விண்டோஸ் புரோகிராமிங்

இன்று உலகில் உள்ள பெரும்பாலான கணிப்பொறிகளில் உபயோகப்படுத்தப்படும் ஆப்ரேட்டிங் சிஸ்டம் மைக்ரோசாஃப்ட் நிறுவனத்தின் விண்டோஸ் ஆப்ரேட்டிங் சிஸ்டமாகும். இந்த விண்டோஸ் ஆப்ரேட்டிங் சிஸ்டத்தில் இயங்கும்படியான அப்ளிகேஷன்களை உருவாக்க உதவுகிற டெவலப்மெண்ட் டூல்தான் விஷுவல் சி ப்ளஸ் ப்ளஸ் ஆகும். இது வரைகலை அடிப்படையிலான சாதாரண கால்குலேட்டர் அப்ளிகேஷன் முதல் டேட்டாபேஸ் நெட்வொர்க்கிங் போன்ற மிகப்பெரிய அளவிலான அப்ளிகேஷன்களை எளிதாகவும், விரைவாகவும் உருவாக்க உதவுகிறது. சி ப்ளஸ் ப்ளஸ் அறிந்தவர்கள் மிக விரைவாக விஷுவல் சி ப்ளஸ் ப்ளஸ் கற்றுக்கொள்ளலாம். விஷுவல் சி ப்ளஸ் ப்ளஸ் மூலம் ஒரு அப்ளிகேஷனை உருவாக்க முதலில் விஷுவல் சி ப்ளஸ் ப்ளஸ் சாப்ட்வேர் கணிப்பொறியில் முறைப்படி நிறுவப்பட்டிருக்க வேண்டும். இந்த டெவலப்மெண்ட் டூல் மைக்ரோசாஃப்ட் நிறுவன வெளியீடான விஷுவல் ஸ்டுடியோ பதிப்பில் உள்ளது. அதன்படி நிறுவப்பட்டிருப்பின் இந்நூலில் உதாரணத்திற்காக அளிக்கப்பட்டுள்ள சிறு சிறு அப்ளிகேஷன்களை செயல்படுத்திப் பார்த்து கற்றுக்கொள்ளலாம்.

டாஸ் அல்லது யூனிக்ஸ் ஆப்ரேட்டிங் சிஸ்டம் நிறுவப்பட்டுள்ள கணிப்பொறியில் Hello World எனும் ஒரு செய்தியை திரையில் வெளிப்படுத்தச் செய்ய கீழ்கண்ட ஐந்து வரியாலான சி புரோகிராம் கோடுகளை எழுதி செயல்படுத்தி பார்த்துவிடலாம்.

Hello.c

```
#include <stdio.h>
int main() {
 printf("Hello World!\n");
 return 0;
}
```

இதே Hello World செய்தியை கிராஃபிகல் யூசர் இண்டர்பேஸ் ஆப்ரேட்டிங் சிஸ்டமான விண்டோஸ் ஆப்ரேட்டிங் சிஸ்டம் நிறுவப்பட்டுள்ள கணிப்பொறியில்

வெளிப்படுத்தச் செய்ய முதலில் ஒரு விண்டோவை உருவாக்கி அதனுள்தான் இந்த செய்தியை வெளிப்படுத்தவைக்க முடியும். காரணம் எந்தவொரு விண்டோஸ் அப்ளிகேஷனும் அதற்கென ஒரு தனித்த விண்டோவில்தான் செயல்பட முடியும். இதன் காரணமாக மைக்ரோசாஃப்ட் நிறுவனம் சி, சி++ புரோகிராம் மூலமாக விண்டோஸ் அப்ளிகேஷன்களை உருவாக்க உதவுவதற்கென விண்டோஸ் புரோகிராமிற்கான ஃபங்ஷன்கள் போன்றவை அடங்கிய SDK (Software Development Kit) ஒன்றை வெளியிட்டுள்ளது.

எப்படி டாஸில் செயல்படும்படியான சி புரோகிராம்களை உருவாக்கி போர்லேண்ட் அல்லது டர்போ சி கம்ப்பைலர் உதவியுடன் கம்ப்பைலிங் மற்றும் லிங்கிங் செய்து exe பைலாக மாற்றி செயல்படுத்துகிறோமோ அதுபோல இந்த SDK உதவியுடன் விண்டோஸிற்கான சி புரோகிராம்களை உருவாக்கி கம்ப்பைலிங் மற்றும் லிங்கிங் செய்து exe பைலாக மாற்றி விண்டோஸ் ஆப்ரேட்டிங் சிஸ்டங்களில் இயங்கும் கணிப்பொறிகளில் செயல்படுத்தலாம்.

32 பிட் விண்டோஸ் ஆப்பரேட்டிங் சிஸ்டம் நிறுவப்பட்டுள்ள கணிப்பொறியில் உள்ள c:\windows\system டைரக்டரியில் user32.dll, gdi32.dll மற்றும் kernel32.dll ஆகிய மூன்று முக்கிய பைல்கள் காணப்படும். இந்த மூன்று பைல்கள்தான் விண்டோஸ் ஆப்பரேட்டிங் சிஸ்டத்தின் உள்ளீட்டு, வெளியீட்டு, கோப்பு மற்றும் நினைவக மேலாண்மை போன்ற முக்கிய பணிகளை கவனித்துக்கொள்கின்றன.

இதில் 32.dll பைலானது உள்ளீட்டு பணிகளை (Window Management, Including Messages, Menus, Cursors, Communications, Timer, etc) கவனித்துக்கொள்கிறது.

gdi32.dll பைலானது வெளீயிட்டு பணிகளை (User Interface and Graphics Drawing, Including Windows Metafiles, Bitmaps, Device Contexts and Fonts) பணிகளை கவனித்துக்கொள்கிறது. Graphics Device Interface என்பதன் சுருக்கமே GDI.

kernel32.dll பைலானது நினைவகம், கோப்புகள் போன்றவற்றை மேலாண்மை செய்யும் பணிகளை (Low-level Functions of Memory, Task and Resource Management) கவனித்துக்கொள்கிறது.

இந்த மூன்று முக்கிய பணிகளே Application Program Interface என்றழைக்கப்படுகிறது. விண்டோஸில் கொடுக்கப்படும் அனைத்துக் கட்டளைகளும் இவற்றின் மூலமாகத்தான் செயல்படுத்தப்படுகின்றன.

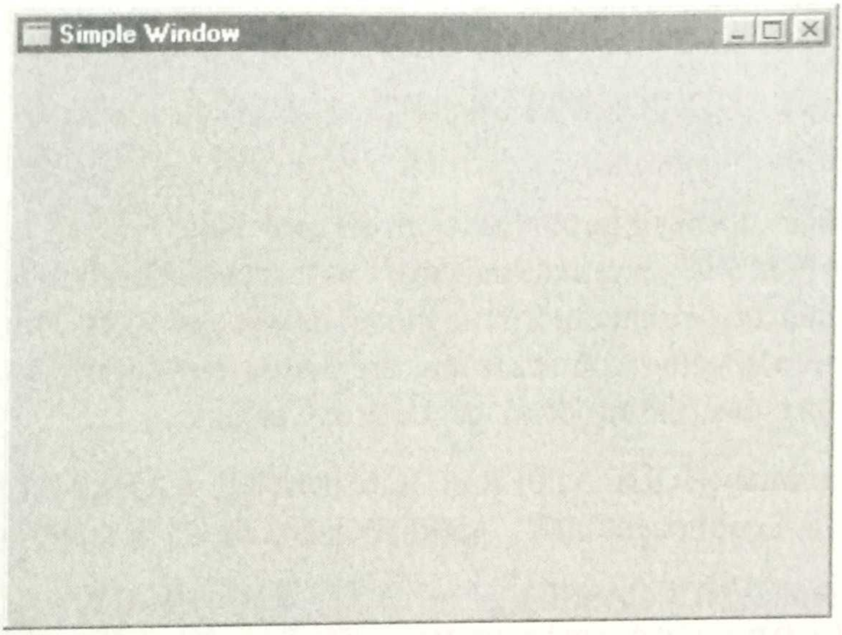
உள்ளீட்டு பணிகளுக்குத் தேவையான அணைத்து பங்ஷன்கள், ஆப்ஜெக்ட்டுகள், ரிசோர்ஸ்களும் user32.dll பைலில் வரையறுக்கப்பட்டுள்ளன.

அதுபோல வெளியீட்டுப்பணிகளுக்கு தேவையான அனைத்து பங்ஷன்கள், ஆப்ஜெக்ட்டுகள், ரிசோர்ஸ்களும் gdi32.dll பை-ல் வரையறுக்கப்பட்டுள்ளன.

நினைவகம், கோப்புகளை கையாளும் பணிகளுக்கான அனைத்து பங்ஷன்கள், ஆப்ஜெக்ட்டுகள், ரிசோர்ஸ்களும் kernel32.dll பை-ல் வரையறுக்கப்பட்டுள்ளன.

இந்த பங்ஷன்கள், ஆப்ஜெக்ட்டுகள், ரிசோர்ஸ்களை நன்கு அறிந்துகொண்டு விண்டோஸிற்கான சி, சி++ மொழி புரோகிராம் மூலமாக நம் விருப்பத்திற்கேற்ற விண்டோஸ் அப்ளிகேஷன்களை உருவாக்க முடியும்.

படம் 1.1இல் Simple Window என்ற தலைப்பிட்ட ஒரு வெற்று விண்டோவைக் காணலாம்.



படம் 1.1

இந்த வெற்று விண்டோவை உருவாக்க உதவும் simplewindow.c எனும் சி மொழி புரோகிராமை கீழே காணலாம். இந்த புரோகிராம் API ஃபங்ஷன்களை உபயோகித்து உருவாக்கப்பட்டதாகும்.

simplewindow.c

```
#include <windows.h>        //include all the basics

//API functions used: CreateWindowEx,DefWindowProc,DispatchMessage,GetMessage,
//GetSystemMetrics,LoadImage,MessageBox,PostQuitMessage,RegisterClassEx,ShowWindow,
//UpdateWindow,TranslateMessage,WinMain.
```

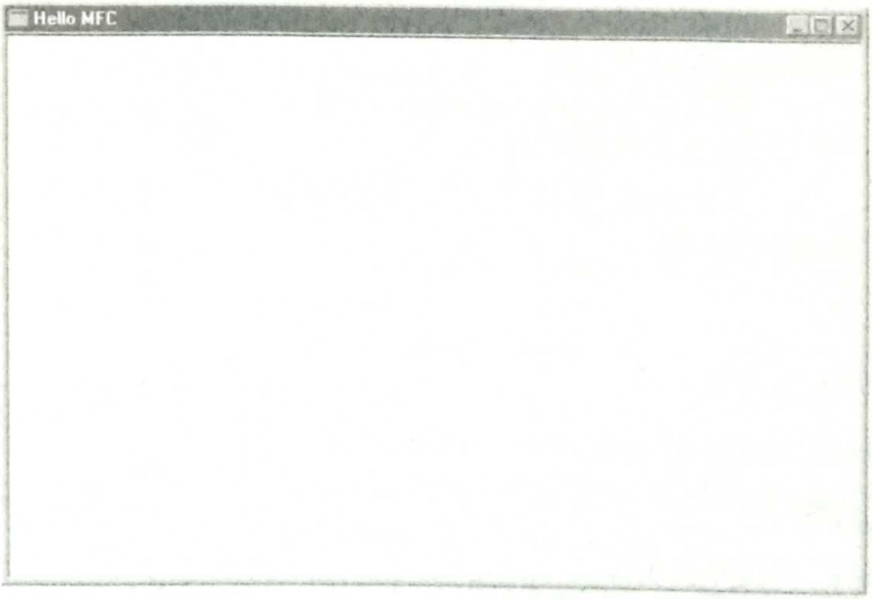
```
//declare the Window procedure where all messages will be handled LRESULT CALLBACK
WndProc(HWND hwnd,UINT Message,WPARAM wParam,LPARAM lParam);

//start the application; all win32 applications require a WinMain function
//that the windows operating system looks for as an entry point to that application

int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR
lpCmdLine,int nCmdShow)
{
HWND  hwnd;   //the wnd handle
MSG   Msg;    //a simple structure for storing message information
HICON  hIcon;  //window icon
HCURSOR hCursor; //window cursor

//declare and initialise wnd registration information variables

TCHAR chClassName[]=TEXT("SIMPLEWND");
WNDCLASSEX wcx;

//this structure is used for storing information about the wnd 'class'
//use 'LoadImage' to load wnd class icon and cursor as it supercedes the //obsolete functions
//'LoadIcon' and 'LoadCursor', although these functions will still work. Because the icon and
//cursor are loaded from system resources ie they are shared, it is not necessary //to free the
//image resources with either 'DestroyIcon' or 'DestroyCursor'.

hIcon=(HICON)LoadImage(0,IDI_APPLICATION,IMAGE_ICON,0,0,LR_SHARED);
hCursor=(HCURSOR)LoadImage(0,IDC_ARROW,IMAGE_CURSOR,0,0,LR_SHARED);

wcx.cbSize        = sizeof(WNDCLASSEX);                //byte size of WNDCLASSEX struct
wcx.style  = CS_HREDRAW|CS_VREDRAW;                   //ensure wnd is always redrawn
wcx.lpfnWndProc  = (WNDPROC)WndProc;                  //pointer to the Window Procedure
wcx.cbClsExtra    = 0;                                 //Extra bytes for 'class' wnds
wcx.cbWndExtra   = 0;                                  //Extra bytes for this wnd
wcx.hInstance     = hInstance;                         //Application instance
wcx.hIcon         = hIcon;                             //Application icon
wcx.hCursor       = hCursor;                           //Cursor for wnd
wcx.hbrBackground  = (HBRUSH)(COLOR_BTNFACE+1);  //Background wnd colour
wcx.lpszMenuName   = NULL;                             //Name of wnd menu
wcx.lpszClassName  = chClassName;                      //Name of this wnd 'class'
wcx.hIconSm        = NULL;                             //Icon in top-left corner of wnd

//Register the wnd class with the Windows system
if(!RegisterClassEx(&wcx))
```

```
  {
  //Registration has failed so inform the user
  MessageBox( NULL,
        TEXT("Failed to register wnd class"),
         TEXT("ERROR"),
          MB_OK|MB_ICONERROR);
  return FALSE;
  }

//create wnd of the 'class' just registered
hwnd=CreateWindowEx(0,                        //more or 'extended' styles
            chClassName,                 //the 'class' of window to create
           TEXT("Simple Window"),          //the window title
            WS_OVERLAPPEDWINDOW,           //window style: how it looks
          GetSystemMetrics(SM_CXSCREEN)/4,  //window position: left
           GetSystemMetrics(SM_CYSCREEN)/4,  //window position: top
           GetSystemMetrics(SM_CXSCREEN)/2,  //window width
           GetSystemMetrics(SM_CYSCREEN)/2,  //window height
             NULL,                        //parent window handle
             NULL,                        //handle to this windows's menu
             hInstance,                   //application instance
             NULL);                       //user defined information
if(!hwnd)
  {
  //wnd creation has failed so inform the user
  MessageBox( NULL,
        TEXT("Failed to create wnd"),
        TEXT("ERROR"),
         MB_OK|MB_ICONERROR);
  return FALSE;
  }

//Display the wnd

ShowWindow(hwnd,nCmdShow);
UpdateWindow(hwnd);

//start message loop

while (GetMessage(&Msg,NULL,0,0)>0)
  {
```

```
  TranslateMessage(&Msg);
  DispatchMessage(&Msg);
   }
return Msg.wParam;
}

LRESULT CALLBACK WndProc(HWND hwnd,UINT Message,WPARAM
wParam,LPARAM lParam)
{
switch (Message)
   {
   case WM_DESTROY:
     PostQuitMessage(0);   //signal end of application
      return 0;
   default:
     return DefWindowProc(hwnd,Message,wParam,lParam); //let system deal with msg
   }
}
```

சாதாரணமாக ஒரு வெற்று விண்டோவை உருவாக்க இத்தனை விண்டோஸ் பங்ஷன்களை உபயோகித்து நூற்றுக்குமேற்பட்ட வரிகளில் சி புரோகிராம் உருவாக்க வேண்டியிருக்கிறது. இந்த சிரமத்தை குறைப்பதற்காக மைக்ரோசாஃப்ட் நிறுவனம் சி++ புரோகிராமின் கிளாஸ் வசதியைக்கொண்டு MFC (Microsoft Foundation Classes) எனும் 2000ற்கும் மேற்பட்ட கிளாஸ்களைக் கொண்ட லைப்ரரியை வெளியிட்டுள்ளது. உண்மையில் இந்த MFC classesகள் API பங்ஷன்களை பயன்படுத்திதான் உருவாக்கப்பட்டுள்ளன. இந்த கிளாஸ்களை எந்தவொரு சி++ புரோகிராமிலும் சுலபமாக அப்படியே ஆப்ஜெக்ட்டாக அல்லது inherit செய்துகொண்டு இவற்றிலுள்ள டேட்டா மற்றும் மெம்பர் பங்ஷன்களை பயன்படுத்திக் கொள்ள முடியும்.

படம் 1.2இல் Hello MFC என்ற தலைப்பிட்ட ஒரு வெற்று விண்டோவைக் காணலாம்.



படம் 1.2

இந்த வெற்று விண்டோவை உருவாக்க உதவும் hellomfc.cpp எனும் சி++ மொழி புரோகிராமை கீழே காணலாம். இந்த புரோகிராம் MFC Classes களை உபயோகித்து உருவாக்கப்பட்டதாகும்.

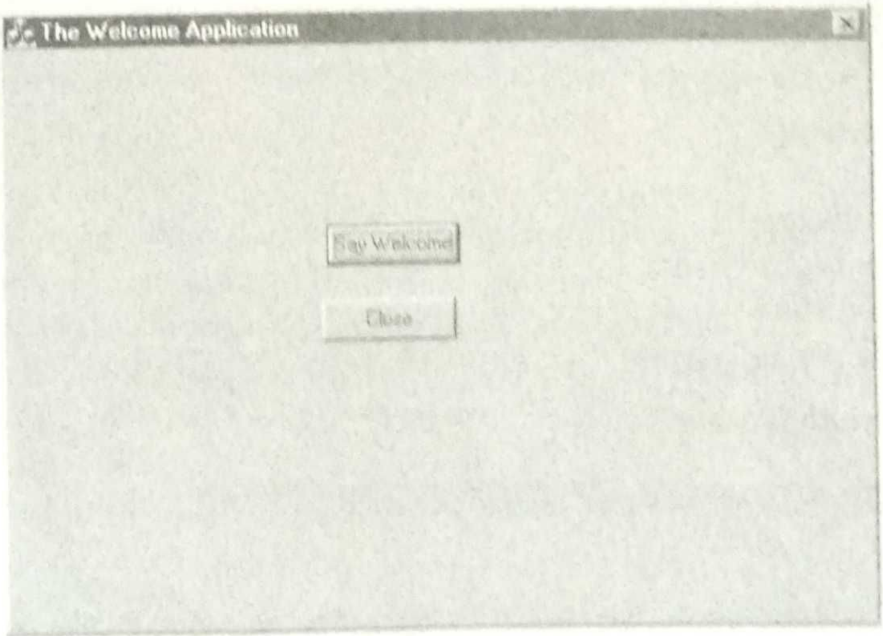
hellomfc.cpp

```
#include <afxwin.h>
class myframe:public CFrameWnd
{
public:
        myframe()
        {
                Create(0,"Hello MFC");
        }
};
class myapp:public CWinApp
{
public:
        int InitInstance()
        {
                myframe *p;
                p=new myframe;
                p->ShowWindow(1);
                m_pMainWnd=p;
                return 1;
        }
};
myapp a;
```

இப்படி புரோகிராம் கோடுகள் எழுதுகிற சிரமத்தையும் குறைப்பதற்கான மைக்ரோசாஃப்ட் நிறுவனம் தனது விஷுவல் சி ப்ளஸ் ப்ளஸ் டெவலெப்மெண்ட் டூலில் AppWizard, ClassWizard எனும் இரண்டு விசார்டுகளை புகுத்தியுள்ளது. Wizard என்றால் மந்திரவாதி என்று அர்த்தம். விண்டோவில் பெருமளவு உபயோகப்படுத்தப்படும் டயலாக் பேஸ்டு, சிங்கிள் டாகுமெண்ட் இண்டர்பேஸ் மற்றும் மல்ட்டிபிள் டாக்குமெண்ட் இண்டர்பேஸ் போன்றவகையான அப்ளிகேஷன்களை இந்த விசார்டுகளின் உதவியுடன் மிகச் சுலபமாக உருவாக்கிவிடலாம். இதற்கு இந்த விசார்டில் உள்ள பல்வேறு வசதிகளில் தேவையானவற்றை தேர்வு செய்து அதற்கான பதில்களை அளித்தால் போதும். இந்த விசார்டுகள் நமது விண்டோஸ் புரோக்ராமிற்கு தேவையான அடிப்படையான சி++ புரோகிராம் கோடுகளை எழுதிவிடும். இனி இந்த MFC AppWizard மற்றும் Class Wizard ஆகிய விசார்டுகளின் உதவியுடன் எவ்வாறு விண்டோஸ் அப்ளிகேஷன்களை உருவாக்குவது எனக் காண்போம்.

WELCOME அப்ளிகேஷன்

நாம் இப்போது WELCOME எனும் சிறு அப்ளிகேஷனை உருவாக்கப்போகிறோம். அதை இயக்கினால் இரண்டு புஷ் பட்டன்கள் கொண்ட The Welcome Application என்ற தலைப்புடன் ஒரு விண்டோ தோன்றும். பார்க்க படம் 2.1.



படம் 2.1.

அதில் Say Welcome என்ற புஷ் பட்டனை க்ளிக் செய்தால் Welcome to II என்ற செய்தியுடன் ஒரு விண்டோ தோன்றும். பார்க்க படம் 2.2.

படம் 2.2.

இந்த விண்டோ மெஸேஜ் பாக்ஸ் என்றழைக்கப்படுகிறது. இதிலுள்ள OKஇல் பட்டனை க்ளிக் செய்தால் இது மறைந்துவிடும்.

The Welcome Application என்ற தலைப்பு கொண்ட விண்டோவிலுள்ள Close என்ற புஷ் பட்டனை க்ளிக் செய்தால் அப்ளிகேஷன் நிறைவு செய்யப்படும்.

இனி மேற்கண்ட இந்த பணிக்கான அப்ளிகேஷனை எவ்வாறு விஷுவல் சி ப்ளஸ் ப்ளஸ் மூலம் உருவாக்குவது எனக் காண்போம்.

பொதுவாக ஒரு சட்டம் தோன்றும்படி புரோக்ராம் எழுதுவதற்குகூட விண்டோசில் மிக அதிக அளவில் கோடிங் எழுத வேண்டிவரும். இந்த சிரமத்தைக் குறைப்பதற்காக Class Wizard, App Wizard போன்ற இரண்டு விசார்டுகள் விஷுவல் சி ப்ளஸ் ப்ளஸில் உள்ளது. இவற்றைப் பயன்படுத்துவதன் மூலம் ப்ரோக்ராம் எழுதுவதற்கான நேரத்தையும் உழைப்பையும் மிச்சப்படுத்தலாம்.

MFC AppWizard என்ற விசார்டைப் பயன்படுத்தி நமது அப்ளிகேஷனிற்கான புரோக்கிராமை உருவாக்கப் போகிறோம். இந்த விசார்ட் நமது விண்டோஸ் புரோக்கிராமிற்கு தேவையான அடிப்படையான கோடுகளை எழுதிவிடும். இதற்கு இந்த விசார்டில் உள்ள பல்வேறு வசதிகளில் தேவையானவற்றை தேர்வு செய்து அதற்கான பதில்களை அளித்தால் போதும்.

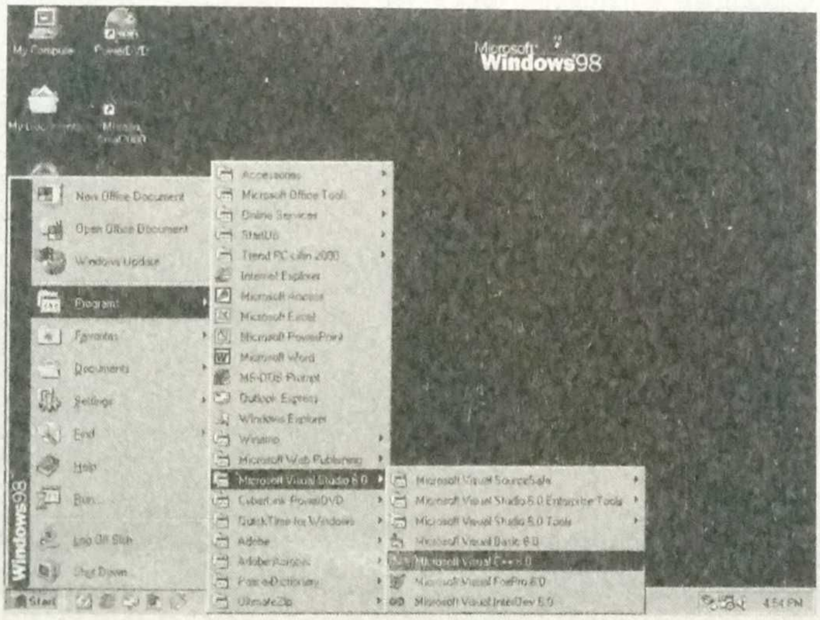
விஷுவல் சி ப்ளஸ் ப்ளஸை இயக்க கீழ்க்கண்டவாறு செயல்படுங்கள்.

1. விண்டோஸ் டெஸ்க்டாப்பில் உள்ள Start பட்டனை க்ளிக் செய்யுங்கள்

2. Start மெனுவிலுள்ள Programs என்பதை க்ளிக் செய்யுங்கள்

3. Programs மெனுவிலுள்ள Microsoft Visual Studio 6.0 என்பதை க்ளிக் செய்யுங்கள்.

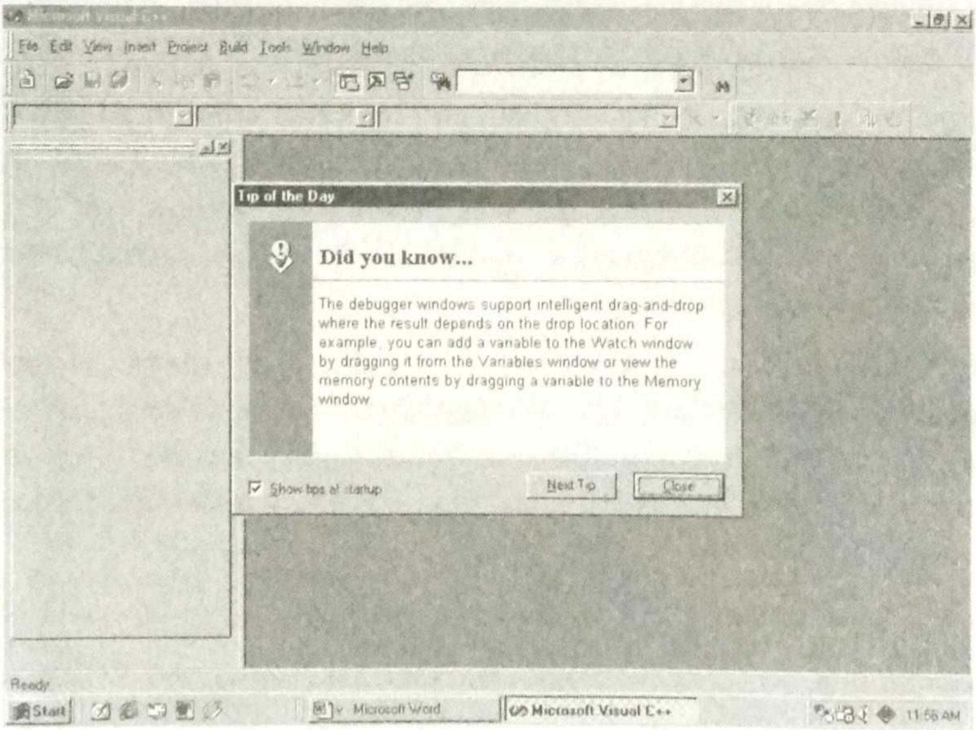
4. Microsoft Visual Studio 6.0 மெனுவிலுள்ள Microsoft Visual C++ 6.0 என்பதை க்ளிக் செய்யுங்கள். பார்க்க படம் 2.3.



படம் 2.3

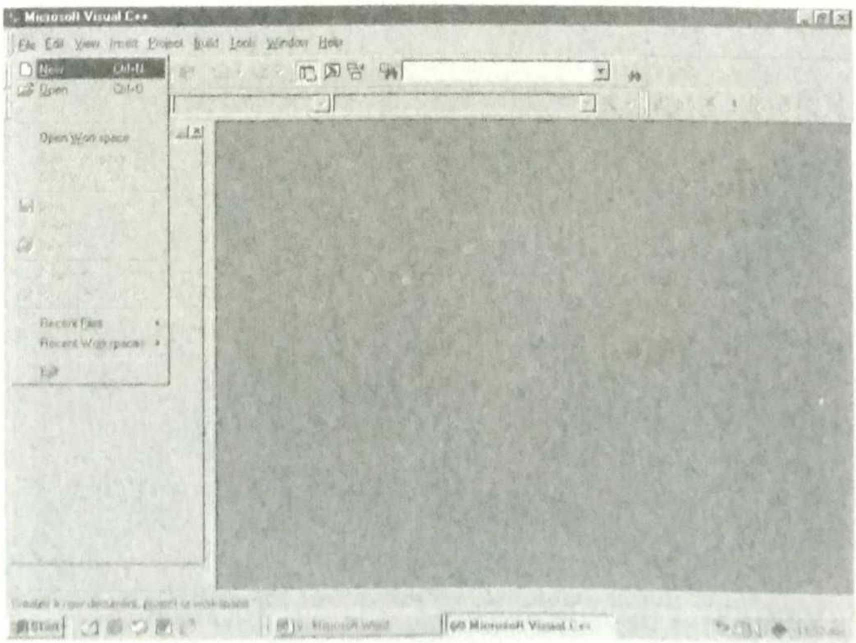
இப்போது படம் 2.4இல் உள்ளவாறு விஷுவல் சி ப்ளஸ் ப்ளஸின் மெயின் விண்டோ வெளிப்படுத்தப்படும்.

(Tip of the Day எனும் டயலாக் பாக்ஸ் வெளிப்படுத்தப்பட்டிருந்தால் அதன் Close பட்டனை க்ளிக் செய்து அதை நிறைவு செய்து விடுங்கள்.)



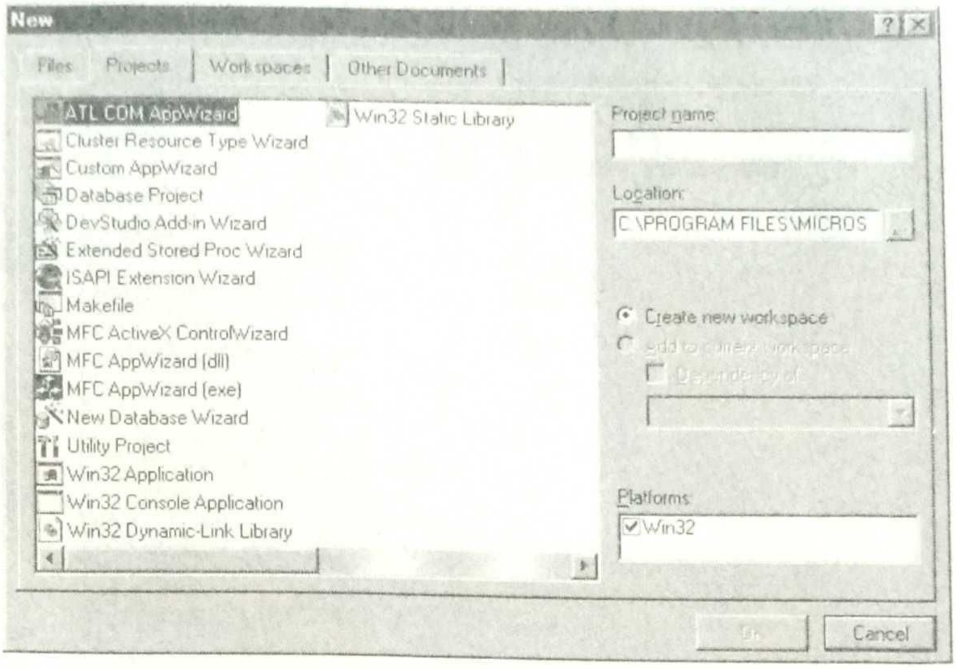
படம் 2.4

இதில் File மெனுவில் உள்ள New என்பதை க்ளிக் செய்யுங்கள். படம் 2.5.



படம் 2.5

இப்போது New எனும் தலைப்புடன் ஒரு டயலாக் பாக்ஸ் Projects என்ற டேப் செலக்ஷனுடன் வெளிப்படுத்தப்படும். படம் 2.6 (விஷுவல் சி ப்ளஸ் ப்ளஸில் ஒரு சிறு புரோகிராமைக்கூட ப்ராஜெக்ட் எனும் முறையில்தான் உருவாக்க முடியும்.)

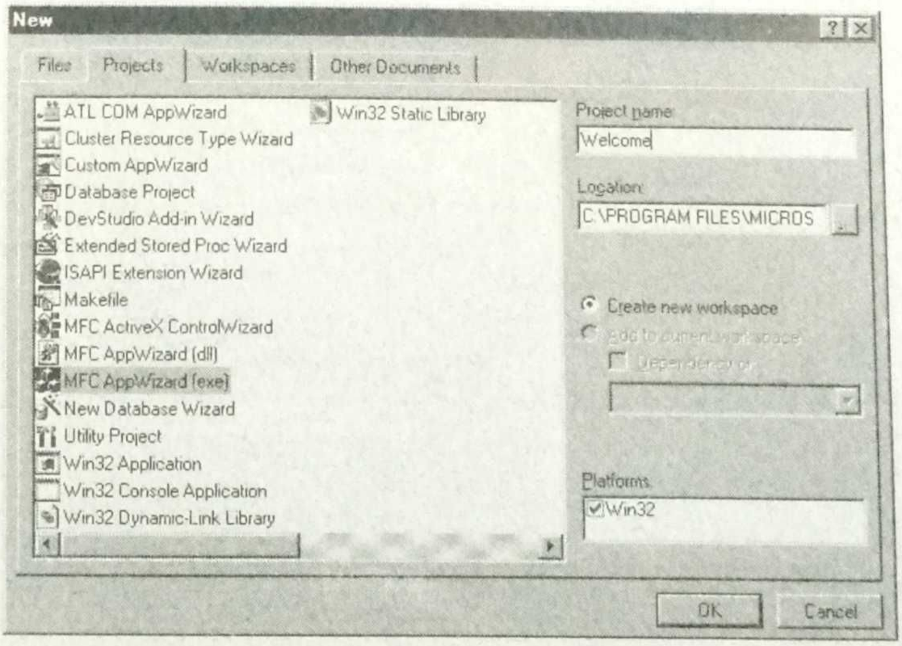


படம் 2.6

இதில் MFC AppWizard[exe] என்பதை க்ளிக் செய்யுங்கள்.

ப்ராஜெக்ட் பெயர் பட்டிய-ல் Welcome என டைப் செய்யுங்கள்

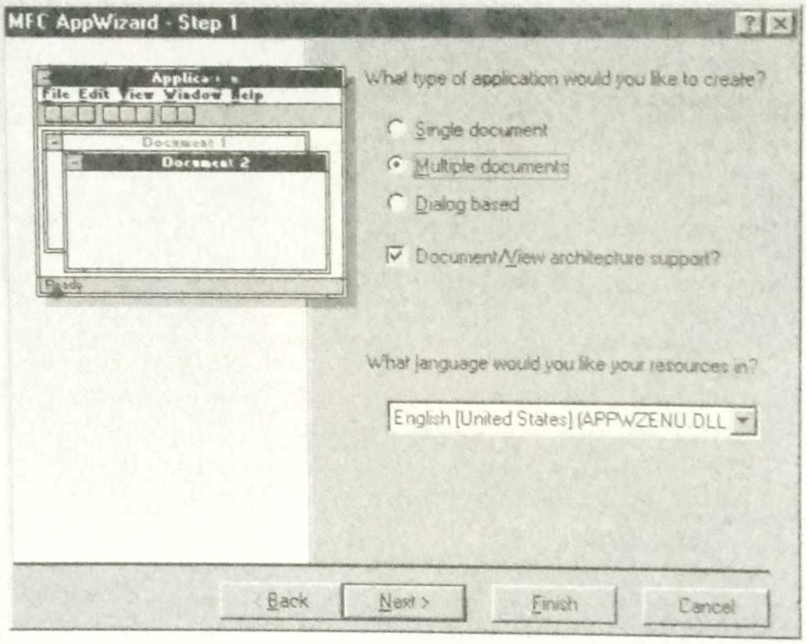
லொக்கேஷன் பட்டிய-ல் வேண்டிய டைரக்டரியை டைப் செய்யலாம் அல்லது அப்படியே விட்டுவிடலாம். படம் 2.7



படம் 2.7

இனி OK பட்டனை க்ளிக் செய்யுங்கள்.

இப்போது MFC AppWizard-Step 1 எனும் தலைப்புடன் ஒரு டயலாக் பாக்ஸ் வெளிப்படுத்தப்படும். பாக்க படம் 2.8.

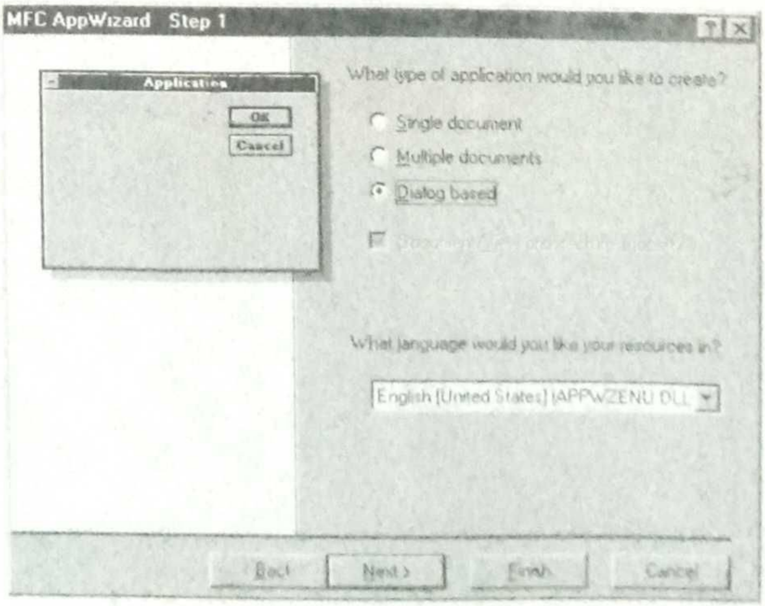


படம் 2.8

இதில் Dialog based எனும் ரேடியோ பட்டனை க்ளிக் செய்யுங்கள். (காரணம் நமது Welcome அப்ளிகேஷன் ஒரு டயலாக் பாக்ஸாக இருப்பதனால்)

What language would you like your resources in?
English [United States] (APPWZENU.DLL)

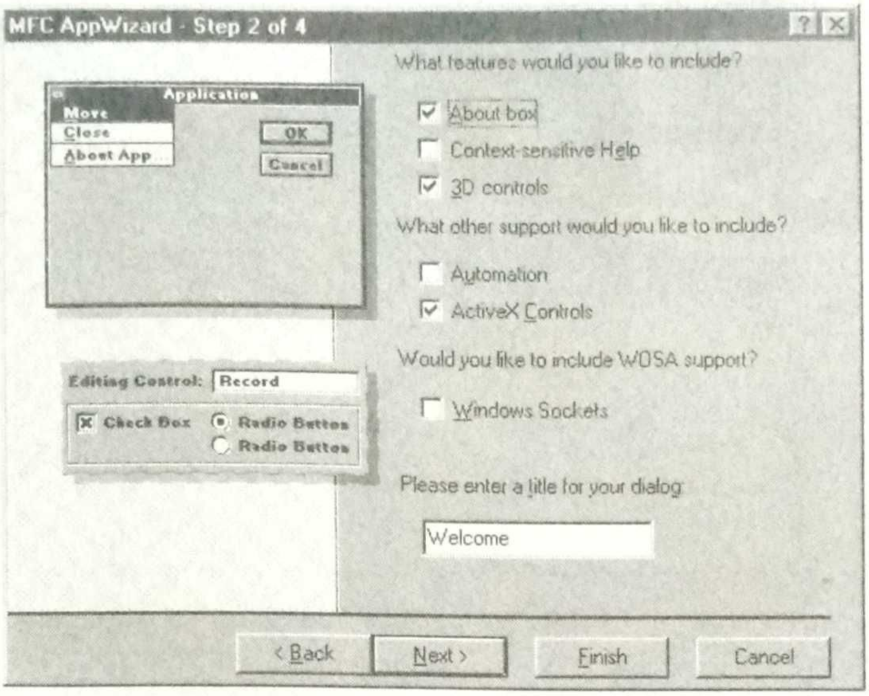
எனும் Combo Box ஐ அப்படியே விட்டுவிடவும். படம் 2.9.



படம் 2.9

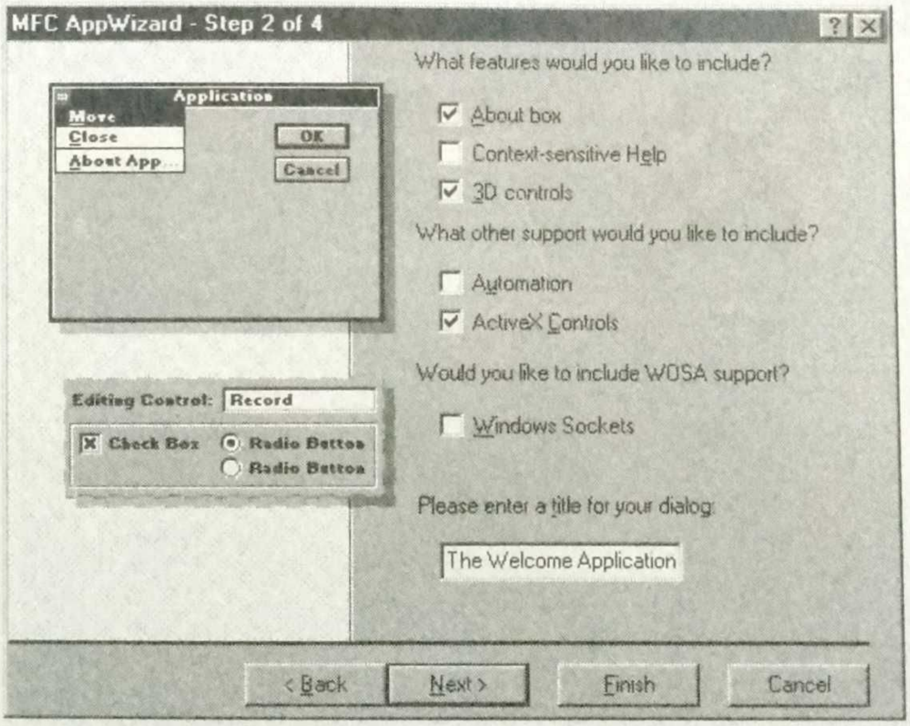
இனி Next பட்டனை க்ளிக் செய்யுங்கள்.

இப்போது MFC Appwizard-Step 2 of 4 எனும் தலைப்புடன் ஒரு டயலாக் பாக்ஸ் வெளிப்படுத்தப்படும். படம் 1.10.



படம் 2.10

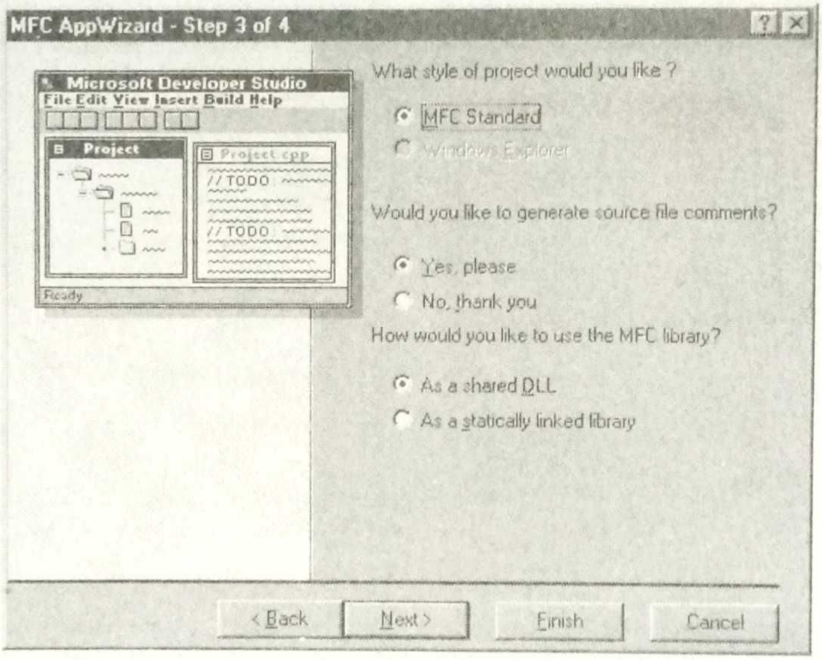
இதில் Please enter a title for your dialog : எனும் டைட்டில் டயலாக் பாக்ஸில் மட்டும் The Welcome Application என்று டைப் செய்யுங்கள். படம் 2.11



படம் 2.11

இனி Next பட்டனை க்ளிக் செய்யுங்கள்.

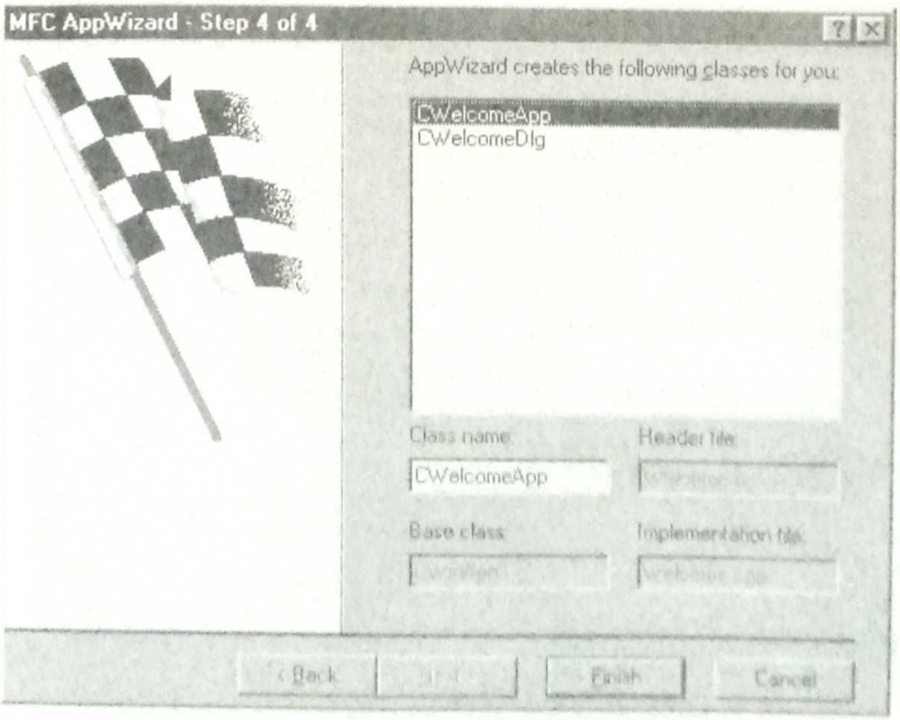
இப்போது MFC Appwizard-Step 3 of 4 எனும் தலைப்புடன் ஒரு டயலாக் பாக்ஸ் வெளிப்படுத்தப்படும். படம் 1.12.



படம் 2.12

இதில் Next பட்டனை மட்டும் க்ளிக் செய்யுங்கள்.

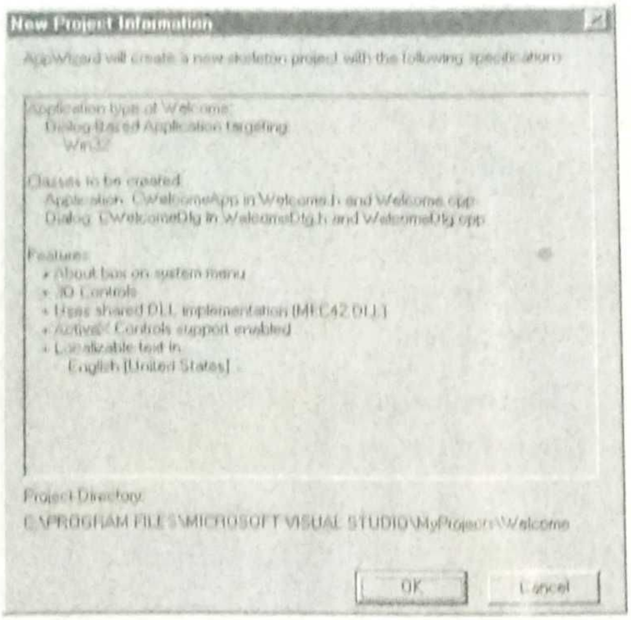
இப்போது MFC Appwizard-Step 4 of 4 எனும் தலைப்புடன் ஒரு டயலாக் பாக்ஸ் வெளிப்படுத்தப்படும். படம் 1.13.



படம் 2.13

இதில் Finish பட்டனை க்ளிக் செய்யுங்கள்.

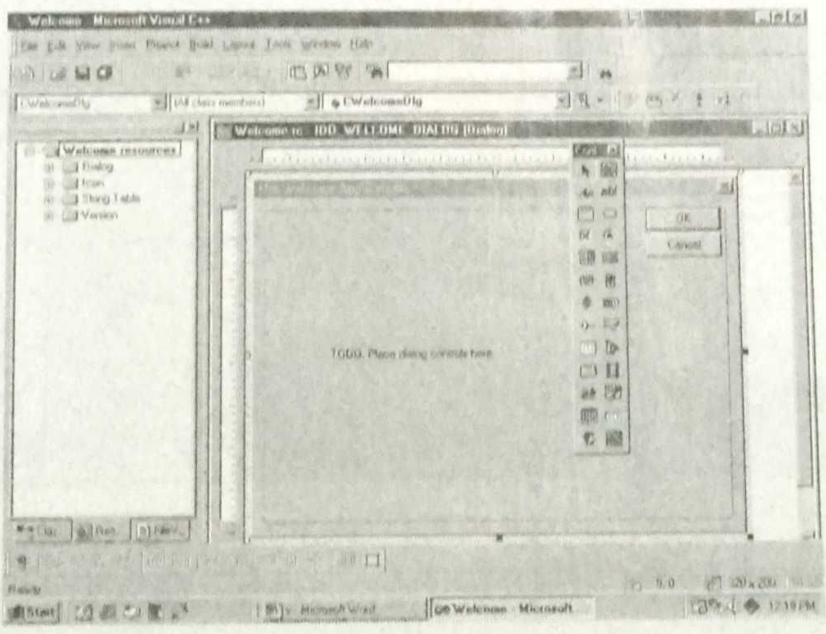
இப்போது New Project information எனும் டயலாக் பாக்ஸ் வெளிப்படுத்தப்படும். இதில் நாம் இதுவரை தேர்வு செய்து அளித்த பதில்களுக்குத் தகுந்தாற்போல் உருவாக்கிய ஹெடர் மற்றும் புரோக்ராம் பைல்களின் விவரங்கள் வெளிப்படுத்தப்படும். படம் 2.14.



படம் 2.14

இதில் OK பட்டனை க்ளிக் செய்யுங்கள்.

இப்போது விஷூவல் சி ப்ளஸ் ப்ளஸின் விண்டோ Welcome என்ற தலைப்புடன் தோன்றும் மற்றும் Welcome resources என்ற டயலாக் பாக்ஸ் வெளிப்படுத்தப்பட்டிருக்கும். இது Project Workspace என்றழைக்கப்படுகிறது. படம் 2.15 (இந்த ப்ராஜெக்ட் வொர்க்ஸ்பேஸ் தெரியவில்லையெனில் View மெனுவில் உள்ள Workspace என்பதை க்ளிக் செய்யுங்கள்.)



படம் 2.15

இந்த Project Workspace இல் கீழ்க்கண்ட நான்கு டேபுகள் வெளிப்படுத்தப்பட்டிருக்கும்.

Class View
Resource View
File View

விசார்ட் நமக்காக உருவாக்கிய கிளாஸ் பைல்களை Class View வில் பார்வையிட, எடிட் செய்ய முடியும்.

Dialog Boxes, Menus, Icons, BMP Pictures போன்ற ரிசோர்ஸ்களை Resource Viewவில் பார்வையிட, எடிட் செய்ய முடியும்.

ப்ராஜெக்ட் ஃபைல்களை File View வில் பார்வையிட முடியும்.

இனி நமது அப்ளிகேஷனிற்கான விஷுவல் தோற்றத்தை நாம் விரும்பியபடி எப்படி உருவாக்குவது என்று காண்போம்.

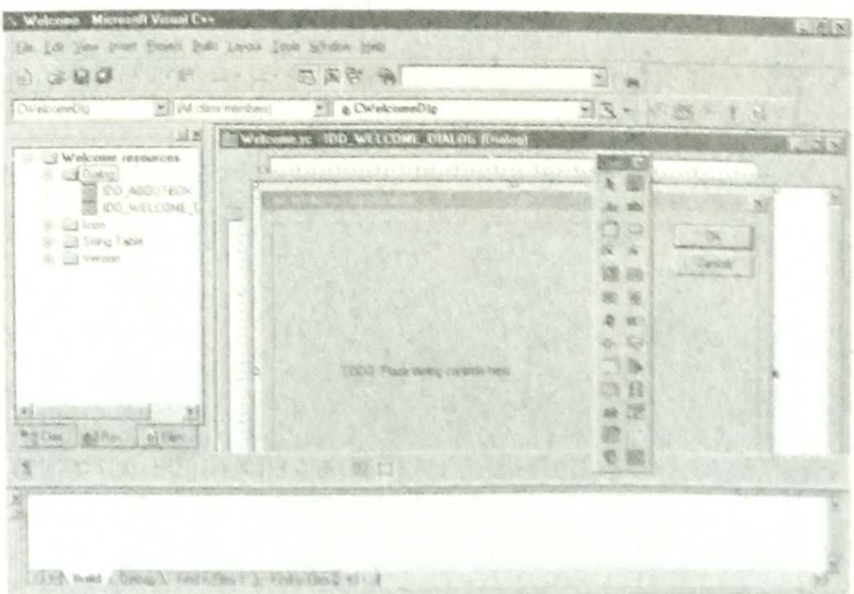
Resource View எனும் டேபை க்ளிக் செய்யுங்கள். (இயல்பாக இந்த Resource டேப்தான் தேர்வு நிலையில் இருக்கும்,

அடுத்து அதன் கீழ் வெளிப்படும் Welcome resources என்பதை டபுள் க்ளிக் செய்யுங்கள் (அல்லது ப்ளஸ் ஐகானை க்ளிக் செய்யுங்கள்.)

இப்போது Welcome Resources ற்கு கீழ் Dialog, Icon, String Table, Version எனும் நான்கு துணை ஐட்டங்கள் வெளிப்படுத்தப்படும்.

Dialog ஐட்டத்தை டபுள் க்ளிக் செய்யுங்கள் (அல்லது ப்ளஸ் ஐகானை க்ளிக் செய்யுங்கள்.)

இப்போது Dialog ஐட்டத்திற்கு கீழ் IDD_ABOUTBOX, IDD_WELCOME_DIALOG எனும் இரண்டு துணை ஐட்டங்கள் வெளிப்படுத்தப்படும். படம் 2.16.



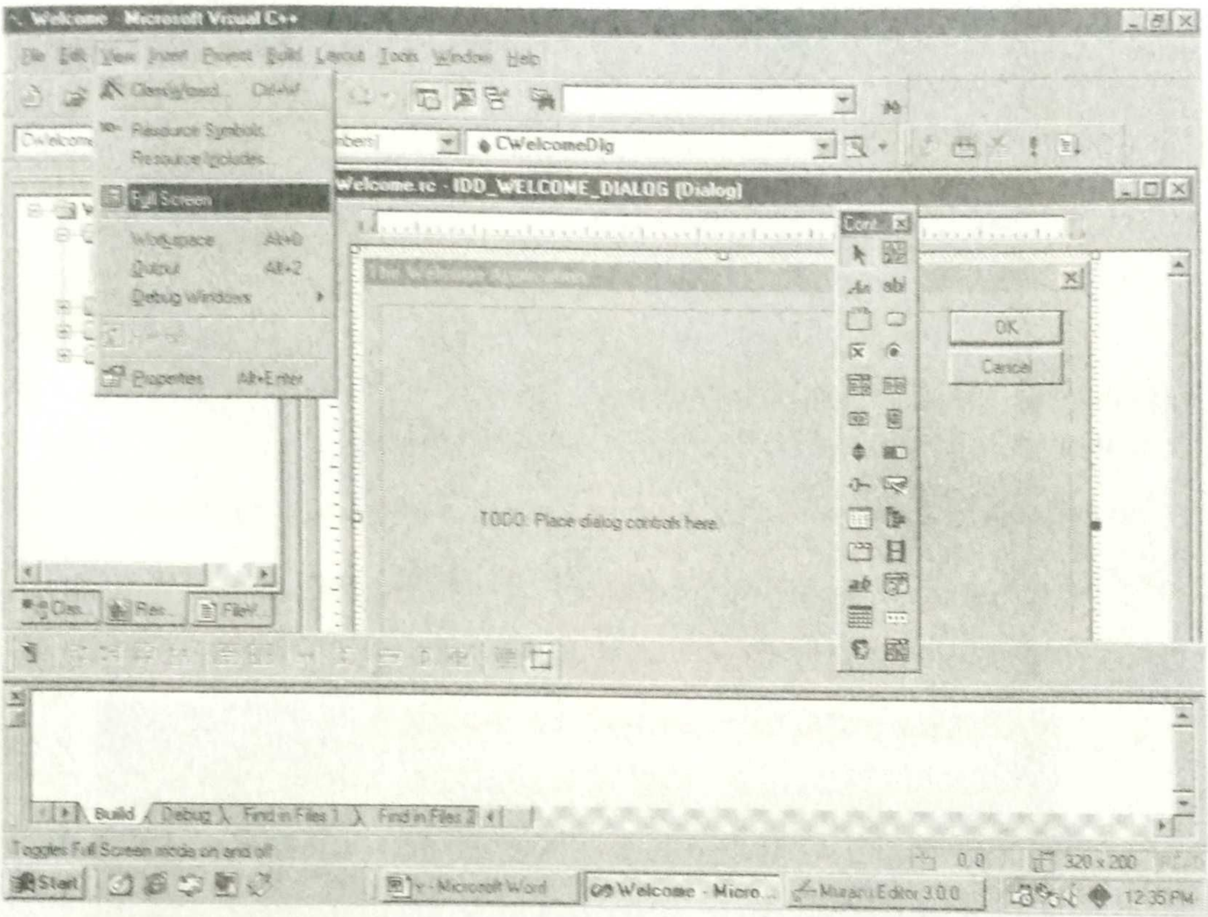
படம் 2.16

இதில் IDD_WELCOME_DIALOG ஐட்டத்தை டபுள் க்ளிக் செய்யுங்கள்.

இப்போது இந்த ஐட்டம் சைன் லோடில் இருக்கும்.

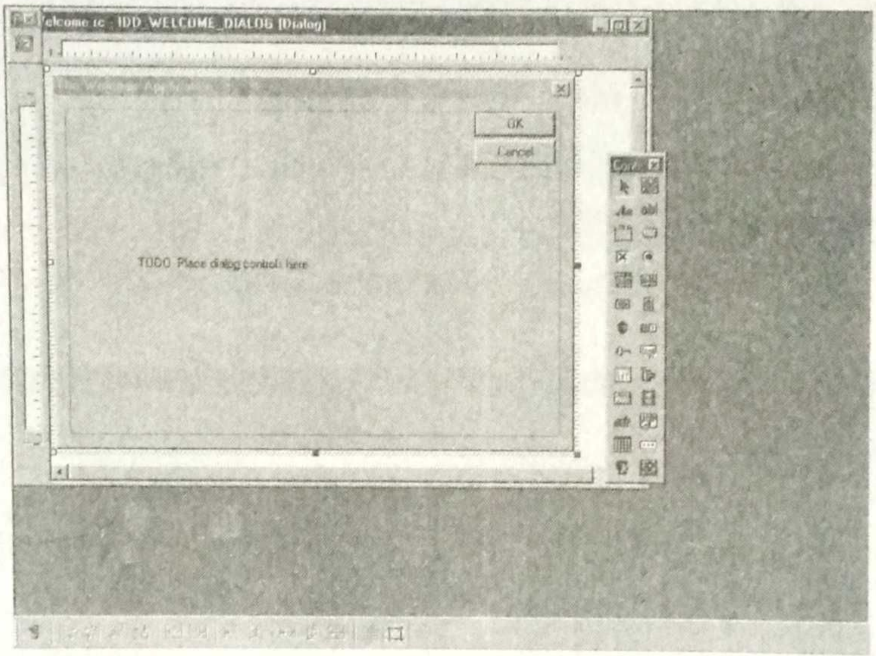
(Welcome.rc - IDD_WELCOME_DIALOG(Dialog) என்ற தலைப்புடன் கண்ட்ரோல் டூல்பாருடன் வெளிப்படுத்தப்பட்டிருக்கும்.)

View மெனுவில் Full Screen என்பதை க்ளிக் செய்யுங்கள். படம் 2.17



படம் 2.17

இப்போது OK push button, Cancel push button, TODO: Place dialog controls here. எனும் டெக்ஸ்ட் கொண்ட மூன்று கன்ட்ரோல்களுடன் ஒரு டயலாக் பாக்ஸ் வெளிப்படுத்தப்படும். உடன் கன்ட்ரோல் டூல்பாரும் வெளிப்படுத்தப்படும். படம் 2.18. (கன்ட்ரோல் டூல்பார் வெளிப்படுத்தப்படவில்லையெனில் Tools மெனுவில் Customized என்பதை க்ளிக் செய்யுங்கள்.)



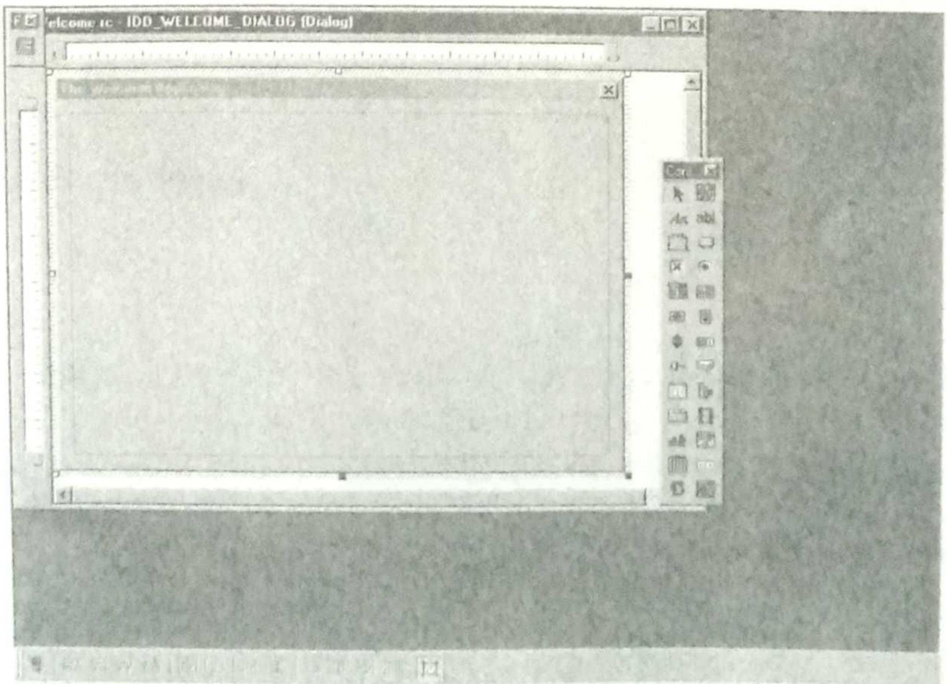
படம் 2.18

இப்போது இந்த மூன்று Control களையும் டெ-ட் செய்ய வேண்டும். இனி OK புஷ் பட்டனை க்ளிக் செய்து டெ-ட் கீயை அழுத்துங்கள்.

Cancel புஷ் பட்டனை க்ளிக் செய்து டெ-ட் கீயை அழுத்துங்கள்.

TODO: டெக்ஸ்ட்டை க்ளிக் செய்து டெ-ட் கீயை அழுத்துங்கள்.

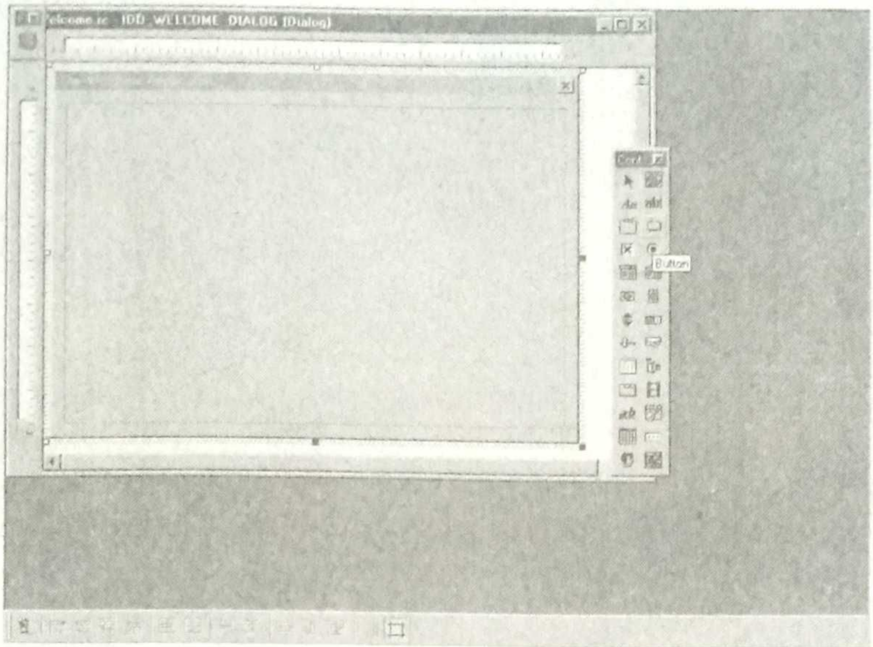
இப்போது டயலாக் பாக்ஸ் கா-யாக இருக்கும். படம் 2.19



படம் 2.19

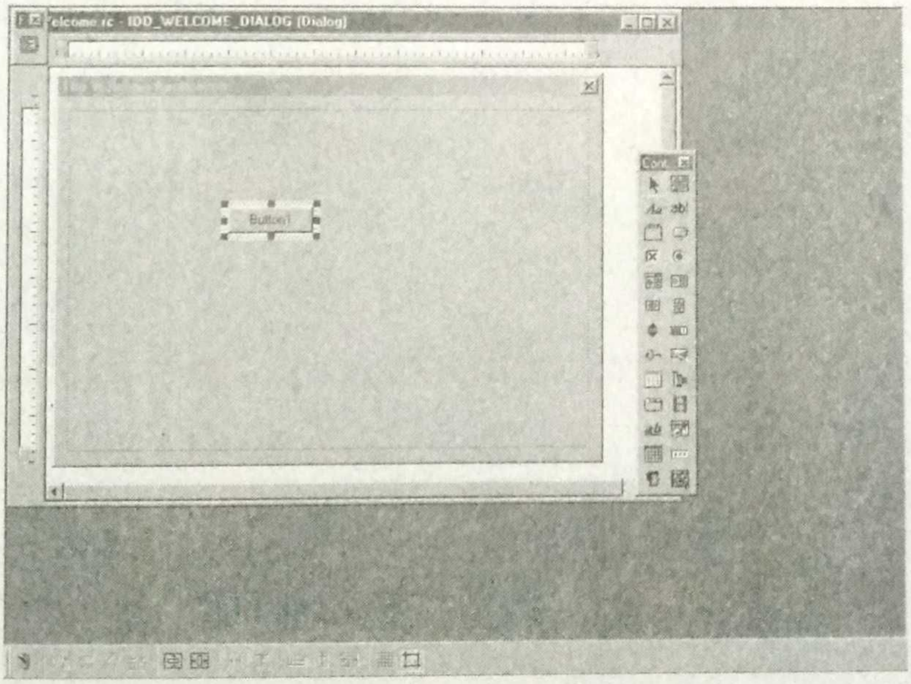
Control டூல்பாரிலுள்ள Controlகள் மீது மவுஸ் பாயிண்டரை கொண்டு சென்றால் அவற்றின் பெயர்கள் வெளிப்படுத்தப்படும்.

அதில் புஷ் பட்டன் Controlஐ க்ளிக் செய்யுங்கள். உடன் டயலாக் பாக்ஸ் மீது மவுஸ் பாயிண்டரை கொண்டு வந்து க்ளிக் செய்யுங்கள். படம் 2.20.



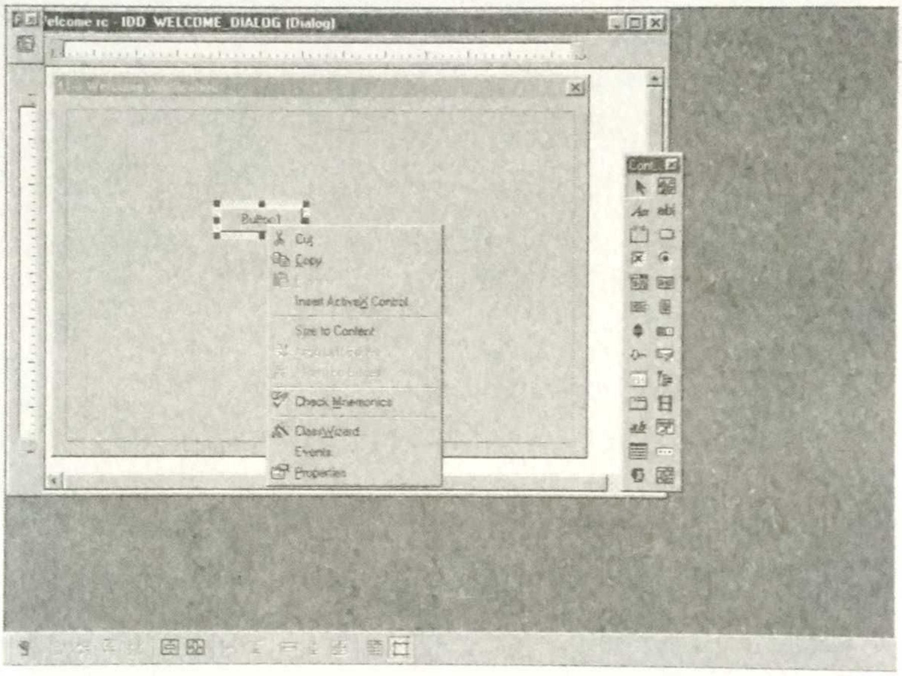
படம் 2.20

இப்போது அந்த இடத்தில் புஷ் பட்டன் கண்ட்ரோல் ஒன்று Button 1 என்ற பெயருடன் (Caption) வெளிப்படுத்தப்படும். Button 1-ஐ க்ளிக் செய்தால் அதைச் சுற்றி ஒரு செவ்வக வட்டம் handleகள் எனப்படும் எட்டு கறுப்பு சின்னங்களுடன் தோன்றும். இந்த ஹேண்டில்களை மவுஸ் மூலம் நகர்த்தலாம். பெரிதுபடுத்தலாம். படம் 2.21.



படம் 2.21

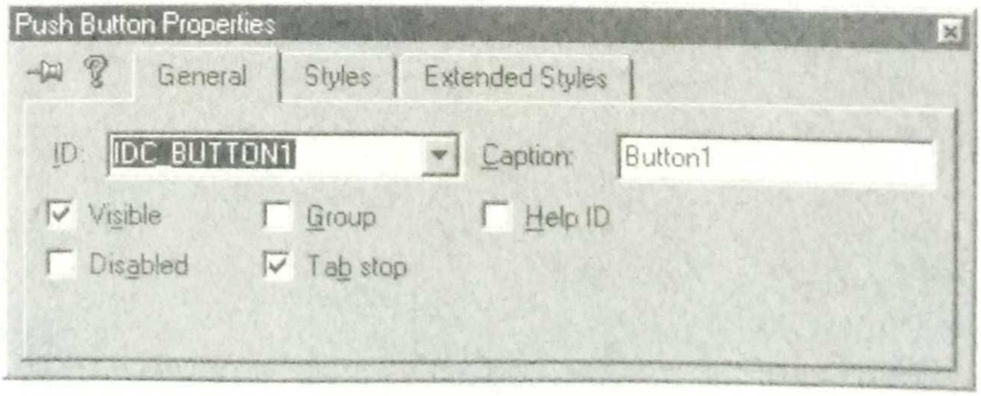
இனி Button 1 மீது மவுஸ் பாயிண்டரை கொண்டு வந்து வலது க்ளிக் செய்யுங்கள். படம் 2.22.



படம் 2.22

இப்போது ஒரு பாப் அப் மெனு வெளிப்படுத்தப்படும். அதிலுள்ள Properties என்பதை க்ளிக் செய்யுங்கள்.

இப்போது Push button Properties எனும் ஒரு டயலாக் பாக்ஸ் வெளிப்படுத்தப்படும். படம் 2.23.



படம் 2.23

Caption இல் Button 1 என்ற வார்த்தை மீது மவுஸ் பாயிண்டரை கொண்டு வந்து க்ளிக் செய்துவிட்டு அந்த இடத்தில் Say Welcome என டைப் செய்யுங்கள்.

அதேபோல் IDC_BUTTON1 என்பதை IDC_SAYWELCOME_BUTTON என்று மாற்றிக்கொள்ளுங்கள். படம் 2.24.