

Resource.h

//{{NO\_DEPENDENCIES}}

// Microsoft Visual C++ generated include file.

// Used by Resource.rc

#define IDR\_MENU1 101

#define IDM\_OPEN 40001

#define IDM\_NEW 40002

#define IDM\_P1 40003

#define IDM\_P2 40004

// 新对象的下一组默认值

//

#ifdef APSTUDIO\_INVOKED

#ifndef APSTUDIO\_READONLY\_SYMBOLS

#define \_APS\_NEXT\_RESOURCE\_VALUE 101

#define \_APS\_NEXT\_COMMAND\_VALUE 40001

#define \_APS\_NEXT\_CONTROL\_VALUE 1001

#define \_APS\_NEXT\_SYMED\_VALUE 101

#endif

#endif

Resource.rc

// Microsoft Visual C++ 生成的资源脚本。

//

#include "resource.h"

#define APSTUDIO\_READONLY\_SYMBOLS

/////////////////////////////////////////////////////////////////////////////

//

// 从 TEXTINCLUDE 2 资源生成。

//

#include "winres.h"

/////////////////////////////////////////////////////////////////////////////

#undef APSTUDIO\_READONLY\_SYMBOLS

/////////////////////////////////////////////////////////////////////////////

// 中文(简体，中国) 资源

#if !defined(AFX\_RESOURCE\_DLL) || defined(AFX\_TARG\_CHS)

LANGUAGE 4, 2

#ifdef APSTUDIO\_INVOKED

/////////////////////////////////////////////////////////////////////////////

//

// TEXTINCLUDE

//

1 TEXTINCLUDE

BEGIN

"resource.h\0"

END

2 TEXTINCLUDE

BEGIN

"#include ""winres.h""\r\n"

"\0"

END

3 TEXTINCLUDE

BEGIN

"\r\n"

"\0"

END

#endif // APSTUDIO\_INVOKED

//

//menu

IDR\_MENU1 MENU

BEGIN

POPUP "FILE"

BEGIN

MENUITEM "OPEN" IDM\_OPEN

MENUITEM "NEW" IDM\_NEW

END

POPUP "项目"

BEGIN

MENUITEM "P1" IDM\_P1

MENUITEM "P2" IDM\_P2

END

END

#endif // 中文(简体，中国) 资源

Main.h

#pragma once

#if !defined WINNIE\_H

#define WINNIE\_H

//------------------------------------

// winnie.h

// (c) Bartosz Milewski, 1995, 97

//------------------------------------

#include <windows.h>

// Forward declaration of our Window Procedure

LRESULT CALLBACK WindowProcedure

(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam);

// We'll be creating windows of this Class in our program

class WinClass

{

public:

WinClass(WNDPROC wndProc, char const \* className, HINSTANCE hInst);

void Register()

{

::RegisterClass(&\_class);

}

private:

WNDCLASS \_class;

};

// Creates a window of a given Class

class WinMaker

{

public:

WinMaker() : \_hwnd(0) {}

WinMaker(char const \* caption, char const \* className, HINSTANCE hInstance);

void Show(int cmdShow)

{

::ShowWindow(\_hwnd, cmdShow);

::UpdateWindow(\_hwnd);

}

protected:

HWND \_hwnd;

};

#endif

Main.cpp

//------------------------------------

// winnie.cpp

// (c) Bartosz Milewski, 1995

//------------------------------------

#include "winnie.h"

#include "resource.h"

// This is the entry point of every Windows program

int WINAPI WinMain

(HINSTANCE hInst, HINSTANCE hPrevInst, char \* cmdParam, int cmdShow)

{

char className[] = "Winnie";

// Define a Window Class and register it under the name "Winnie"

WinClass winClass(WindowProcedure, className, hInst);

winClass.Register();

// Create and show a window

WinMaker win("Hello Windows!", className, hInst);

win.Show(cmdShow);

MSG msg;

int status;

// Keep pumping messages--they end up in our Window Procedure

while ((status = ::GetMessage(&msg, 0, 0, 0)) != 0)

{

if (status == -1)

return -1;

::DispatchMessage(&msg);

}

return msg.wParam;

}

WinClass::WinClass(WNDPROC wndProc, char const \* className, HINSTANCE hInst)

{

\_class.style = 0;

\_class.lpfnWndProc = wndProc; // Window Procedure: mandatory

\_class.cbClsExtra = 0;

\_class.cbWndExtra = 0;

\_class.hInstance = hInst; // owner of the class: mandatory

\_class.hIcon = 0;

\_class.hCursor = ::LoadCursor(0, IDC\_ARROW); // optional

\_class.hbrBackground = (HBRUSH)(COLOR\_WINDOW + 1); // optional

\_class.lpszMenuName = MAKEINTRESOURCE(IDR\_MENU1);

\_class.lpszClassName = className; // mandatory

}

WinMaker::WinMaker

(char const \* caption, char const \* className, HINSTANCE hInstance)

{

\_hwnd = ::CreateWindow(

className, // name of a registered window class

caption, // window caption

WS\_OVERLAPPEDWINDOW, // window style

CW\_USEDEFAULT, // x position

CW\_USEDEFAULT, // y position

CW\_USEDEFAULT, // witdh

CW\_USEDEFAULT, // height

0, // handle to parent window

NULL,//LoadMenu(hInstance,MAKEINTRESOURCE(IDR\_MENU1)), // handle to menu

hInstance, // application instance

0); // window creation data

}

// Window Procedure called by Windows with all kinds of messages

LRESULT CALLBACK WindowProcedure

(HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)

{

auto hmm = GetMenu(hwnd);

auto hmsub1 = GetSubMenu(hmm, 0);

switch (message)

{

case WM\_COMMAND:

switch (LOWORD(wParam))

{

case IDM\_OPEN:

CheckMenuRadioItem(hmm, IDM\_OPEN, IDM\_NEW, IDM\_OPEN, MF\_BYCOMMAND);

MessageBox(hwnd, "open info", "open", MB\_OK);

break;

case IDM\_NEW:

CheckMenuRadioItem(hmm, IDM\_OPEN, IDM\_NEW, IDM\_NEW, MF\_BYCOMMAND);

MessageBox(hwnd, "create new file or not ?", "new file window ", MB\_OKCANCEL);

break;

case IDM\_P1:

MessageBox(hwnd, "program 1", "pro window", MB\_OK);

break;

case IDM\_P2:

MessageBox(hwnd, "program 2", "pro window", MB\_OK);

break;

default:

break;

}

return 0;

// In this simple program, this is the only message we are processing

case WM\_DESTROY:

::PostQuitMessage(0);

return 0; // return zero when processed

}

// All the unprocessed messages go there, to be dealt in some default way

return ::DefWindowProc(hwnd, message, wParam, lParam);

}