

Общие элементы управления

Общие элементы управления

индикатор (Progress Bar)

счетчик (Spin Control)

ползунок (Slider Control, Trackbar)

панель инструментов (Toolbar)

и др.

нужно подключить библиотеку **comctl32.dll**

Создание элемента управления

- подключить заголовочный файл `commctrl.h`
- прописать путь к библиотеке
(`#pragma comment (lib, "comctl32")`)
- создать объект структуры типа `INITCOMMONCONTROLSEX`
- зарегистрировать оконный класс
(вызвать функцию `InitCommonControlsEx`)
- создать нужный элемент

Создание элемента управления (пример)

```
#include "commctrl.h"
```

```
#pragma comment(lib, "comctl32")
```

```
// ...
```

```
INITCOMMONCONTROLSEX cc; // создание объекта структуры
```

```
cc.dwSize = sizeof(cc); // установка размера структуры
```

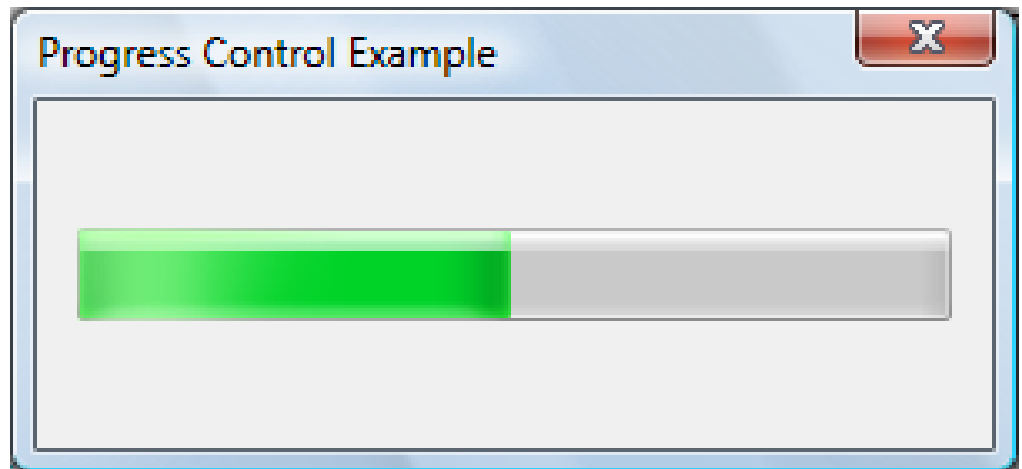
```
cc.dwICC = ICC_WIN95_CLASSES; // установка флагов  
                                (какие элементы загружаем)
```

```
InitCommonControlsEx(&cc); // регистрация класса
```

```
// ...
```

Индикатор (Progress Bar)

- отражает процесс выполнения операции
- относится к классу `PROGRESS_CLASS`
- программно создается с помощью функции `CreateWindow` или `CreateWindowEx`
- получает **сообщения** от родительского окна



Стили (свойства) индикатора

- начинаются с префикса **PBS_**

PBS_VERTICAL (Vertical)	Индикатор приобретает вертикальное положение
PBS_SMOOTH (Smooth)	Индикатор заполняется без фрагментирования (используется сплошное заполнение)

- общие стили, которые поддерживаются

WS_EX_NOPARENTNOTIFY	Отменяет посылку сообщений родительскому окну
WS_TABSTOP (Tabstop)	Позволяет переходить на элемент по нажатию Tab
WS_DISABLED (Disabled)	Делает элемент управления запрещенным
WS_BORDER (Border)	Устанавливает границу вокруг индикатора

Создание индикатора

```
static HWND hProgressBar;  
// ...  
hProgressBar = CreateWindowEx(NULL,  
    PROGRESS_CLASS, // класс  
    NULL,  
    WS_CHILD | WS_VISIBLE |  
    PBS_SMOOTH, //Стили  
    10, 10, 200, 35,  
    hWndParent, // Родительское окно  
    NULL,  
    GetModuleHandle(NULL),  
    NULL);
```

Коды сообщений

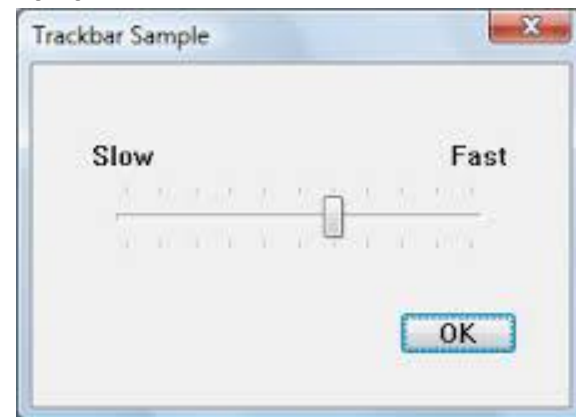
Код сообщения	wParam	lParam	Действие
PBM_SETBKCOLOR	0	(COLORREF) clrBk	Задать цвет фона индикатора
PBM_SETBARCOLOR	0	(COLORREF) clrBar	Задать цвет самого индикатора
PBM_SETRANGE	0	MAKELPARAM (min, max)	Установить диапазон для индикатора
PBM_SETPOS	nNewPos	0	Установить позицию индикатора в nNewPos
PBM_DELTAPOS	nDelta	0	Прибавить к текущей позиции индикатора nDelta
PBM_SETSTEP	nStep	0	Задать шаг индикатора nStep
PBM_STEPIT	0	0	Прибавить к текущей позиции заданный шаг nStep
PBM_GETPOS	0	0	Получить текущую позицию

Обработка сообщений (пример)

```
static HWND hPb;  
// ...  
case WM_INITDIALOG:  
    hPb = CreateWindowEx(NULL, PROGRESS_CLASS, NULL,  
        WS_CHILD | WS_VISIBLE | WS_BORDER | PBS_SMOOTH,  
        30, 100, 280, 25,  
        hwnd, NULL, GetModuleHandle(NULL), NULL);  
    SendMessage(hPb, PBM_SETBKCOLOR, 0, RGB(255, 255, 255));  
    SendMessage(hPb, PBM_SETBARCOLOR, 0, RGB(50, 250, 100));  
    SendMessage(hPb, PBM_SETRANGE, 0, MAKELPARAM(0, 100));  
    SetTimer(hwnd, 1, 1000, 0);  
return true;  
case WM_TIMER:  
    SendMessage(hPb, PBM_DELTAPOS, 1, 0);  
return true;
```

Ползунок (Slider Control)

- позволяет выбирать дискретные значения в заданном диапазоне
- относится к классу `TRACKBAR_CLASS`
- программно создается с помощью функции `CreateWindow` или `CreateWindowEx`
- получает **сообщения** от родительского окна
- **посылает сообщения** родительскому окну (`WM_NOTIFY`)



Стили (свойства) ползунка

- начинаются с префикса **TBS_**

TBS_HORZ (Orientation: Horizontal)	Задаёт горизонтальную ориентацию ползунка
TBS_VERT (Orientation: Vertical)	Задаёт вертикальную ориентацию ползунка
TBS_NOTICKS (Tick Marks: False)	Устанавливает полное отсутствие меток
TBS_AUTOTICKS (Auto Ticks: True)	Устанавливает метки для всех значений в диапазоне; иначе метки лишь в крайних позициях
TBS_BOTH, TBS_LEFT, TBS_RIGHT (Point: Both / Left / Right)	Устанавливает, где будут метки: с двух сторон, сверху / слева, снизу / справа
TBS_TOOLTIPS (Tooltips: True)	Добавляет всплывающие подсказки

Создание ползунка

```
static HWND hTrackBar;  
// ...  
hTrackBar = CreateWindowEx(NULL,  
                           TRACKBAR_CLASS, // класс  
                           NULL,  
                           WS_CHILD | WS_VISIBLE |  
                           TBS_BOTH | TBS_TOOLTIPS |  
                           TBS_AUTOTICKS, //Стили  
                           30, 150, 200, 40,  
                           hWndParent,  
                           (HMENU) IDC_TRACKBAR,  
                           GetModuleHandle(NULL),  
                           NULL);
```

Коды сообщений

Код сообщения	wParam	lParam	Действие
TBM_SETRANGE	bRedraw	MAKELPARAM(min, max)	Задаёт диапазон для ползунка (по умолчанию от 0 до 100; bRedraw – флаг для перерисовки)
TBM_SETPOS	bRedraw	iNewPos	Устанавливает ползунок в позицию iNewPos
TBM_GETPOS	0	0	Возвращает текущую позицию ползунка
TBM_SETLINESIZE	0	iSize	Задаёт шаг iSize при перемещении ползунка с помощью стрелок
TBM_SETPAGESIZE	0	iSize	Задаёт шаг iSize при перемещении с помощью PageUp и PageDown
TBM_SETTICFREQ	iStep	0	Задаёт шаг меток iStep

Сообщения от ползунка

- если родителю посылается сообщение **WM_HSCROLL** или **WM_VSCROLL** :
 - lParam – **дескриптор** ползунка
 - LOWORD(wParam) – **код сообщения**

TB_BOTTOM	Нажата клавиша End (выставлена максимальная позиция)
TB_TOP	Нажата клавиша Home (выставлена мин. позиция)
TB_PAGEUP	Нажата клавиша PageUp или щелчок мышью левее / выше
TB_PAGEDOWN	Нажата клавиша PageDown или щелчок мышью правее / ниже
TB_LINEUP	Нажата стрелка влево / вверх
TB_LINEDOWN	Нажата стрелка вправо / вниз
TB_THUMBTRACK	Ползунок перемещается с помощью мыши

Обработка сообщений (пример)

case WM_INITDIALOG:

```
hTrackBar = CreateWindowEx(NULL, TRACKBAR_CLASS, NULL,  
    WS_CHILD | WS_VISIBLE | TBS_TOOLTIPS | TBS_BOTH |  
    TBS_AUTOTICKS, 30, 150, 200, 40,  
    hwnd, (HMENU)IDC_TB, GetModuleHandle(NULL), NULL);  
SendMessage(hTrackBar, TBM_SETRANGE, true,  
    MAKELPARAM(0, 200));  
SendMessage(hTrackBar, TBM_SETLINE SIZE, 0, 1);  
SendMessage(hTrackBar, TBM_SETTICFREQ, 50, 0);
```

return true;

case WM_HSCROLL:

```
if (SendMessage(hTrackBar, TBM_GETPOS, 0, 0) > 150)  
    { SendMessage(hTrackBar, TBM_SETPOS, true, 0); }
```

return true;

Счетчик

(Spin Control, Up-Down Control)

- позволяет выбирать дискретные значения в заданном диапазоне, инкрементируя или декрементируя их
- относится к классу `UPDOWN_CLASS`
- программно создается с помощью функции `CreateWindow`, `CreateWindowEx` или `CreateUpDownControl`

Стили (свойства) счетчика

- начинаются с префикса **UDS_**

UDS_HORZ (Orientation: Horizontal)	Задает горизонтальную ориентацию счетчика (по умолчанию вертикальная)
UDS_WRAP (Wrap: True)	Устанавливает циклический переход от максимального значения к минимальному
UDS_AUTOBUDDY (Auto Buddy: True)	Назначает «приятелем» ближайший подходящий элемент
UDS_SETBUDDYINT (Set Buddy Integer: True)	Указывает, что значения счетчика и значение в «приятеле» должны изменяться синхронно
UDS_ALIGNLEFT UDS_ALIGNRIGHT (Alignment: Left Right)	Задает положение счетчика слева / справа от «приятеля»
UDS_ARROWKEYS (Arrow Keys: True)	Задает управление счетчиком с помощью стрелок

Создание счетчика

```
static HWND hUpDown;  
// ...  
hUpDown = CreateWindowEx(NULL,  
    UPDOWN_CLASS,  
    NULL,  
    WS_CHILD | WS_VISIBLE |  
    UDS_WRAP | UDS_AUTOBUDDY |  
    UDS_SETBUDDYINT |  
    UDS_ALIGNLEFT | UDS_ARROWKEYS,  
    50, 150, 20, 20,  
    hWndParent,  
    (HMENU) IDC_UPDOWN,  
    GetModuleHandle(NULL),  
    NULL);
```

Коды сообщений

Код сообщения	wParam	lParam	Действие
UDM_SETBUDDY	hwndBuddy	0	Задаёт счётчику «приятеля» с дескриптором hwndBuddy
UDM_SETRANGE32	iMin	iMax	Устанавливает диапазон счётчика (от iMin до iMax)
UDM_SETBASE	iBase	0	Устанавливает основание системы счисления для счётчика
UDM_SETPOS32	0	iNewPos	Устанавливает значение счётчика в позицию iNewPos
UDM_GETPOS32	0	bError	Возвращает текущую позицию счётчика; если bError = true , то значение не получено

Сообщения от счетчика

- при нажатии стрелок родителю посылается сообщение `WM_HSCROLL` или `WM_VSCROLL` :
 - `LOWORD(wParam)` содержит код сообщения `SB_THUMBPOSITION`
- при вводе нового значения в «приятеля» счетчика «приятель» посылает родителю сообщение `WM_COMMAND`:
 - `HIWORD(wParam)` содержит код сообщения `EN_CHANGE`

Обработка сообщений (пример)

```
case WM_INITDIALOG:
```

```
// ...
```

```
SendMessage(hUpDown, UDM_SETBUDDY, (WPARAM)hEdit, 0);
```

```
SendMessage(hUD, UDM_SETRANGE32, 0, 300);
```

```
SendMessage(hUD, UDM_SETPOS32, 0, 0);
```

```
return true;
```

```
case WM_VSCROLL :
```

```
curPos = SendMessage(hUD, UDM_GETPOS32, 0, 0);
```

```
MoveWindow(hwnd, curPos, curPos, rect.right, rect.bottom,  
true);
```

```
return true;
```

```
case WM_COMMAND:
```

```
if (HIWORD(wParam) == EN_CHANGE)
```

```
{ SendMessage(hwnd, WM_VSCROLL, 0, 0); }
```

```
return true;
```

Строка состояния (Status Bar)

- выводит вспомогательную информацию
- относится к классу `STATUSCLASSNAME`
- программно создается с помощью функции `CreateWindow`, `CreateWindowEx` или `CreateStatusWindow` (функция устарела)

```
BOOL CreateStatusWindow(LONG style, // стили
                        LPCTSTR lpszText, // текст в первой секции
                        HWND hHwndParent, // дескриптор родителя
                        UINT uID // идентификатор строки состояния
);
```

Стили строки состояния

SBT_TOOLTIPS	Делаются доступны всплывающие подсказки, если текст не помещается в секции целиком
SBARS_TOOLTIPS	Делаются доступны всплывающие подсказки, если текст не помещается в секции целиком
SBARS_SIZEGRIP	Добавляется специальная область для изменения размера родительского окна

- поддерживаются общие стили (**CCS_BOTTOM** и др.)

Создание строки состояния

```
static HWND hStatusBar;  
// ...  
hStatusBar = CreateWindowEx(NULL,  
    STATUSCLASSNAME, // класс  
    NULL,  
    WS_CHILD | WS_VISIBLE |  
    SBARS_TOOLTIPS | SBARS_SIZEGRIP,  
    CW_USEDEFAULT, CW_USEDEFAULT,  
    CW_USEDEFAULT, CW_USEDEFAULT,  
    hWndParent, // Родительское окно  
    (HMENU) IDC_STATUSBAR,  
    GetModuleHandle(NULL),  
    NULL);
```


Коды сообщений

Код сообщения	wParam	lParam	Действие
SB_SIMPLE	bStyle	0	Задаёт стиль строки состояния: bStyle = true – одна секция, bStyle = false – много секций
SB_SETPARTS	iParts	iArray	Разбивает строку состояния на iParts секций, с правыми координатами из массива iArray
SB_SETTEXT	iIndex	lpszText	Устанавливает текст lpszText в секцию с индексом iIndex
SB_GETTEXT	iIndex	szBuf	Сохраняет текст из секции iIndex в буфер szBuf
SB_GETTEXTLENGTH	iIndex	0	Получает длину текста из секции iIndex
SB_GETRECT	iIndex	rRect	Сохраняет размеры секции iIndex в структуру rRect типа RECT
SB_SETTIPTEXT	iIndex	lpszTooltip	Устанавливает текст lpszTooltip как подсказку к секции iIndex

Обработка сообщений (пример 1)

```
static HWND hStBar;
```

```
// ...
```

```
case WM_INITDIALOG:
```

```
    hStBar = CreateStatusWindow (WS_CHILD | WS_VISIBLE |  
        SBARS_TOOLTIPS | SBARS_SIZEGRIP, NULL, hwnd, 10);
```

```
    SendMessage(hStBar, SB_SIMPLE, true, 0);
```

```
    SendMessage(hStBar, SB_SETBKCOLOR, 0, RGB(0, 200, 0));
```

```
    SendMessage(hStBar, SB_SETTIPTTEXT, 0, (LPARAM)L"Status  
Bar - example");
```

```
return true;
```

```
case WM_RBUTTONDOWN:
```

```
    SendMessage(hStBar, SB_SETTEXT, 255, (LPARAM)  
        L"\t\tWM_RBUTTONDOWN message");
```

```
return true;
```

Обработка сообщений (пример 2)

```
static HWND hStBar;  
int parts[3] = {40, 80, -1};  
// ...  
case WM_INITDIALOG:  
    hStBar = CreateStatusWindow (WS_CHILD | WS_VISIBLE |  
        SBARS_TOOLTIPS | SBARS_SIZEGRIP, NULL, hwnd, 10);  
    SendMessage(hStBar, SB_SIMPLE, false, 0);  
    SendMessage(hStBar, SB_SETPARTS, 3, LPARAM(parts));  
    SendMessage(hStBar, SB_SETTEXT, 0, LPARAM(L"Minsk"));  
    SendMessage(hStBar, SB_SETTEXT, 1, LPARAM(L"09:00"));  
    return true;  
case WM_RBUTTONDOWN:  
    SendMessage(hStBar, SB_SETTEXT, 0, (LPARAM)  
        L"\t\tWM_RBUTTONDOWN message");  
    return true;
```

Вопросы?