
第七课 对话框与控件

一、对话框基本概念

1. 对话框的分类

- 1) 模式对话框：当对话框显示时，会禁止本进程内的其它窗口与用户交互。如字体对话框。
- 2) 无模式对话框：当对话框显示时，本进程内的其它窗口仍然可以与用户交互。如查找对话框。

2. 对话框的基本使用

1) 对话框窗口过程函数

```

BOOL CALLBACK DialogProc (
    HWND    hDlg,    // 对话框句柄
    UINT    uMsg,    // 消息标识
    WPARAM wParam,  // 消息参数
    LPARAM lParam   // 消息参数
);

```

返回TRUE：表示自定义窗口过程函数已处理此消息，缺省窗口过程函数无需处理此消息。

返回FALSE：表示自定义窗口过程函数未处理此消息，缺省窗口过程函数应该处理此消息。

普通窗口过程函数：

```

LRESULT CALLBACK WindowProc (HWND hWnd, UINT uMsg,
    WPARAM wParam, LPARAM lParam)
{
    switch (uMsg)
    {
        ...
    }

    return DefWindowProc (hWnd, uMsg, wParam, lParam); // 自定义调用缺省
}

```

对话框窗口过程函数：

```

LRESULT CALLBACK DefDlgProc (HWND hDlg, UINT uMsg,
    WPARAM wParam, LPARAM lParam)
{
    // 函数指针pfuncDialogProc指向自定义的DialogProc函数，缺省调用自定义
    if (pfuncDialogProc (hDlg, uMsg, wParam, lParam))
        return 0;
}

```

```

switch (uMsg)
{
    ...
}
}

```

DefDlgProc: 系统预定义, 与对话框窗口类一起注册到系统中。

DialogProc: 程序员自己定义, 通过函数指针交给系统, 使之可被DefDlgProc调用。

2) 注册窗口类(资源对话框的注册由系统自动完成)

3) 创建对话框

4) 关闭对话框

二、模式对话框

1. 创建

```

int DialogBox (
    HINSTANCE hInstance, // 应用程序实例句柄
    LPCTSTR lpTemplate, // 对话框模板资源名
    HWND hWndParent, // 父窗口句柄
    DLGPROC lpDialogFunc // 对话框窗口过程函数指针
                        // DefDlgProc函数通过此指针调
                        // 用自定义对话框窗口过程函数
);

```

成功返回传给EndDialog函数的nResult参数, 失败返回-1。

1) 需要添加对话框资源。

2) 阻塞函数, 直到对话框关闭该函数才返回, 继续执行其后的代码。

2. 销毁

```

BOOL EndDialog (
    HWND hDlg, // 对话框句柄
    int nResult // DialogBox函数的返回值
);

```

成功返回TRUE, 失败返回FALSE。

模式对话框会用自己的消息循环挂起主消息循环,
只能通过EndDialog函数销毁,
DestroyWindow函数虽然也可以销毁对话框,
但不会恢复主消息循环。

3. 消息

对话框创建之后显示之前没有WM_CREATE消息,
代之以WM_INITDIALOG消息, 处理该消息以实现必要的初始化。

三、无模式对话框

1. 创建

```
HWND CreateDialog (
    HINSTANCE hInstance, // 应用程序实例句柄
    LPCTSTR lpTemplate, // 对话框模板资源名
    HWND hWndParent, // 父窗口句柄
    DLGPROC lpDialogFunc // 对话框窗口过程函数指针
                        // DefDlgProc函数通过此指针调
                        // 用自定义对话框窗口过程函数
);
```

成功返回对话框句柄，失败返回NULL。

非阻塞函数，创建成功即返回对话框句柄，需要调用ShowWindow函数显示对话框。

2. 销毁

```
BOOL DestroyWindow (
    HWND hWnd // 窗口句柄
);
```

成功返回TRUE，失败返回FALSE。

无模式对话框没有自己的消息循环，只是一个普通的子窗口，EndDialog函数只会隐藏该窗口，而非销毁之，收不到WM_DESTROY消息。

3. 消息

对话框创建之后显示之前没有WM_CREATE消息，代之以WM_INITDIALOG消息，处理该消息以实现必要的初始化。

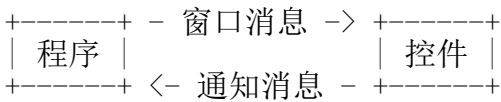
四、对话框与普通窗口比较

	窗口过程函数	创建	销毁	消息
模式对话框	DefDlgProc 调用	DialogBox	EndDialog	WM_INITDIALOG
无模式对话框	DialogProc	CreateDialog		
普通窗口	WindowProc 调用 DefWindowProc	CreateWindow 或 CreateWindowEx	DestroyWindow	WM_CREATE

范例：WinDlg

五、控件基本概念

1. 系统已经定义窗口过程函数并注册其窗口类，如按钮、编辑框等。
2. 无需注册，直接调用CreateWindow/CreateWindowEx函数创建。
3. 作为子窗口的每个控件都有一个唯一标识——控件ID。
4. 程序和控件通过消息交互
 - 1) 窗口消息：程序通过SendMessage函数向控件发送消息，间接调用控件窗口过程函数，设置/获取其状态信息，如禁用按钮、获取编辑框文本等。
 - 2) 通知消息：当控件有事件发生时，向其父窗口发送通知消息，父窗口在其窗口过程函数中处理此消息。



六、静态框

1. 显示文字、图标、位图等

2. 窗口类名：STATIC

3. 风格

```

SS_ICON       - 图标
SS_BITMAP     - 位图
SS_BLACKFRAME - 黑框
SS_BLACKRECT  - 黑块
SS_GRAYFRAME  - 灰框
SS_GRAYRECT   - 灰块
SS_LEFT       - 文字靠左
SS_CENTER     - 文字居中
SS_CENTERIMAGE - 图标/位图居中
SS_NOPREFIX   - 不转义&字符
SS_NOTIFY     - 可发出通知消息

```

4. 创建：CreateWindow/CreateWindowEx

对于带有SS_ICON/SS_BITMAP风格的静态框，可通过CreateWindow/CreateWindowEx函数的lpWindowName参数传入如下形式的图标/位图资源ID：

```

CreateWindow ("STATIC", "#101", ...);
CreateWindowEx (0, "STATIC", "#101", ...);

```

控件ID通过hMenu参数传入。

5. 窗口消息

STM_SETICON - 设置图标句柄

wParam - 图标句柄

lParam - 0

STM_GETICON - 获取图标句柄 (SendMessage的返回值)

wParam - 0

lParam - 0

STM_SETIMAGE - 设置图像句柄

wParam - 图像类型 (IMAGE_ICON/IMAGE_BITMAP/IMAGE_CURSOR)

lParam - 图像句柄

STM_GETIMAGE - 获取图像句柄 (SendMessage的返回值)

wParam - 图像类型

lParam - 0

6. 通知消息

1) 需要在创建时增加SS_NOTIFY风格。

2) 随WM_COMMAND消息发出的通知

WM_COMMAND - 命令通知消息

wParam - LOWORD: 菜单、加速键、控件的ID

HIWORD: 0(菜单)/1(加速键)/通知码(控件)

lParam - NULL(菜单、加速键)/窗口句柄(控件)

通知码:

STN_CLICKED - 单击

STN_DBLCLK - 双击

STN_DISABLE - 禁用

STN_ENABLE - 激活

3) 随WM_CTLCOLORSTATIC消息发出的通知

WM_CTLCOLORSTATIC - 绘制控件背景

wParam - 控件设备上下文句柄

lParam - 控件句柄

范例: WinStatic

七、按钮

1. 风格

1) 下压按钮: BS_PUSHBUTTON、BS_DEFPUSHBUTTON

2) 分组框: BS_GROUPBOX

3) 复选框

两态: BS_CHECKBOX(通过代码设置状态)、
BS_AUTOCHECKBOX(系统自动设置状态)

三态: BS_3STATE(通过代码设置状态)、
BS_AUTO3STATE(系统自动设置状态)

4) 单选框: BS_RADIOBUTTON(通过代码设置状态)、
BS_AUTORADIOBUTTON(系统自动设置状态)

2. 类名

BUTTON

3. 下压按钮

风格: BS_PUSHBUTTON/BS_DEFPUSHBUTTON

单击通知: BN_CLICKED

4. 分组框

1) 风格

BS_GROUPBOX

5. 复选框

1) 风格

BS_CHECKBOX - 点击选择时, 需要自己维护其选择状态
BS_AUTOCHECKBOX - 点击选择时, 系统自动维护其选择状态
BS_3STATE - 点击选择时, 需要自己维护其选择状态
BS_AUTO3STATE - 点击选择时, 系统自动维护其选择状态

2) 窗口消息

BM_SETCHECK - 设置选择状态
wParam - BST_CHECKED(勾)
BST_INDETERMINATE(灰)
BST_UNCHECKED(无)
lParam - 0

BM_GETCHECK - 获取选择状态, SendMessage返回
BST_CHECKED(勾)
BST_INDETERMINATE(灰)
BST_UNCHECKED(无)
wParam - 0
lParam = 0

3) 通知消息

BN_CLICKED - 单击

6. 单选框

1) 风格

BS_RADIOBUTTON - 自己维护选择状态
BS_AUTORADIOBUTTON - 系统维护选择状态

WS_GROUP风格用于为单选框控件分组。
从一个具有WS_GROUP风格的单选框到
下一个具有WS_GROUP风格的单选框之前为一组，
组内互斥，组间独立。

BS_PUSHLIKE风格可用于复选框/单选框，令其具有下压按钮的外观。

范例：WinButton

八、编辑框

1. 分类

单行编辑框：只能编译一行文字。
多行编辑框：可以显示多行文字。
密码编辑框：输入字符回显“*”。

2. 类名

EDIT

3. 风格

ES_AUTOHSCROLL - 水平自动滚动
ES_AUTOVSCROLL - 垂直自动滚动
ES_LEFT - 文字靠左
ES_CENTER - 文字居中
ES_RIGHT - 文字靠右
ES_MULTILINE - 多行
ES_NUMBER - 只能输入数字(0-9)
ES_PASSWORD - 密码
ES_READONLY - 只读

范例：WinEdit

4. 窗口消息

WM_GETTEXTLENGTH - 获取控件中的文本字符数(GetWindowTextLength)
wParam - 0

lParam - 0

WM_GETTEXT - 获取控件中的文本(GetWindowText)
wParam - 缓冲区最大可以容纳的字符数(包括结尾空字符)
lParam - 缓冲区指针

WM_SETTEXT - 设置控件中的文本(SetWindowText)
wParam - 0
lParam - 字符串(空字符结尾)指针

EM_LIMITTEXT - 设置最大字符数
wParam - 最大字符数
lParam - 0

WM_UNDO - 恢复
wParam - 0
lParam - 0

WM_CUT - 剪切
wParam - 0
lParam - 0

WM_COPY - 复制
wParam - 0
lParam - 0

WM_PASTE - 粘贴
wParam - 0
lParam - 0

WM_CLEAR - 删除所选
wParam - 0
lParam - 0

EM_SETSEL - 选取
wParam - 起始字符
lParam - 终止字符

WM_GETFONT - 获取字体, 通过SendMessage返回
wParam - 0
lParam - 0

WM_SETFONT - 设置字体
wParam - 字体句柄
lParam - 是否立即重绘, MAKELPARAM(TRUE/FALSE, 0)

5. 通知消息

EN_ERRSPACE - 分配内存失败
EN_MAXTEXT - 字符数超越上限
EN_UPDATE - 修改文本之后, 更新显示之前
EN_CHANGE - 修改文本并更新显示之后

范例: WinMemo

九、滚动条

1. 分类

风格：SBS_VERT - 垂直滚动条
SBS_HORZ - 水平滚动条

2. 创建

类名：SCROLLBAR

3. 窗口消息

SBM_SETRANGE - 设置滚动范围
wParam - 最小位置
lParam - 最大位置

SBM_GETRANGE - 获取滚动范围
wParam - 最小位置int变量指针
lParam - 最大位置int变量指针

SBM_SETPOS - 设置滚动位置
wParam - 滚动位置
lParam - 是否重绘，TRUE/FALSE

SBM_GETPOS - 获取滚动位置，通过SendMessage返回
wParam - 0
lParam - 0

4. 通知消息

WM_HSCROLL - 水平滚动
WM_VSCROLL - 垂直滚动
wParam - LOWORD滚动码，HIWORD当前位置，
仅对SB_THUMBTRACK/SB_THUMBPOSITION滚动码有效
lParam - 滚动条控件句柄

滚动码：

SB_LINELEFT - 点左箭头
SB_LINERIGHT - 点右箭头
SB_PAGELEFT - 点左空白
SB_PAGERIGHT - 点右空白
SB_LINEUP - 点上箭头
SB_LINEDOWN - 点下箭头
SB_PAGEUP - 点上空白
SB_PAGEDOWN - 点下空白
SB_THUMBTRACK - 拖动滑块
SB_THUMBPOSITION - 释放滑块

范例：WinSB、WinScroll

十、组合框

1. 分类

风格: CBS_SIMPLE - 简单组合框(可编辑)
CBS_DROPDOWN - 下拉组合框(可编辑)
CBS_DROPDOWNLIST - 下拉列表组合框(不可编辑)

2. 创建

类名: COMBOBOX

3. 添加选项

窗口消息:

CB_ADDSTRING - 追加
wParam - 0
lParam - 字符串指针

CB_INSERTSTRING - 插入
wParam - 在此之前插入, 从0开始
lParam - 字符串指针

4. 删除选项

窗口消息:

CB_DELETESTRING - 删除指定选项
wParam - 选项索引
lParam - 0

CB_RESETCONTENT - 删除所有项
wParam - 0
lParam - 0

5. 获取/设置当前选择项

窗口消息:

CB_GETCURSEL - 获取当前选择项,
SendMessage函数返回当前选择项的索引,
无选择项返回CB_ERR (-1)
wParam - 0
lParam - 0

CB_SETCURSEL - 设置当前选择项
wParam - 选项索引
lParam - 0

6. 查找匹配的选项

窗口消息:

CB_FINDSTRING - 根据给定字符串, 查找匹配的选项, 非精确匹配,
SendMessage函数返回匹配选项的索引,
没找到返回CB_ERR
wParam - 查找起点的前一个索引
lParam - 匹配字符串指针

可以不全, 不能写错。

CB_FINDSTRINGEXACT - 精确匹配查找
wParam - 查找起点的前一个索引
lParam - 匹配字符串指针

必须写全。

CB_SELECTSTRING - 非精确匹配查找, 同时选择匹配选项
wParam - 查找起点的前一个索引
lParam - 匹配字符串指针

在CB_FINDSTRING的基础上增加了选择功能。

7. 获取选项内容

窗口消息:

CB_GETLBTEXTLEN - 获取选项字符串长度,
SendMessage函数返回选项字符串的字符数,
不包含结尾空字符,
出错返回CB_ERR
wParam - 选项索引
lParam - 0

CB_GETLBTEXT - 获取选项字符串
wParam - 选项索引
lParam - 缓冲区指针

通过WM_GETTEXTLENGTH/WM_GETTEXT消息,
可以获取SIMPLE和DROPDOWN组合框的编辑框中的内容。

8. 目录的显示

窗口消息:

CB_DIR - 在组合框中显示指定路径下的文件和目录
wParam - 包含那些类型的文件
lParam - 路径字符串指针

wParam可取以下值的位或:

DDL_ARCHIVE - 包含普通文件
DDL_DIRECTORY - 包含目录(带[]显示)
DDL_DRIVES - 包含驱动文件
DDL_HIDDEN - 包含隐藏文件
DDL_SYSTEM - 包含系统文件(.sys)
DDL_READONLY - 包含只读文件
DDL_READWRITE - 包含可读可写文件
DDL_EXCLUSIVE - 排它, 只包含以上属性的文件
无此, 以上属性均为附加包含

9. 设置/获取附加数据

在每个选项中可以保存自定义的数据。

窗口消息:

CB_SETITEMDATA - 设置附加数据
wParam - 选项索引
lParam - 附加数据
CB_GETITEMDATA - 获取附加数据, SendMessage函数返回附加数据
wParam - 选项索引
lParam - 0

10. 通知消息

CBN_SELCHANGE - 当前选择项发生变化
CBN_EDITCHANGE - SIMPLE和DROPDOWN组合框的编辑框中的内容发生变化

范例: WinCombo

十一、列表框

1. 分类

单列列表框
多列列表框

2. 创建

LISTBOX

3. 添加选项

LB_ADDSTRING
LB_INSERTSTRING

4. 删除选项

LB_RESETCONTENT
LB_DELETESTRING

5. 获取/设置当前选择项

LB_GETCURSEL
LB_SETCURSEL

6. 查找匹配的选项

LB_FINDSTRING
LB_FINDSTRINGEXACT
LB_SELECTSTRING

7. 获取选项内容

LB_GETTEXTLEN
LB_GETTEXT

8. 目录的显示

LB_DIR

9. 设置/获取附加数据

LB_SETITEMDATA
LB_GETITEMDATA

10. 通知消息

LBN_SELCHANGE

十二、多文档界面——MDI

1. MDI的组成

MDI主窗口
 MDI客户窗口
 MDI子窗口1
 MDI子窗口2
 ...

2. 创建过程

1) 创建MDI主窗口

A. 没有父窗口，需要注册窗口类。

- B. MDI主窗口的窗口过程函数，不再调用DefWindowProc函数，而是调用DefFrameProc函数。

```
LRESULT DefFrameProc (  
    HWND    hWnd,  
    HWND    hWndMDIClient, // MDI客户窗口句柄  
    UINT    uMsg,  
    WPARAM  wParam,  
    LPARAM  lParam  
);
```

2) 创建MDI客户窗口

- A. 窗口类：MDICLIENT
- B. 父窗口是MDI主窗口。
- C. 创建时要使用CLIENTCREATESTRUCT定义子窗口的起始ID。

3) 创建MDI子窗口

- A. MDI子窗口的父窗口是MDI客户窗口
- B. MDI子窗口类需要注册
- C. MDI子窗口的窗口过程函数调用DefMDIChildProc函数。

范例：WinMDI