

# 资源

## 加载资源

资源：代码以外的东西

微软提供了一组 Loadxxx 系列API来支持加载资源文件，其中宏 MAKEINTRESOURCE 是将资源的ID转换为一个字符指针 `#define MAKEINTRESOURCEW(i) ((LPWSTR)((ULONG_PTR)((WORD)(i))))`

## 资源的编译

资源编译器接受一个扩展名为 .RC 的ASCII文本资源文件作为输入。该文件描述了编译到单个数据文件中的所有资源。之后该资源编译器装载所有的资源，并将所有资源放置在一个具有 .RES 扩展名的大数据文件中。这个 .RES 文件包含了在 .RC 文件中定义的诸如图标、光标、位图、声音等所有资源的二进制数据。

可以使用如下方式调用：

```
rc.exe /n /fo Hello.res Hello.rc
```

# 菜单与快捷键

## 菜单

### 在窗口类中指明

WNDCLASSA 结构中的 lpszMenuName 成员指明了要加载的菜单，使用 MAKEINTRESOURCE 将菜单的ID赋值给此结构

```
WNDCLASSA wc;  
wc.lpszMenuName = MAKEINTRESOURCE(菜单ID)
```

### 在创建窗口时指明

CreateWindow 创建窗口时指明参数 hMenu 可以创建带有此菜单的窗口，使用 LoadMenu

```
HMENU hMenu = LoadMenu(hInstance, MAKEINTRESOURCE(菜单ID))
```

## 快捷键

软件快捷键又称键盘加速器，通过 LoadAccelerators 来加载

```
HACCEL haccel = LoadAccelerators(hInstance, MAKEINTRESOURCE(键盘加速器ID));
```

使用键盘加速器时的消息循环

```
while (GetMessage(&msg, NULL, 0, 0)) {  
    if(!TranslateAccelerator(hwnd, haccel, &msg)) { // 消息被派发返回非0，否则0  
        TranslateMessage(&msg);  
        DispatchMessage(&msg);  
    }  
}
```

`TranslateAccelerator` 确定保存在 `msg` 消息结构中的消息是否是键盘消息。如果是，则寻找 `haccel` 句柄匹配的项目，然后调用 `hwnd` 的窗口过程，如果键盘加速器对应了系统菜单的一个菜单项，则相应的消息为 `WM_SYSCOMMAND`，否则是 `WM_COMMAND`

## 对话框和控件

### 模态对话框

模态对话框会阻塞父窗口，其内部使用了：

1. 内建消息循环
2. `EnableWindow` 阻止了父窗口接收消息

### 创建

```
DialogBox(hInstance, MAKEINTRESOURCE(对话框ID), hwnd,DlgProc);
```

`DlgProc` 为模态对话框的对话框过程

```
Bool CALLBACK DlgProc(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)  
{  
    switch(message) {  
        case WM_INITDIALOG:  
            return TRUE;  
        case WM_COMMAND:  
            switch(LOWORD(wParam)) {  
                case IDOK:  
                case IDCANCEL:  
                    EndDialog(hDlg, 0);  
                    return TRUE;  
            }  
            break;  
    }  
    return FALSE;  
}
```

对于处理过的消息，对话框过程返回 `TRUE`，未处理的消息返回 `FALSE` 即可。

### 销毁

对于模态对话框，销毁要使用 `EndDialog`

```
EndDialog(hDlg, 0);
```

## 非模态对话框

非模态对话框不会阻塞父窗口，需要手动显示窗口

### 创建

```
HWND hDlg = CreateDialog(hInstance, MAKEINTRESOURCE(对话框ID), hwndParent, DialogProc);
```

此参数与 `DialogBox` 相同

消息循环需要更改为：

```
while (GetMessage(&msg, NULL, 0, 0)) {  
    // 如果消息是非模态对话框，则发送给对话框过程，并返回TRUE，否则返回FALSE  
    if(hDlg == 0 || !IsDialogMessage(hDlg, &msg)) {  
        TranslateMessage(&msg);  
        DispatchMessage(&msg);  
    }  
}
```

### 销毁

使用 `DestroyWindow` 来进行销毁

```
DestroyWindow(hDlg);
```

## 控件的使用

### WM\_COMMAND消息

消息源	wParam (high word)	wParam (low word)	lParam
菜单	0	菜单ID	0
快捷键	1	快捷键ID	0
控件	控件通知码	控件ID	控件窗口句柄