#include <windows.h>

#include <stdlib.h>

#include <string.h>

#include <tchar.h>

#define ID\_FILE\_EXIT 40001

static TCHAR szWindowClass[] = \_T("DesktopApp");

static TCHAR szTitle[] = \_T("Windows Desktop Guided Tour Application");

static TCHAR szName[] = \_T("Windows Desktop Program");

HINSTANCE hInst;

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

int CALLBACK WinMain(

\_In\_ HINSTANCE hInstance,

\_In\_ HINSTANCE hPrevInstance,

\_In\_ LPSTR lpCmdLine,

\_In\_ int nCmdShow

)

{

WNDCLASSEX wcex;

wcex.cbSize = sizeof(WNDCLASSEX);

wcex.style = CS\_HREDRAW | CS\_VREDRAW;

wcex.lpfnWndProc = WndProc;

wcex.cbClsExtra = 0;

wcex.cbWndExtra = 0;

wcex.hInstance = hInstance;

wcex.hIcon = LoadIcon(hInstance, IDI\_APPLICATION);

wcex.hCursor = LoadCursor(NULL, IDC\_ARROW);

wcex.hbrBackground = (HBRUSH)(COLOR\_WINDOW + 1);

wcex.lpszMenuName = NULL;

wcex.lpszClassName = szWindowClass;

wcex.hIconSm = LoadIcon(wcex.hInstance, IDI\_APPLICATION);

if (!RegisterClassEx(&wcex))

{

MessageBox(NULL,

\_T("Call to RegisterClassEx failed!"),

\_T("Windows Desktop Guided Tour"),

NULL);

return 1;

}

hInst = hInstance;

HWND hWnd = CreateWindow(

szWindowClass,

szTitle,

WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT, CW\_USEDEFAULT,

500, 100,

NULL,

NULL,

hInstance,

NULL

);

if (!hWnd)

{

MessageBox(NULL,

\_T("Call to CreateWindow failed!"),

\_T("Windows Desktop Guided Tour"),

NULL);

return 1;

}

ShowWindow(hWnd, nCmdShow);

UpdateWindow(hWnd);

MSG msg;

while (GetMessage(&msg, NULL, 0, 0))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return (int)msg.wParam;

}

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

{

PAINTSTRUCT ps;

HDC hdc;

static wchar\_t str[1024];

static INT len = 0;

switch (message)

{

case WM\_PAINT:

{

hdc = BeginPaint(hWnd, &ps);

SetTextColor(hdc, RGB(0, 0, 255));

TextOut(hdc, 0, 0, str, \_tcslen(str));

EndPaint(hWnd, &ps);

return 0;

}

case WM\_CHAR:

{

if (len < 1024) {

str[len++] = wParam;

str[len] = 0;

}

InvalidateRect(hWnd, NULL, 0);

return 0;

}

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

return DefWindowProc(hWnd, message, wParam, lParam);

break;

}

return 0;

}