静态文本

```
#define _CRT_SECURE_NO_WARNINGS
#include <Windows.h>
#include <cstdio>
#include <windowsx.h>
#include "resource.h"

// 自定义窗口过程回调函数
LRESULT CALLBACK MyWindowProc(HWND hwnd, UINT Msg, WPARAM wParam, LPARAM lParam) {

    switch (Msg) {
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    case WM_CREATE: {
        LPCREATESTRUCT pcs = (LPCREATESTRUCT)lParam;
        HWND hBtn = CreateWindowA(TEXT("button"), TEXT("点击我! "), WS_CHILD | WS_VISIBLE | BS_AUTOCHECKBOX,
        30, 30, 80, 30, hwnd, (HMENU)1001, pcs->hInstance, NULL);

        HWND hStatic1 = CreateWindowA(TEXT("static"), TEXT("我是一只猫"), WS_CHILD | WS_VISIBLE | SS_CENTER,
        30, 80, 120, 25, hwnd, (HMENU)2001, pcs->hInstance, NULL);

        HWND hStatic2 = CreateWindowA(TEXT("static"), TEXT("我也一只猫"), WS_CHILD | WS_VISIBLE | SS_CENTER,
        30, 120, 120, 25, hwnd, (HMENU)2002, pcs->hInstance, NULL);

        HWND hStatic3 = CreateWindowA(TEXT("static"), NULL, WS_CHILD | WS_VISIBLE | SS_BITMAP | SS_NOTIFY,
            30, 160, 180, 84, hwnd, (HMENU)2003, pcs->hInstance, NULL);

        HBITMAP hBmp = LoadBitmapA(pcs->hInstance, MAKEINTRESOURCEA(IDB_BITMAP1));

        SendMessageA(hStatic3, STM_SETIMAGE, IMAGE_BITMAP, (LPARAM)hBmp);

    }break;

    case WM_COMMAND: {
        WORD id = LOWORD(wParam);
        WORD code = HIWORD(wParam);
        HWND hCtrl = (HWND)lParam;
        if (id == 1001 && code == BN_CLICKED) {
            //MessageBoxA(NULL, TEXT("按钮被单击"), TEXT("cat"), MB_OK);
            if(Button_GetCheck(hCtrl) == BST_CHECKED)
                MessageBoxA(NULL, TEXT("复选框被选中"), TEXT("cat"), MB_OK);
            else
                MessageBoxA(NULL, TEXT("复选框未选中"), TEXT("cat"), MB_OK);
        }
        else if(id == 2003 && code == STN_CLICKED)
            MessageBoxA(NULL, TEXT("静态文本被点击"), TEXT("cat"), MB_OK);
```

```
    }break;

    case WM_CTLCOLORSTATIC: {
        HDC hdc = (HDC)wParam;
        HWND hStatic = (HWND)lParam;
        UINT CtrlID = GetWindowLongA(hStatic, GWL_ID);
        if (CtrlID == 2001) {
            SetTextColor(hdc, RGB(255, 0, 0));
            SetBkColor(hdc, RGB(0, 255, 0));
            return (LRESULT)GetStockObject(GRAY_BRUSH);
        }
        else if(CtrlID == 2002){
            SetTextColor(hdc, RGB(255, 0, 0));
            SetBkMode(hdc, TRANSPARENT);
            return (LRESULT)CreateSolidBrush(RGB(0, 0, 255));
        }
    }break;
    }

    return DefWindowProcA(hwnd, Msg, wParam, lParam);
}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR
lpCmdLine, int nShowCmd) {

    // 注册窗口类
    WNDCLASS wnd;

    wnd.cbClsExtra = 0;
    wnd.cbWndExtra = 0;

    wnd.hbrBackground = (HBRUSH)(GetStockObject(WHITE_BRUSH));
    wnd.hCursor = LoadCursor(NULL, IDC_ARROW);
    wnd.hIcon = LoadIcon(NULL, IDI_APPLICATION);
    wnd.lpfnWndProc = MyWindowProc;
    wnd.lpszClassName = TEXT("blackcat");
    wnd.lpszMenuName = NULL;
    wnd.style = CS_HREDRAW;
    wnd.hInstance = hInstance;

    RegisterClassA(&wnd);

    // 创建窗口 返回之前发送 WM_CREATE
    HWND hwnd = CreateWindowA(
        TEXT("blackcat"),
        TEXT("黑猫编程"),
        WS_OVERLAPPEDWINDOW,
        100, 100, 300, 300, NULL, NULL, hInstance, NULL
    );

    // 显示窗口
    ShowWindow(hwnd, nShowCmd);

    // 更新窗口
    UpdateWindow(hwnd);

    // 消息循环 收到 WM_QUIT 退出
    MSG msg;
```

```
    while (GetMessageA(&msg, NULL, 0, 0)) {
        TranslateMessage(&msg);
        DispatchMessageA(&msg);
    }

    return 0;
}
```