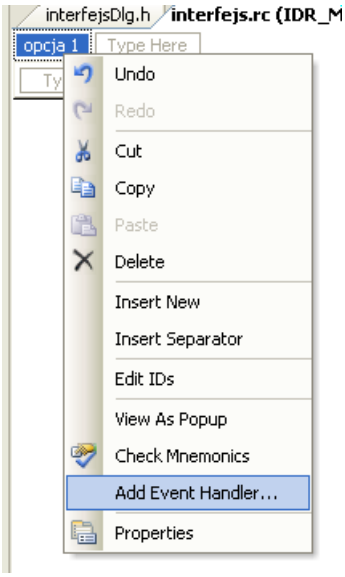
Dodajmy teraz menu do naszego okna dialogowego. W tym celu należy wybrać zakładkę "Properties" i następnie w polu "Menu" wpisać nazwę używanego menu:



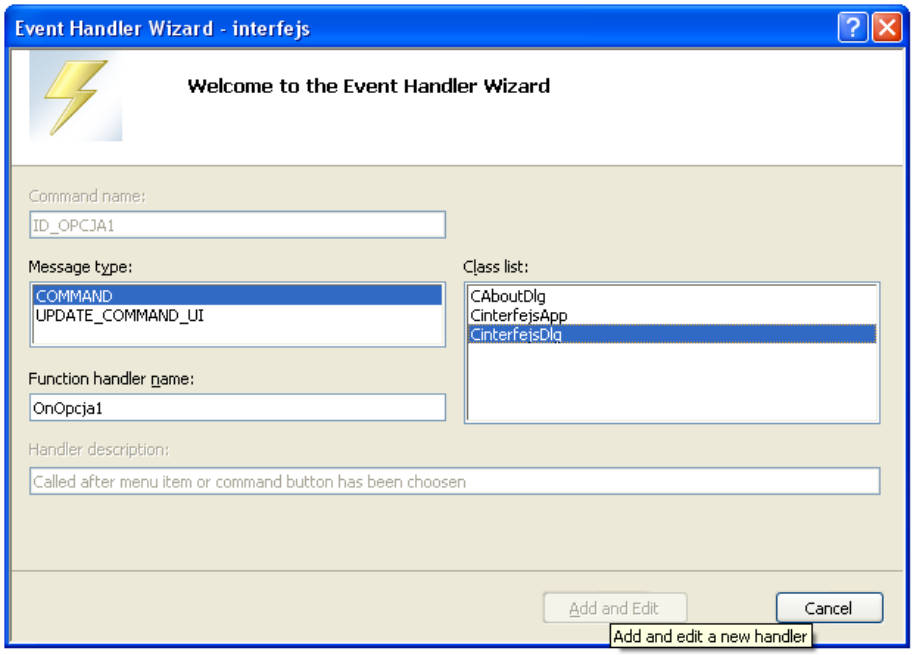
Aby powiązać elementy menu z odpowiednimi zdarzeniami należy (będąc na zakładce Menu) zmienić właściwość "popup" na "false" :



Teraz należy kliknąć prawym przyciskiem na odpowiednią opcję w menu i wybrać pole "Add Event Handler":



Następnie należy wybrać klasę, w której powinno zostać dodane zdarzenie (dla przykładu klasa zwiazana z podstawowym dialogiem CinterfejsDlg):

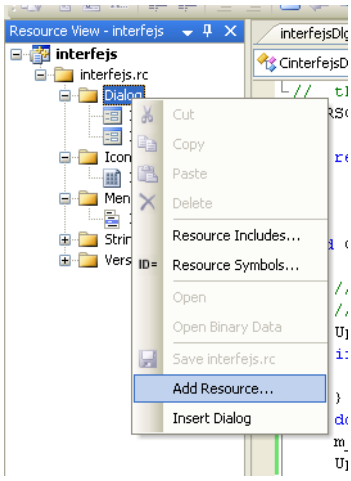


Klikamy na przycisk "Add and Edit" i zostajemy automatycznie przeniesieni do metody wywoływanej po wybraniu danej opcji z menu:

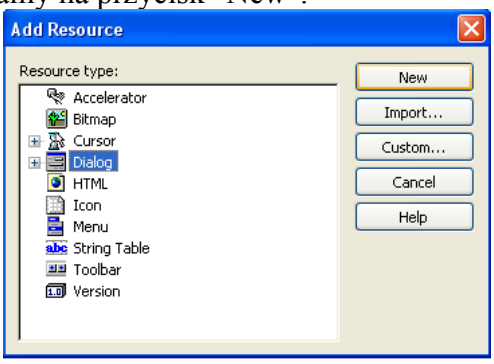
```
void CinterfejsDlg::OnOpcja1()
{
	// TODO: Add your command handler code here
}
```

7. Okna dialogowe

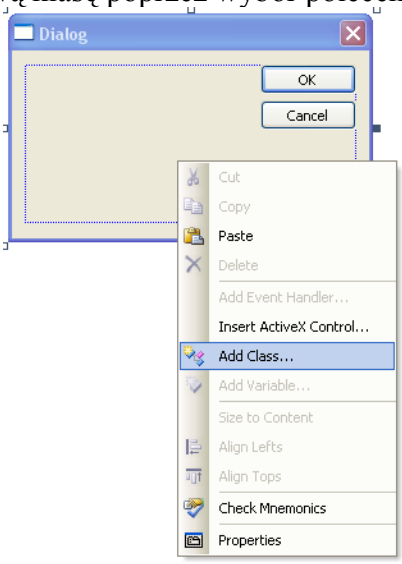
Dodajmy nowy dialog klikając prawym przyciskiem na pole "Dialogs" w menu "Resources" i wybierając "Add Resource":



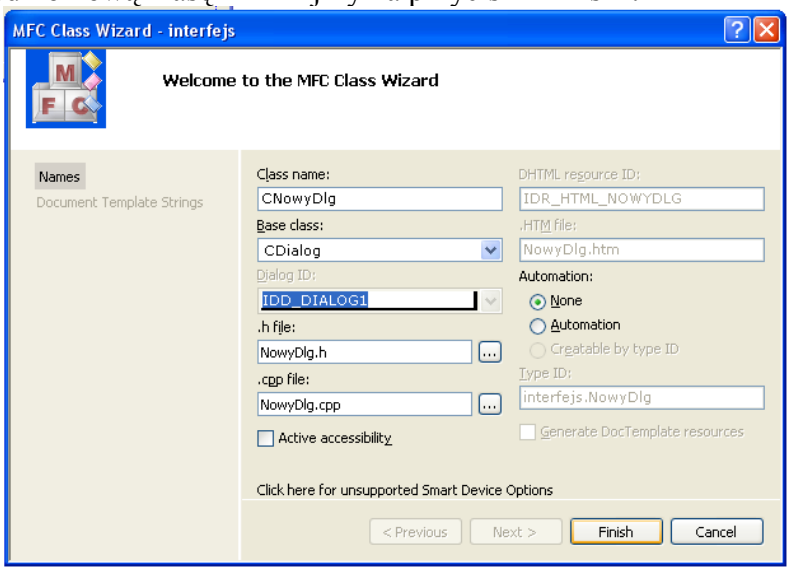
Wybieramy nowy dialog i klikamy na przycisk "New":



Do projektu zostanie dodane nowe okno dialogowe. Kliknijmy na nim prawym przyciskiem myszki, a następnie stwórzmy nową klasę poprzez wybór polecenie "Add Class":



Nazwijmy odpowiednio nową klasę i kliknijmy na przycisk "Finish":



Zostanie utworzony szablon nowej klasy. CnowyDlg. Dodajmy tę klasę do projektu. W tym celu dołączymy do klasy związanej z pierwszym dialogiem (w tym przypadku interfejsDLg.cpp) odpowiedni nagłówek:

```
#include "NowyDlg.h"
```

Dodajmy w odpowiednie miejsce(plik nagłówkowy klasy) również obiekt będący naszym nowym oknem:

```
CNowyDlg dlgNowy;
```

W metodzie OnInitDialog dodajmy odpowiednie funkcje inicjalizujące obiekt:

```
// TODO: Add extra initialization here
        dlgNowy.Create(IDD_DIALOG1,this); //tworzymy dialog
        dlgNowy.ShowWindow(SW_HIDE); //ukrywamy okno dialogowe
```

Teraz możemy otworzyć nasze nowe okno po kliknięciu na opcję w menu:

```
void CinterfejsDlg::OnOpcja1()
{
	// TODO: Add your command handler code here
	dlgNowy.ShowWindow(SW_SHOW);
}
```

Pozostałe parametry przekazywane do funkcji ShowWindow(...); można znaleźć w internecie. Podobnie przebiega obsługa pozostałych elementów interfejsu. W zasobach internetu znajduje się ogromna liczba kursów i tutoriali do MFC. Pomocna może być również dokumentacja MFC, a przede wszystkim: www.google.com lub alternatywna wyszukiwarka.