# Tutorial sur les MFC

# 1 Avant de commencer

Il peut naturellement rester des fautes d'inattention. Si tel est le cas, veuillez m'en excuser.

Important :ce tutorial a pour but de programmer une interface graphique de façon PRATIQUE sans tomber dans<br/>les considérations techniques les plus poussées.Il s'adresse à un large public et je pars du principe que vous êtes assez grands pour chercher les<br/>informations complémentaires dont vous aurez besoin, tout seul.

# 2 Outils

Microsoft Visual C++

# 3 But

Nous voulons donc réaliser une interface graphique permettant d'additionner deux chiffres. C'est totalement inutile, sauf dans le cas de ce document :

MFCTest	×
Bonjour. Entrez 2 nombres :	<u> </u>
0 * 2	Annuler
Dialog	2
Sommer	OK
	Cancel
La somme es	t de : 2

# 4 Réalisation

# 4.1 Création du projet





/hat features would you like to include?    About box  Context-sensitive Help
About box     Context-sensitive Help     3D controls /hat other support would you like to include?     Automation
/hat other support would you like to include?
C Automation
and the second
ActiveX Controls
/ould you like to include WOSA support?
I windows Sockets
lease enter a title for your dialog:
MFCTest



CMFCTestApp CMFCTestDlg	ne rollowing glasses for you:
Cjass name:	Header file:
The second	
CMFCTestApp	MFCTest.h
CMFCTestApp Base class:	MFCTest.h

# 4.2 Création de l'interface graphique

Pour vous aider, faîtes apparaître la barre nommée « Control ». Un simple glisser-déplacer vous suffira alors pour mettre en place les éléments de l'interface.

Insérons un texte d'explication :

Version	OK     Image: Static Text       Annuler     Image: Static Text       Image: Static Text     Image: Static Text       Im

ajoutons un champ qui recevra le premier chiffre :

MFCT est resources *	MFETest	t.uhmtmi
IDD_ABOUTBOX IDD_MFCTEST_DIALOG ICon I	Bonjour. Entrez 2 nombres ; Edit X Cut Sopy Easte Insert Active≚ Control Size to Content I;t: Align Left Edges IT Align Top Edges Check Mnemonics ClassWizard Events Events	OK. Annuler

et définissons lui des propriétés (ici, donnons lui un nom clair. C'est surtout utile en présence de nombreux champs pour s'y retrouver) par un clic-droit - Propriétés :

Edit Propertie:	•	N 2
HA ? Ger	eral Styles Extended	Styles   13
	IT_NB1	
🔽 Visible	□ <u>G</u> roup □ <u></u>	<u>⊣</u> elp ID
Dis <u>a</u> bled	🔽 Tab stop	

ajouter de la même façon le deuxième champ (IDC\_EDIT\_NB2), ainsi qu'un bouton pour sommer :



### 4.3 Quelques notes avant de commencer à programmer

Une erreur bête peut être évitée en vérifiant que *tous* les fichiers incluent le fichier stdafx.h, si jamais vous deviez en ajouter un.



Le commentaire généré est assez (!) clair. En général il est indiqué où insérer votre code par un //TODO : ...



Avant de commencer, il va falloir initialiser les champs de l'interface graphique. En effet, la valeur d'un élément graphique est divisée en 2 variables :

- une variable de type CEdit (car nous manipulons un champ) qui définit ce qui va être affiché dans l'interface
- une variable de type int qui va contenir l'opérande à manipuler

En effet, on ne peut pas manipuler directement le contenu du champ. On est obligé de le mettre à jour à partir d'une seconde variable selon le schéma suivant (sur lequel nous reviendrons) :



Ainsi, il va falloir définir des variables associées aux 2 boutons :



Ajouter 2 variables :

Add Member Variable	<u>?</u> ×	Add Member Variable	<u>?×</u>
Member variable <u>n</u> ame: m_edit_nb1_y <u>Category:</u> Value Variable <u>type:</u> int T	OK Cancel	Member variable game: m_edit_nb1_d Category: Control Variable type: CE dit	DK Cancel
Description: int with validation		Description: map to CEdit member	

(ainsi, UpdateData(TRUE) copie les variables de types Control dans les variables de types Value)

#### Jusqu'à obtenir :

roject:	, Class n	amer		-
		anic. FestDla	-	Add Class 🔻
\Temp\MFCTest\MFCTest	lah C:\Temp\MECTes	t\MECTestDla.con		Add Variable
ontrol <u>I</u> Ds:	Туре	Member		Delete Variable
DC BUTTON SOMME	CEdit	m edit nb1 c	e d	Update <u>C</u> olumn
DC_EDIT_NB1 DC_EDIT_NB2 DC_EDIT_NB2 DCANCEL DOK	int CEdit int	m_edit_nb1_v m_edit_nb2_c m_edit_nb2_v		<u>Bind All</u>
escription;				

Ajoutons une méthode appelée lors de l'appuis sur le bouton somme. Retourner sur l'interface graphique :



cliquer sur le bouton somme et aller dans Edit<br/>+Classwizard :

message maps   me	mber variables   Automation   Actives Liverius   cia	ss mio
Project:	Class <u>n</u> ame:	Add Class 🔻
MFCTest	CMFCTestDlg	
):\\MFCTest\MFCTe	stDlg.h, D:\\MFCTest\MFCTestDlg.cpp	Add Function.
)bject <u>I</u> Ds:	Messages:	Delete Function
CMFCTestDlg	BN_CLICKED	Edu Pada
IDC_EDIT_NB1 IDC_EDIT_NB1 IDC_EDIT_NB2 IDCANCEL	E BN_DOUBLECLICKED.	Edir Code
IDC BOTTUN SUMM IDC EDIT_NB1 IDC EDIT_NB2 IDCANCEL IDOK fember functions:	E BN_DOUBLECLICKED	Earcoos
IDC BOTTUN SUMM IDC_EDIT_NB1 IDC_EDIT_NB2 IDCANCEL IDCANCEL IDOK Member functions:	E BN_DOUBLECLICKED	
IDC EDIT NB1 IDC EDIT_NB1 IDC EDIT_NB2 IDCANCEL IDDK Member functions: V DoDataExchange W OnInitDialog		
IDC BOTTON SUMM IDC EDIT_NB1 IDC EDIT_NB2 IDCANCEL IDOK Member <u>f</u> unctions: V DoDataExchange W OnInitDialog W OnPaint	E BN_DOUBLECLICKED ON_WM_INITDIALOG ON_WM_PAINT	
IDC EDIT NB1 IDC EDIT NB1 IDC EDIT NB2 IDCANCEL IDOK Aember functions: V DoDataExchange W OnInitDialog W OnPaint W OnQueryDraglcor	E BN_DOUBLECLICKED ON_WM_INITDIALOG ON_WM_PAINT ON_WM_QUERYDRAGICON	

sélectionner le bouton somme et la méthode d'appel (ici un simple clic). Faire « Add function » :



Cette méthode s'appellera OnButtonSomme. Elle est automatiquement insérée là où il faut (dans le .cpp et le .h).

il va falloir définir les valeurs d'initialisation pour les champs :



Le code est le suivant :

$m_edit_nb1_v = 0;$	11	on définit la première valeur à O
$m_edit_nb2_v = 2;$	11	on définit la deuxième valeur à 2
UpdateData(FALSE);	11	on fait « remonter » les valeurs dans
		les champs de l'interface graphique

# 4.4 Passons à la programmation de notre méthode Somme ()

Première étape : ajouter notre méthode à la classe CMFCTestDlg

MFCTest - Microsoft Visual C++ - [MFCTestDlg.cpp *	1
E Eile Edit View Insert Project Build Tools Window	Help
12 🖬 🖬 🖉   % 🖬 📾   🕮 + 😂 + 🚾 🕽	
CMFCTestDlg (All class members)	💊 OnButtonSomme 💽 🌂 🔹
	CRect rect; GetClientRect(▭); int x = (rect.Width() - cxlcon + 1) int y = (rect.Height() - cylcon + 1 // Draw the icon dc.DrawIcon(x, y, m_hIcon); } else { CDialog::OnPaint(); } he system calls this to obtain the curse the minimized window. SOR CMFCTestDlg::OnQueryDragIcon() return (HCURSOR) m_hIcon; CMFCTestDlg::OnButtonSomme() // TODO: Add your control notification ()

elle prend 2 variables de type int renvoie un int

function <u>Type</u>	£.		
int			
unction Decl	aration:		Lancel
Somme(int a,	int b)		
Access			
Public	C Protected	C Private	

finalement, cliquer sur le bouton « Sommer ! » appelle OnButtonSomme() puis Somme(int , int) de la classe CMFCTestDlg.

La méthode Somme () contient le code suivant :

```
int CMFCTestDlg::Somme(int a, int b)
{
     return(a+b);
}
```

Corsons un peu le tout nous allons voir 2 méthodes pour afficher le résultat :

- un affichage par une fenêtre popup simple
- un affichage par une fenêtre ayant un champ grisé

### 4.4.1 Un popup simple

Le code est :

```
void CMFCTestDlg::OnButtonSomme()
{
    // TODO: Add your control notification handler code here
    int res;    // le résultat du calcul
    CString Str_res;    // la valeur de la chaine de résultat
    // On va chercher la valeur entrée dans les champs
    // (on copie m_edit_nbX_c dans m_edit_nbX_v)
    UpdateData(TRUE);
    // on fait la somme des valeurs
    res = Somme(m_edit_nb1_v,m_edit_nb2_v);
    // on forme la chaîne contenue dans le popup
    Str_res.Format("La somme vaut %d",res);
    // on affiche le popup
    AfxMessageBox(Str_res);
```

qui affiche :

MFCTest	
Bonjour. Entrez 2 nombres :	Annuler
	MFCTest X
Sommer (	

### 4.4.2 Une fenêtre avec un champ

Nous allons déjà créer la boîte de dialogue qui va apparaître lors de l'appui sur somme.



Ne pas oublier de changer le nom du champ :

dit Properties		×
🗝 🎖 🛛 Gene	ral   Styles   Extended Styles	
ID: IDC EDI	RESULT	
🔽 Visible	□ <u>G</u> roup □ <u>H</u> elp ID	
Disabled	I Tab stop	

Cette fois, nous n'ajouterons que la variable de type Value ! En effet, s'agissant d'un champ de résultat, l'utilisateur n'aura pas à entrer un chiffre, donc nous n'auront pas besoin de faire un UpdateData(TRUE).

Allez dans le classwizard. Vous allez automatiquement générer la classe C++ liée à la boîte de dialogue nouvellement créée :

	Adding a Cla	55	-			1 ×		
	IDD_DIALOO Since it is a c create a new existing class	i_SOMME is a lialog resource : class for it. Yo	new resour vou probab u can also	ce. ly want to select an	OK Cano	:el		
	~							
	I Leate a C Select ar	new class nexisting class						
Net	w Class					?	×	
	Class information	n			-	OK.	1	
	<u>N</u> ame:	CDIgSomme				Cancel		
	File name:	DIgSomme.c	pp		1 -	Lancei		
				Change	1			
	Race class:		-		-			
	<u>D</u> ase class.	EDialog			1			
	Dialog ID:	IDD_DIALO	G_SOMME					
	<ul> <li>None</li> <li>Automation</li> <li>C Createable t</li> </ul>	ay type ID : MI	CTest Dig:	omme	-			
ClassWizar	None     Automation     C Egesteable b	ay type ID: M	CT est Dig	ĵomme				?
ClassWizar lessage Maps	Ngne     Automation     Createable.t	ay type ID : Mi	CTest Digt	Somme StiveX E vents	Class I	nfo		?
ClassWizar lessage Mapo oject:	Ngne     Automation     Careateable t	ables Autom	CTest Digt ation Ac ss <u>pame</u> :	Somme StiveX E vents	Class I	nfo	.dd Class	
ClassWizar Iessage Maps oject: IFCTest \\MFCTest	None     Automation     Createable t      d	ables Autom Cla MFCTest\Di	CTest Dig: ation Ac ss <u>name:</u> JigSomme sSomme.cp	comme stiveX E vents	Class I	nto	.dd Class	? 
ClassWizar Iessage Maps oject: IFCTest \MFCTest ontrol [Ds:	Ngne     Automation     Createable.b	ables Autom Cla MFCTest\Dly Type	CTest Digs ation Ac ss <u>pane:</u> JigSomme.cpj }	omme stiveX Events o 4ember	Class	nto	.dd Class Add Variab Delete Vari	? Je
ClassWizar dessage Maps roject: dFCTest :\MFCTest antrol IDs: DC EDIT RE DCANCEL	None     Automation     Createable t      Member Vari      DigSomme.h, D:	ables Autom Cla Cla CMFCTest\Di Type int	ation Ac ss <u>pame</u> : JSomme cpp r	Somme StiveX Events D Member n edit result	Class I	nto	.dd Class 3dd Variab 2elete Vari	iable
ClassWizar Message Maps roject: AFCTest :\\MFCTest :\\MFCTest DCANCEL DCANCEL DOK	Ngne     Automation     Geneticable t      d     Member Vari      S     Member Vari      SULT	ables Autom Cla 	ation   Au ss <u>pame</u> JgSomme gSomme.cpj r	comme ctiveX E vents o Member n edit result	Class I	nto	xdd Class 3dd Variab 2elete Vari pdate Eol Bind Al	iable umn
ClassWizar dessage Maps oject: IFCTest \\MFCTest ontrol [Ds: DC EDIT_RE DCANCEL DOK	None     Automation     Createable b      d      Member Vari      SULT	ables Autom Cla Cla CMFCTest\Dl, Type int	ation Ac s: <u>pame</u> : JgSomme tSomme.cpj t r	comme stiveX Events o Member n edit result	Class I	nto	dd Class Add Variab Qelete Vari pdate Dol Bind Al	iable I
ClassWizar dessage Maps oject: IFCTest \MFCTest ontrol [Ds: DC CANCEL DOCANCEL DOCANCEL	Ngne     Automation     Createable.t      Member Vari      NDIgSomme.h, D:      SULT	ables Autom Cla MFCTest\Dly Type int	ation Ac ss <u>pane:</u> JgSomme.cpj }	comme stiveX Events o dember n edit result	Class I	nfo	.dd Class 3dd Variab 2elete Vari pdate <u>F</u> ol Bind Al	. ~ ole iable
ClassWizar Iessage Maps oject: IFCTest \\MFCTest antrol IDs: DCANCEL DOK	None     Automation     Createable t      d      Member Vari      DIgSomme.h, D:      SULT	ables Autom Cla Cla Cla Cla Cla Cla Type int	ation Ar ss. pame: JgSomme.cpj ł	Comme StiveX Events S Member n edit result	↓ Class I	nto	xdd Class Add Variab Qelete Vari pdate Doll Bind Al	, T
ClassWizar dessage Maps oject: IFCTest X\MFCTest antrol [Ds: DC EDIT RE DCANCEL DOK	Ngne     Automation     Createable t      d     Member Vari      SULT	ables Autom Cla Cla CLMFCTest\Di, Type int	ation Ac ss <u>pame</u> JgSomme Somme.cpj r	ctiveX E vents	Class I		xdd Class Add Variab Qelete Vari pdate Dol Bind Al	iable
ClassWizar fessage Maps oject: IFCTest AMFCTest OCANCEL OCANCEL OCANCEL OOK	Ngne     Automation     Createable.b      d      Member Vari      SULT      int with validation	ables Autom Cla Cla 	ation   Ac s: <u>pane:</u> JgSomme.cpj r	comme stiveX Events o Member n edit result	V Class I	nfo	dd Class 3dd Variab 2elete Vari pdate Dol Bind Al	iable
ClassWizar fessage Maps oject: IFCTest MFCTest oc.EDIT RE DCANCEL DOK escription: inimum Value:	Ngne     Automation     Createable.t      Member Vari      More Vari      SULT	ables Autom Cla MFCTest\Dl Type int	ation Ac ss <u>pane:</u> JgSomme.cpj t	comme stiveX Events o dember n edit result	✓ Class I	nfo	dd Class 3dd Variab 2elete Vari pdate <u>D</u> ol Bind Al	, T

Une nouvelle classe (un fichier DlgSomme.cpp et un fichier DlgSomme.h) ont été créés. Allons jeter un coup d'œil dedans.

Les méthodes qui nous intéressent sont le constructeur et OnInitDialog().

Pour pouvoir afficher le résultat dans le champ, il va falloir transmettre le résultat à la boîte de dialogue. Nous pouvons par exemple modifier le constructeur

```
CDlgSomme::CDlgSomme(CWnd* pParent /*=NULL*/)
    :CDialog(CDlgSomme::IDD, pParent)
{
    //{{AFX_DATA_INIT(CDlgSomme)
    m_edit_result_v = 0;
    //}}AFX_DATA_INIT
```

qui devient

```
CDlgSomme::CDlgSomme(CWnd* pParent, int insertres)
        :CDialog(CDlgSomme::IDD, pParent)
{
        //{{AFX_DATA_INIT(CDlgSomme)
        m_edit_result_v = insertres;
        //}}AFX_DATA_INIT
}
```

Ne pas oublier de modifier DlgSomme.h!

```
class CDlgSomme : public CDialog
{
   // Construction
   public:
        CDlgSomme(CWnd* pParent = NULL); // standard constructor
   ...
```

devient

```
class CDlgSomme : public CDialog
{
    // Construction
    public:
        CDlgSomme(CWnd* pParent = NULL, int insertres = 0); // standard constructor
    ...
```

Ajoutons une fonction OnInitDialog() à la classe CDlgSomme :

Message Maps   Member Variable	es   Automation   ActiveX Events   Class In	10
Project:	Class name:	Add Class 🔻
MFCTest	CDIgSomme	
C:\\code\MFCTest\DlgSomme.h, )	C:\\MFCTest\DIgSomme.cpp	Add Function
Dbject <u>I</u> Ds:	Messages:	Delete Function
CDIgSomme IDC_EDIT_RESULT ID0K	WM_DRAWITEM WM_HELPINF0 WM_HSCROLL	Edit Code
	WM_INITDIALOG WM_KEYDOWN WM_KEYUP WM_KILLFOCUS	- -
Member <u>f</u> unctions:		
V DoDataExchange		
Description: Sent to a dialog box	before the dialog box is displayed	

Modifier ensuite OnInitDialog() (qui est dans la classe CDlgSomme, je le rappelle):

```
// TODO: Add extra initialization here
UpdateData(TRUE);
```

Il faut que la boîte de résultat apparaisse lors du clic sur le bouton Sommer ! Nous allons alors devoir ajouter l'appel à la fonction OnButtonSomme() de la classe CMFCTestDlg. Il faut avant tout inclure dans DlgSomme.cpp :

```
#include "DlgSomme.h"
```

puis modifier la méthode OnButtonSomme():

```
void CMFCTestDlg::OnButtonSomme()
{
    // TODO: Add your control notification handler code here
    int res; // le résultat du calcul
    CString Str_res; // la valeur de la chaine de résultat
    // On va chercher la valeur entrée dans les champs
    // (on copie m_edit_nbX_c dans m_edit_nbX_v)
    UpdateData(TRUE);
    // on fait la somme des valeurs
    res = Somme(m_edit_nbl_v,m_edit_nb2_v);
    // on instancie une boîte de dialogue...
    CDlgSomme dlg(NULL, res);
    // ... que l'on rend visible.
    dlg.DOModal();
}
```

## **5** Annexe

### 5.1 Réduire la taille de l'exécutable compilé

Par défaut, Visual C compile en mode debug. Pour notre exemple, l'exécutable généré pèse près de 1.4 Mo. Pour le réduire, il faut passer en mode release : dans le menu Build, sélectionner Set Active Configuration et choisir le mode release. La taille de votre exécutable passe alors à 120Ko environ.

### 5.2 Librairies statiques et dynamiques

Par défaut, et pour ce tutorial notamment, j'ai choisi d'utiliser des librairies statiques. Dans le cadre de librairies dynamiques, la taille de l'exécutable passe à 105Ko (mode debug), mais impose la présence dans le PATH des librairies nécessaires. A vous de voir.

	Statique	Dynamique
Debug	1400 Ko	120 Ko
Release	105 Ko	20 Ko

### 5.3 Analyse de l'exécutable par Scanbin

Nota : scanbin est un utilitaire de Jean-Claude Bellamy (www.bellamyjc.net).

L'exécutable a été compilé en mode release avec des librairies dynamiques.