Chapter 9

MFC

The Microsoft Foundation Classes provide all the classes needed to produce GUI (Microsoft) Windows programs. A typical development cycle with MFC involves using a rapid application development tool such as the wizard found in Visual C++, and then modifying the resultant source code. The RAD development cycle is relatively easy, but unfortunately, the second phase is not.

9.1   MFC menus

There are many ways to create menus in MFC, but it is common to use a special menu resource file. A resource file for a simple File/Quit menu might look like this:

```
#define MYAPP_EXIT 3210
MyApp MENU
    POPUP "File"
    {
        MENUITEM "Exit",MYAPP_EXIT
    }
}
```

In the **Create** call, you can do something like this:

```
Create( NULL, "Example", ..., CRect(...), NULL, "MyApp" );
```

The **MYAPP_EXIT** message may be bound using the **DECLARE_MESSAGE_MAP()** macro, and with the following declaration:
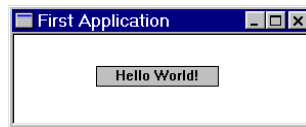
```
ON_COMMAND( MYAPP_EXIT,OnExit )
```

Finally, we need a message handler:

```
afx_msg void CMenusWin::OnExit()
{
    SendMessage( WM_CLOSE );
}
```

9.2   MFC Programming

Here is a simple initial example of an MFC application built using Microsoft Visual C++, modi-fied from the Deitel&Deitel MFC book:

The code to do this is here:

```
CODE LISTING                          FirstApp.cpp

#include <afxwin.h>

class CFirstWindow : public CFrameWnd {
public:
    CFirstWindow();
    ~CFirstWindow();
private:
    CStatic *m_pGreeting;
};

CFirstWindow::CFirstWindow()
{
    Create( NULL,
            "First Application",
            WS_OVERLAPPEDWINDOW,
            CRect( 100, 100, 400, 220 ) );
    m_pGreeting = new CStatic;
    m_pGreeting->Create(
        "Hello World!", // text
        WS_CHILD | WS_VISIBLE | WS_BORDER
        | SS_CENTER,
        CRect( 80, 30, 200,50 ),
        this );
}

CFirstWindow::~CFirstWindow()
{
    delete m_pGreeting;
}

class CFirstApp : public CWinApp {
public:
    BOOL InitInstance()
    {
        m_pMainWnd = new CFirstWindow();
        m_pMainWnd->ShowWindow( m_nCmdShow );
        m_pMainWnd->UpdateWindow();
        return TRUE;
    }

} FirstApp;
```
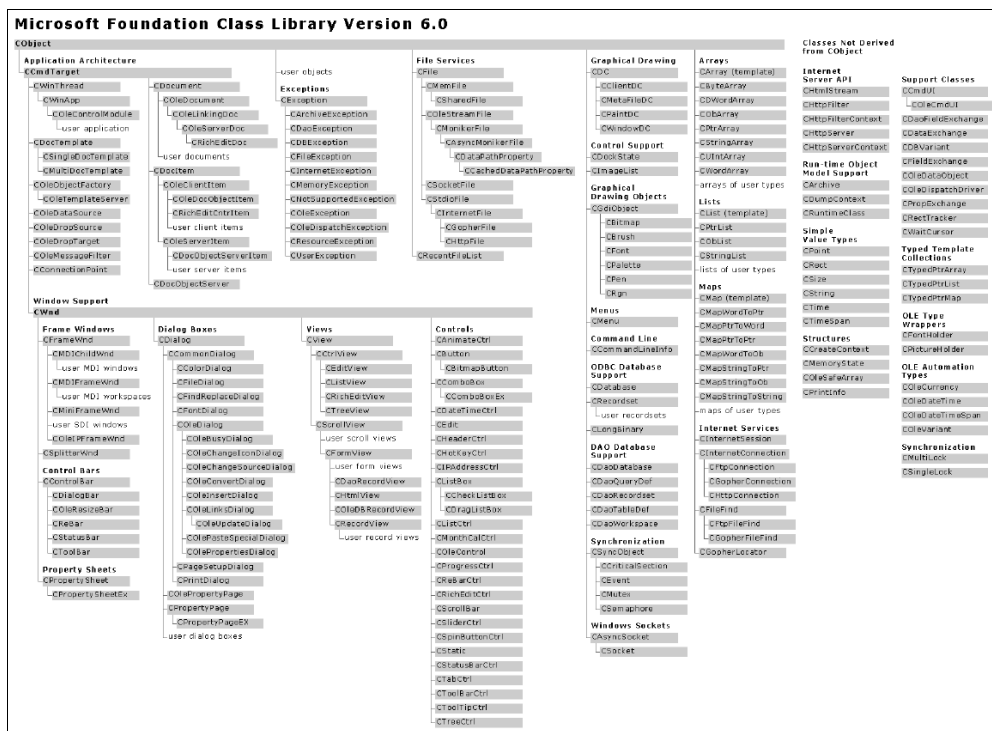
Note the use of Hungarian notation:

Prefix	Meaning
C	Class declaration
m_	Class member variable
p	Pointer
n or **i**	Integer
On	Event or message handler

This appears relatively easy, just instantiating a **Cstatic** object (a simple window). We can use much the same techniques to create dialogs (**CDialog**) or drawing windows (**CFrameWnd**)

9.3   MFC class hierarchy

The following heirarchy diagram shows the MFC components.

9.4   Summary of topics

In this module, we introduced the following topics:

- MFC
- MFC class heirarchy
- Simple programming

Questions for this module

1. Give a minimal menu driven application for MFC.
2. Compare the MFC message model with the Java Event model.
3. Outline a strategy for porting an MFC program to UNIX.
4. Outline a strategy for porting a Tcl/Tk program to MFC.