

## Laborator VII

### ***APLICAȚII WINDOWS – Casete de dialog***

#### **1. Scopul lucrării**

Lucrarea de laborator are ca scop familiarizarea cu modul de creare a unei aplicații Windows utilizând biblioteca de clase MFC în mediul Microsoft Visual C++ 2010.

#### **2. Resurse**

Tipurile de resurse utilizate de aplicațiile Windows sunt:

- *Bitmaps* – ce stochează imagini
- *Cursoare (Cursors)* - ce indică poziția curentă a mouse-ului.
- *Ferestre de dialog (Dialog boxes)* sunt ferestre utilizate pentru a interacționa cu utilizatorul.
- *Icoane (Icons)* – sunt mici imagini bitmap, ce pot fi utilizate pentru a reprezenta diverse obiecte utilizate în aplicațiile Windows.
- *Meniuri (Menus)* - sunt grupuri de comenzi prin intermediul cărora se realizează funcțiile unui program.
- *Bara de instrumente (Toolbar)* – este un tip special de meniu, care afișează opțiuni de comandă folosind pictograme (imagini grafice mici).
- *Acceleratoare (Accelerators)* - sunt taste sau combinații de taste care atunci când sunt apăsată au același efect ca și selectarea unui articol din meniu.

#### **3. Mesaje și funcții**

În situațiile în care este necesar ca sistemul de operare Windows să comunice cu o aplicație, îl trimite acesteia un mesaj. Trimiterea unui mesaj este similară apelului unei funcții (de fapt, biblioteca MFC va rula majoritatea mesajelor ca apeluri de funcții în aplicații). De exemplu, în aplicația AppWizard biblioteca MFC apelează funcția OnDraw ori de câte ori Windows-ul trimite un mesaj de tip WM\_PAINT. Atunci când este necesar ca o aplicație să comunice cu o fereastră îl trimite acesteia un mesaj. De exemplu, pentru a activa un control i se trimită acestuia mesajul WM\_ENABLE.

Biblioteca Microsoft Foundation Class (MFC) reprezintă o colecție de clase C++ care ajută programatorul în crearea de aplicații care rulează sub sistemul de operare Microsoft Windows

#### **4. Crearea proiectelor bazate pe casete de dialog**

O casetă dialog (dialog box) este o fereastră specializată folosită pentru a interacționa cu utilizatorul, preluând date sau diverse tipuri de opțiuni selectate de acesta. Casetele se mai pot utiliza și pentru comunicații unidirectionale, pentru a afișa doar diverse informații de interes (de tip copyright, start-up, diverse alte notificări)

Casetele dialog sunt de diferite tipuri și forme, de la simple casete de mesaje, capabile să afișeze o singură linie de text, până la casete ce conțin controale sofisticate.

Astfel există :

- Casete de mesaje (Message boxes)
- Casete dialog modale (Modal dialog boxes)

- Casete dialog nemodale (Modeless dialog boxes)

O casetă dialog modală impiedică utilizatorul să efectueze orice altă operație cu programul în cauză până când dialogul prin intermediul casetei respective nu s-a încheiat (și implicit fereastra s-a închis).

O casetă dialog nemodală permite utilizatorului să efectueze orice altă operație cu programul în cauză pe durata afișării casetei respective ( și deci dialogul prin intermediul casetei respective nu s-a încheiat, fereastra fiind în continuare deschisă).

O aplicație de tip casetă de dialog (*dialog box-based project*) folosește o fereastră de tip casetă de dialog drept fereastră principală a programului și ea este folosită frecvent pentru programe de dimensiuni mici cu care se interacționează printr-o singură casetă de dialog.

O aplicație de tip fereastră de dialog are un singur meniu accesibil prin intermediul meniului sistem în colțul din stânga sus al ferestrei.

#### **4.1. Crearea proiectului cu generatorul de aplicații AppWizard**

Deoarece proiectele bazate pe ferestre de dialog sunt relativ simple, sunt necesari puțini pași pentru a fi create:

1. Se creează un nou proiect MFC Application cu numele HelloDialog. Se va deschide o fereastră ca în Figura 1. Se apasă pe butonul Next.

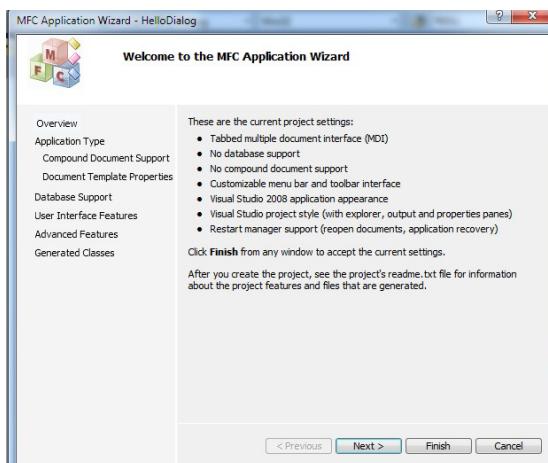


Figura 1.

2. În fereastra din Figura 2, se selectează tipul proiectului ca fiind bazat de fereastră de dialog (dialog box-based project ).

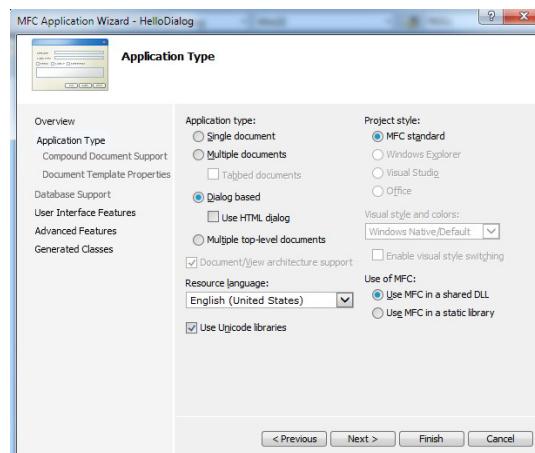


Figura 2.

3. Din partea stângă a ferestrei se selectează User Interface Features. În câmpul Dialog Title se șterge numele implicit și se scrie, de exemplu: **Prima mea fereastră de dialog**, ca în Figura 3.

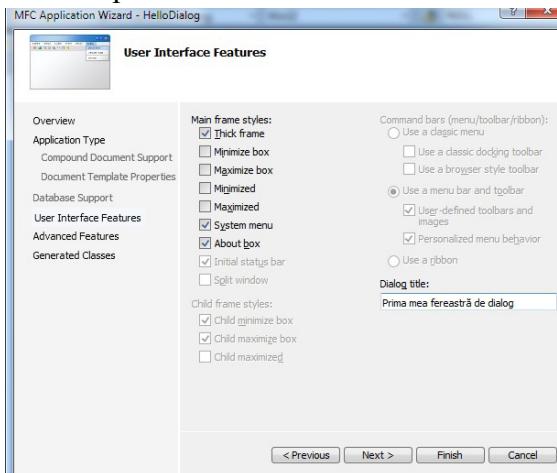


Figura 3.

4. Se acționează butonul Finish.

După ce s-a creat proiectul, explorați structura acestuia (Figura 4). (De observat fișierele și resursele create de mediu pentru proiect.

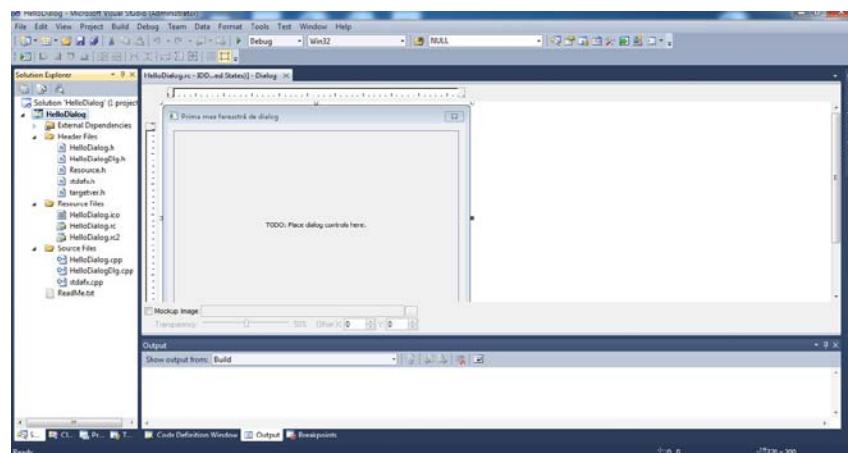


Figura 4.

5. Se dă comanda Build și apoi Debug->Start Without Debugging. Se va crea fereastra dialog din Figura 5.

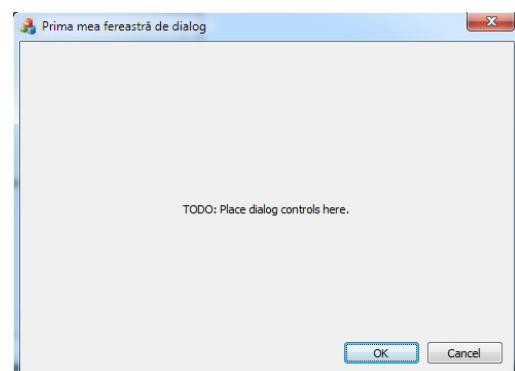


Figura 5.

#### **4.2. Utilizarea editorului de casete de dialog**

Din fereastra Solution Explorer se face dublu clic pe HelloDialog.rc, din directorul Resource Files. Deschiderea editorului se face cu dublu click pe imaginea casetei `IDD_HELLODIALOG_DIALOG`. Se afișează caseta de dialog împreună cu o bară de instrumente/uelte flotantă (*dockable toolbar*). Caseta de dialog ca conține implicit un control de tip text static (static text control). Modificați acest control astfel încât el să conțină textul "Hello Dialog Project", așa cum se arată în Figura 6.

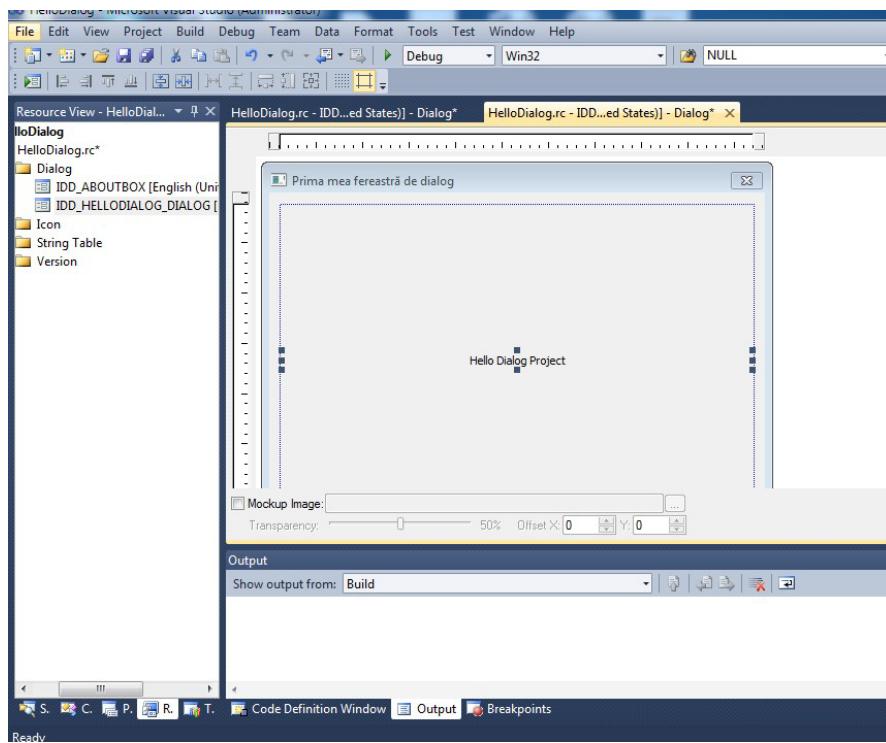


Figura 6.

Compilați și executați proiectul, analizând componentele și exploatarea acestuia în diverse situații. Fereastra obținută este reprezentată în Figura 7.

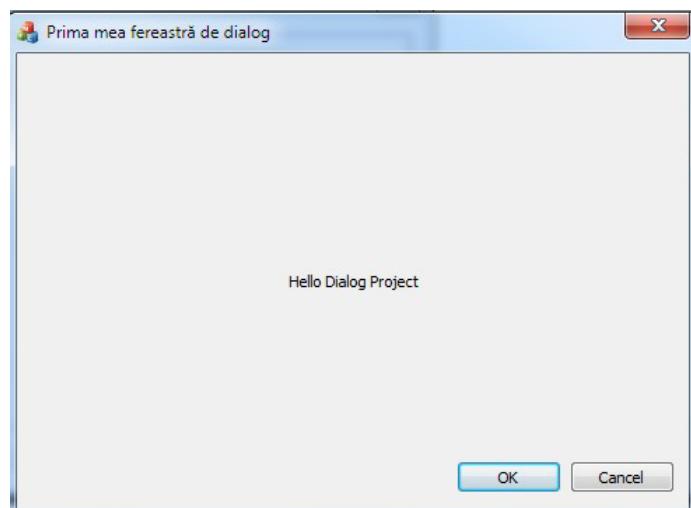


Figura 7. Fereastra de dialog pentru proiectul *HelloDialog*.

### **4.3. Casete de mesaje**

Cel mai simplu tip de caseta de dialog este caseta de mesaje (message box), folosită pentru a afișa diverse informații.

O astfel de fereastră se poate crea folosind biblioteca de clase MFC (păstrând parametrii implicați furnizați de aceasta) prin următorul apel :

```
AfxMessageBox(_T("Hello World"));
```

Se crează astfel o casetă de mesaje ce conține un semn de exclamare afișat în interiorul unui triunghi galben.

Există posibilitatea de a afișa și alte simboluri, așa cum rezultă din figura 2.



Figura 2. Simboluri ce se pot afișa într-o fereastră de mesaje

Pentru a adăuga o casetă de mesaje este suficient să se apeleze metoda `AfxMessageBox`, căreia trebuie să îi se furnizeze cel puțin un parametru și anume textul care se va afișa. Opțional, se pot specifica tipul de imagine afișată și respectiv un aranjament de butoane.

Semnificația și numele stilului imaginii afișate sunt prezentate în Tabelul 1.

Tabelul 1. Imagini folosite în casetele de mesaje din Windows

Imagine afișată	Semnificație	Numele stilului
Semnul exclamării(Exclamation mark)	Atenție (Warning)	MB_ICONEXCLAMATION
Litera "i" într-un cerc(An "i" in a circle)	Informație(Information)	MB_ICONINFORMATION
Semnul întrebării (Question mark)	Intrebare(Question)	MB_ICONQUESTION
Semnul stop (Stop sign)	Eroare(Error)	MB_ICONSTOP

Suplimentar se poate specifica o anumită combinație de butoane. Implicit se include un singur buton cu numele OK. Este util însă uneori să se preia de la utilizator răspunsul la o întrebare simplă, de exemplu confirmarea execuției unei anumite acțiuni.

De exemplu codul de mai jos afișează o casetă de mesaje ce conține semnul întrebării, interogând utilizatorul dacă fișierul curent trebuie șters:

```
int nChoice = AfxMessageBox(_T("Overwrite existing file?"),
                           MB_YESNOCANCEL | MB_ICONQUESTION );
if( nChoice == IDYES )
{
    // Overwrite file
```

{}

Utilizatorul poate alege între butoanele inscriپionate Yes, No sau Cancel.

In Tabelul 2 se prezintă diferite aranjamente posibile pentru butoane folosite în casetele de mesaje.

Tabelul 2. Aranjamente de butoane

Stil	Butoane incluse în caseta de dialog
MB_ABORTRETRYIGNORE	Abort, Retry, and Ignore
MB_OK	OK
MB_OKCANCEL	OK and Cancel
MB_RETRYCANCEL	Retry and Cancel
MB_YESNO	Yes and No
MB_YESNOCANCEL	Yes, No, and Cancel

Valoarea returnată de apel reprezintă selecția făcută de utilizator. In Tabelul 3 se prezintă valorile posibile a fi întoarse de AfxMessageBox.

Tabelul 3. Valorile returnate de apelul AfxMessageBox

Valoare	Buton
IDABORT	Abort
IDCANCEL	Cancel
IDIGNORE	Ignore
IDNO	No
IDOK	OK
IDRETRY	Retry
IDYES	Yes

Observații:

Pentru a combina diverse opțiuni folosite în apelul funcției AfxMessageBox se folosește operatorul | (SAU pe biți - *bitwise OR operator* ).

Se pot combina doar un stil de imagine și un stil de butoane. Implicit se afișează semnul exclamării și butonul OK.

## **5. Folosirea casetelor de dialog pentru introducerea de informații**

Un alt mod de utilizare a casetelor de dialog este acela de a colecta date de intrare, cu avantajul de a putea prevedea o serie de controale care să faciliteze preluarea datelor.

O parte din ferestrele de dialog frecvent folosite în Windows, numite *common dialog boxes*, sunt incluse în sistemul de operare. Ele sunt disponibile printr-un simplu apel de funcție și nu necesită crearea unei resurse *ad-hoc*.

Ferestrele de dialog nu sunt altceva decât ferestre specializate. Datorită faptului că ele sunt folosite doar pe durate scurte de timp, ele se stochează ca resurse ale programului și se încarcă numai atunci când este necesar.

Inserarea unei casete de dialog se face în principiu în patru pași de bază:

1. Se crează o resursă de tip fereastră de dialog folosind mediul de dezvoltare.
2. Se folosește utilitarul ClassWizard pentru a crea o clasă derivată din clasa CDialog care va controla fereastra de dialog.
3. Se adaugă, dacă este necesar, funcții pentru a controla mesajele trimise ferestrei de dialog.
4. Dacă fereastra de dialog este selectată din meniul principal, se modifică adevarat resursa de tip meniu și se crează funcții pentru controlul mesajelor.

### **5.1. Crearea unei resurse de tip Dialog folosind mediul de dezvoltare**

Pentru a ilustra interactiv modalitatea creării unei astfel de resurse se construiește un proiect de tip document (*Single Document*) cu numele HelloSDI. Proiectul se crează în mod similar cazului proiectelor de tip fereastră de dialog, selectând de data aceasta opțiunea Single document.

Pentru a crea o resursă de tip Casetă de dialog se poate folosi una din metodele prezentate mai jos:

1. Din meniul **Project** se selectează sub-meniuul **Add Resource...**, apoi se selectează tipul resursei : **Dialog**.
2. Din meniul **View** se alege submeniu **Resource View**. Se expandează directorul HelloSDI.rc și se dă clic dreapta pe directorul **Dialog**, după care se alege comanda **Add Resource...** și se apasă butonul **New**.

Ca rezultat de afișează Editorul de Casete de dialog (Figura 8)

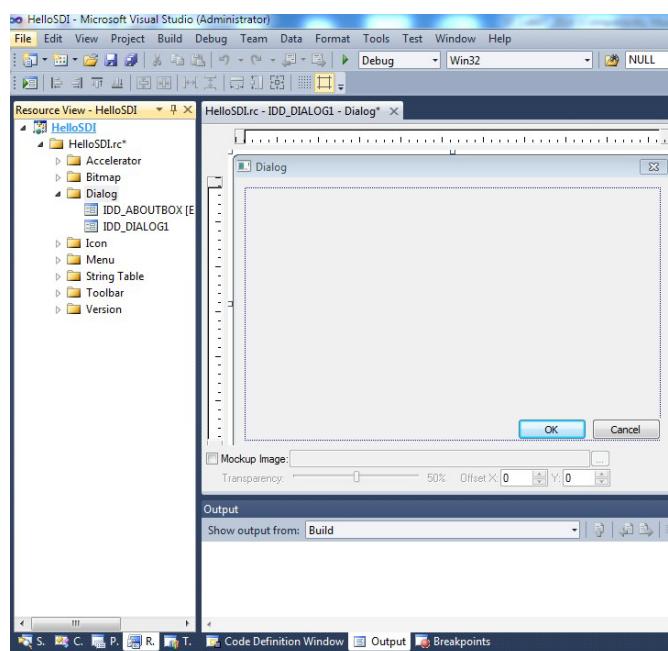


Figura 8. Editorul de Casete de dialog

### **5.1.2. Proprietățile Casetelor de Dialog**

Fiecare casetă de dialog are o serie de proprietăți care se pot selecta din meniul pop-up Properties (Figura 9).

- Atribute de stil (Eticheta Appearance) : 3D Look, Border, Minimize Box, Maximize Box, Style, Title Bar, System Menu, Horizontal Scroll, Vertical scroll, Clip Siblings, Clip Children, Caption, Absolute Align etc.
- Eticheta Behavior: Accept Files, Visible, Disabled, Set Foreground, System Modal etc.
- Eticheta Font: Font (Size), Use System Font.
- Eticheta Position: X Pos, Y Pos, Center
- Alte atribute (Eticheta Misc): ID, Center Mouse, Menu, Name, Local Edit, Context Help, Control, No Idle Message, No Fail Create etc.

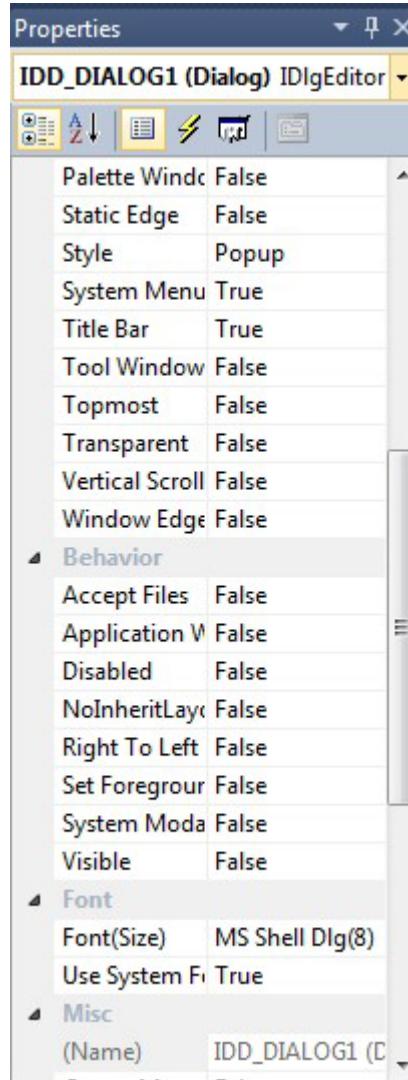


Figura 9.

### **5.1.3. Adăugarea unui control de tip text static (Static Text Control)**

Controlul de tip static nu necesită interacțiune cu fereastra de dialog, ci se utilizează, aşa cum s-a arătat și în cazul proiectului anterior, ca etichetă (eventual pentru alte controale).

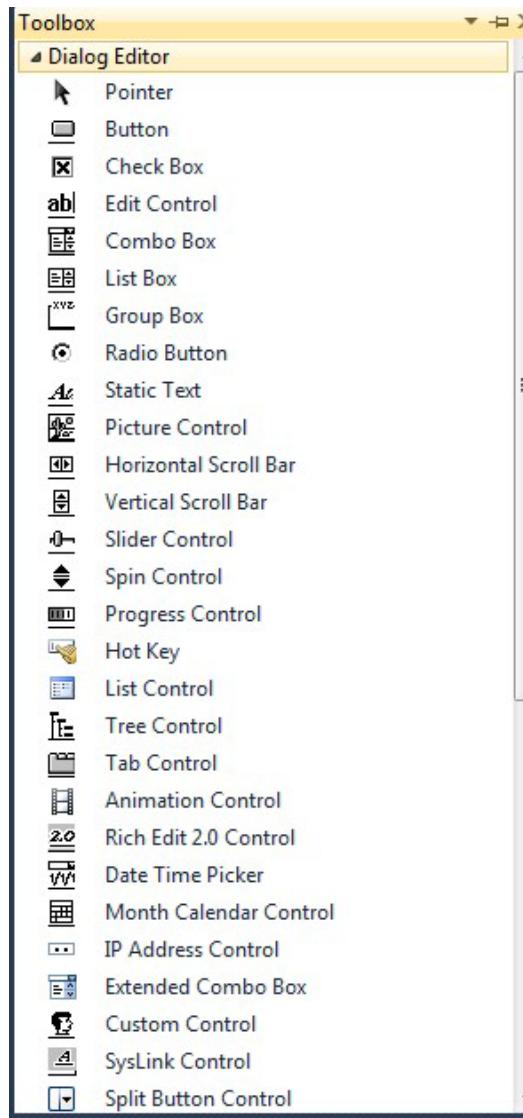


Figura 10.

Pentru a adăuga un astfel de control se parcurg următorii pași:

1. Se selecteză icoana **Static Text** control de pe bara de instrumente de control (**Toolbox**). Se observă schimbarea formei cursorului (în forma semnului plus).
2. Se plasează cursorul peste fereastra de dialog box, și se apasă pe butonul din stânga al mouse-ului. Se crează astfel un control de tip text static, ce conține eticheta **Static**.
3. Se modifică eticheta acționând butonul drept al mouse-ului în dreptul controlului și selectând din meniu comanda **Properties**. Modificați textul în "Hello World", ca în Figura 11. Se salvează fișierul.

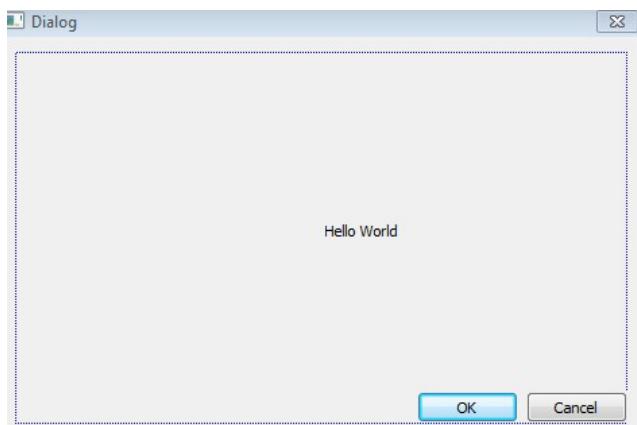


Figura 11.

## 5.2. Crearea unei clase pentru caseta de dialog

Clasa CDialog poate controla majoritatea interacțiunilor cu casetele de dialog ale unei aplicații, ea furnizând funcții membru adecvate acestui scop. Folosind utilitarul ClassWizard se poate deriva cu ușurință o nouă clasă adaptată cerințelor.

Lansarea utilitarului ClassWizard se poate face prin una dintre metodele de mai jos:

- Acționând combinația de taste Ctrl+Shift+X.
- Acționând butonul din dreapta al mouse-ului oriunde în interiorul editorului ferestrei de dialog și alegând ClassWizard din meniul pop-up.

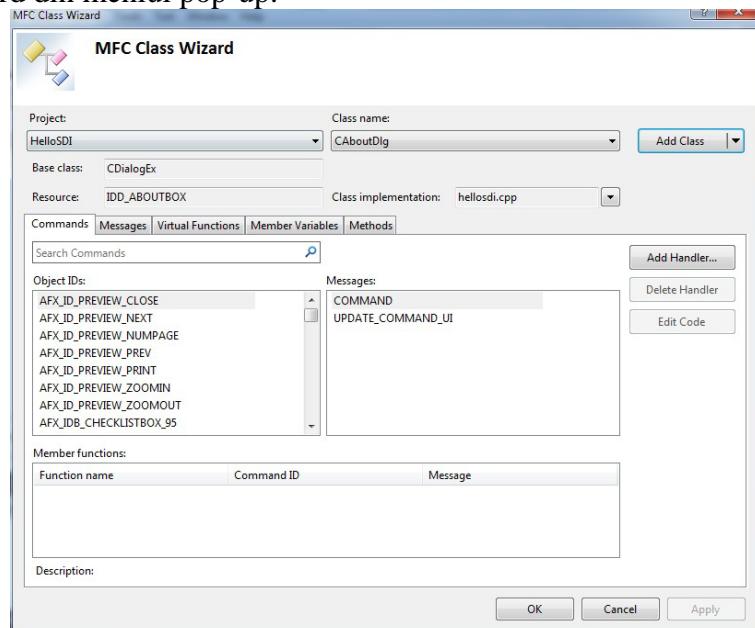


Figura 12.

ClassWizard recunoaște adăugarea unei noi resurse (în acest caz cea cu ID-ul IDD\_HELLODLG), utilizatorul este invitat să aleagă între două opțiuni:

- Să creeze o nouă clasă (uzual)
- Să selecteze o clasă existentă

Se afișează caseta de dialog New Class aşa cum se arată în figura 13.

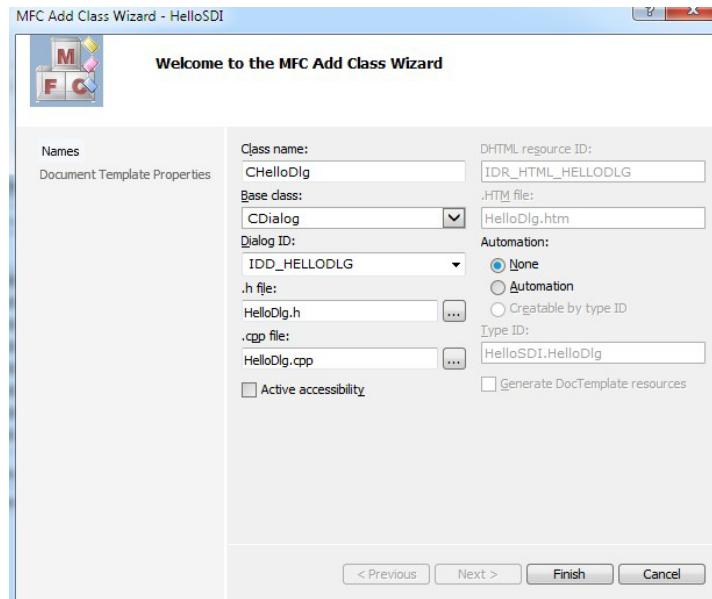


Figura 13. Fereastra de dialog *New Class*

Valorile furnizate în această casetă sunt folosite de ClassWizard pentru a crea o clasă ce va controla noua resursă de tip casetă de dialog.

Utilizați în acest scop valorile din tabelul 4 pentru caseta cu ID `IDD_HELLODLG`.

Tabelul 4. Exemplu de valori pentru completarea casetei New Class

Control	Value
Name	CHelloDlg
File Name	HelloDlg.cpp
Base Class	CDialog
Dialog ID	IDD_HELLODLG
OLE Automation	None

În felul acesta se generează clasa `CHelloDlg` class și se adaugă două fișiere în proiect :

- fișierul `HelloDlg.h` ce conține declarația clasei.
- fișierul `HelloDlg.cpp` ce conține definiția clasei.

### **5.3. Crearea unui handler de mesaje**

După adăugarea clasei `CHelloDlg` la proiectul HelloSDI, se vor adăuga funcții pentru tratarea mesajelor.

Se parcurg următorii pași:

1. Se selectează din meniul **View** submeniul **Class View** și se selectează din lista de clase, clasa care va trata mesajul, în acest caz `CHelloDlg`.
2. Se afișează fereastra **Properties** și se alege opțiunea **Overrides**.
3. Se selectează funcția `OnInitDialog` și se alege valoarea `Add OnInitDialog`.

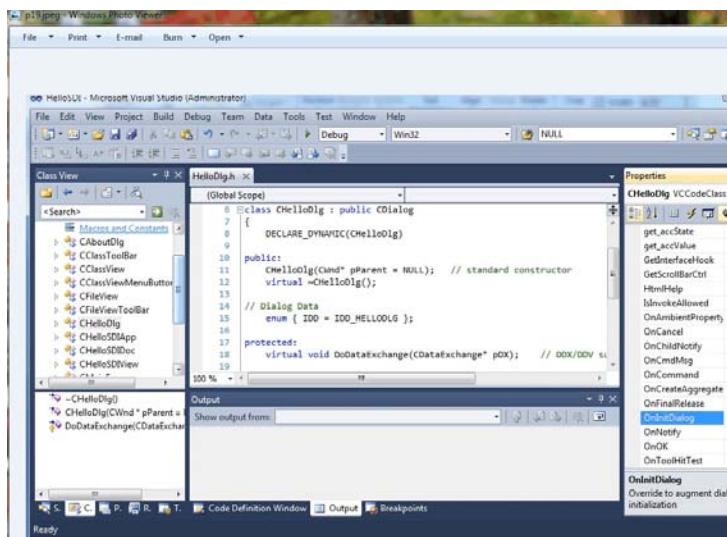


Figura 14.

Pentru exemplul ales funcția `CHelloDlg::OnInitDialog` va afișa doar un mesaj. Editați `OnInitDialog` în conformitate cu listingul 1.

#### **Listing 1. Funcția `CHelloDlg::OnInitDialog`.**

```
BOOL CHHelloDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    AfxMessageBox(_T("WM_INITDIALOG received" ));
    return TRUE;
}
```

#### **5.4. Adăugarea unei selecții de tip meniu pentru o casetă de dialog**

Urmează să ilustrăm modalitatea de a adăuga un articol într-un meniu folosit de proiectul HelloSDI.

Meniurile sunt stocate într-un proiect drept resurse. Pentru afișarea resurselor de tip meniu curente se selectează **ResourceView** din fereastra workspace a proiectului. Expandând folderul se pot observa diversele resurse definite pentru proiectul curent. Unul dintre foldere este denumit **Menu**.

Se deschide folderul **Menu** și se observă un articol de tip meniu denumit **IDR\_MAINFRAME**. Resursele de tip meniu se deschid cu ajutorul unui dublu click pe resursa corespunzătoare. În acest mod meniul poate fi editat. Prin intermediul unui click pe orice meniu din bara de meniuri, apare un meniu pop-up asociat selecției făcute, aşa cum se arată în Figura 15.

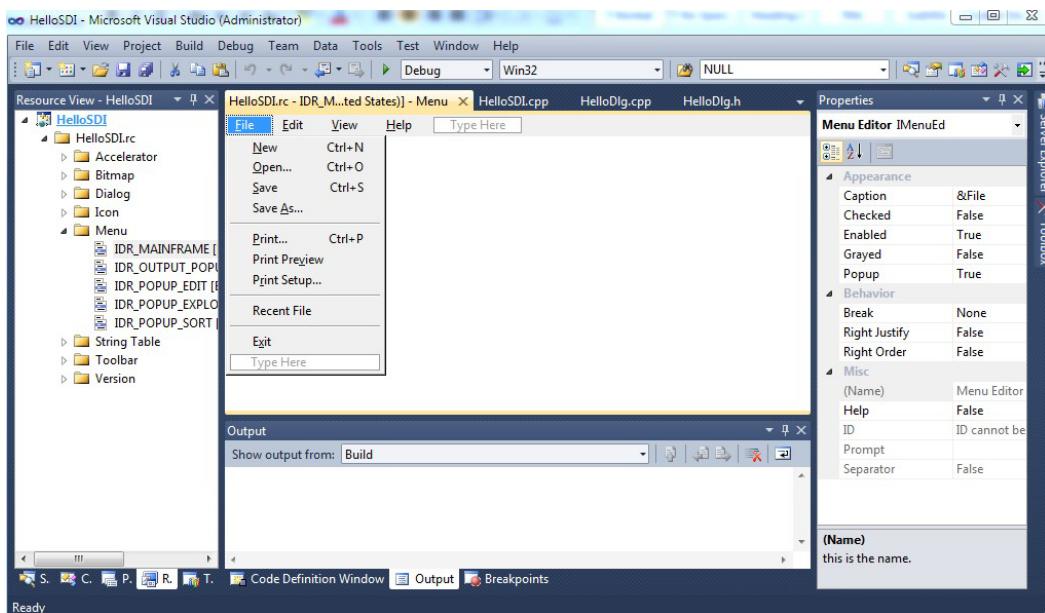


Figura 15. Utilizarea editorului de resurse pentru editarea meniurilor

Ultimul articol din orice meniu este o casetă goală care se poate folosi pentru a adăuga noi comenzi în meniul respectiv.

Pentru a adăuga un nou articol într-un meniu se fac următorii pași:

1. Se face dublu click pe caseta goală din meniu File pentru a afișa caseta de dialog Properties asociată meniului.
2. Pentru a adăuga un articol în meniu, se furnizează ID-ul meniului și un nume (titlu). Pentru acest exemplu, `ID_FILE_HELLO` ca ID și `&Hello` ca nume.
3. Click oriunde în afara casetei Properties pentru a vă întoarce în editor.

După adăugarea unui articol în menu, pasul următor este de a adăuga o funcție care să trateze noul articol. Pentru exemplul ales, adăugarea unei funcții pentru articolul cu ID-ul `ID_FILE_HELLO` se parcurg următorii pași:

1. Se selectează din meniu **View** submeniul **Class View** și se selectează din lista de clase, clasa care va trata mesajul, în acest caz, `CMainFrame`.
2. Se afișează fereastra **Properties** și se alege opțiunea **Events**.
3. Selectați obiectul care generează mesajul din lista de ID-uri, în acest caz, `ID_FILE_HELLO`.
4. Selectați mesajul `COMMAND` din lista de mesaje și acționați butonul `<Add>`. Acceptați numele implicit sugerat de ClassWizard: `OnFileHello`.
5. Acționați `OK` pentru a închide ClassWizard.

Editați funcția `CMainFrame::OnFileHello` în conformitate cu listingul 2.

**Listing 2. Funcție ce tratează mesajul comanda Hello din meniu.**

```
void CMainFrame::OnFileHello()
{
    CHelloDlg    dlgHello;

    if( dlgHello.DoModal() == IDOK )
        AfxMessageBox(_T("OK button pressed" ));
    else // IDCANCEL
        AfxMessageBox(_T("Cancel button pressed" ));
}
```

Adăugați o directivă `#include` în MainFrm.cpp ce conține definiția clasei pentru ChelloDlg din HelloDlg.h (deasupra directivei include pentru MainFrm.h)

```
#include "HelloDlg.h"
```

Compilați și executați proiectul HelloSDI.