

WinAPI - Anatomia okna

Tomasz Dzieciak

Torun, 20/01/2012



Nazwa klasy okna

- identyfikuje okno aplikacji
- opisuje rolę aplikacji (informacja)

Procedura zdarzeniowa

- wskazuje procedurę, która zajmie się obsługą zdarzeń w oknie aplikacji

Instancja aplikacji

- identyfikuje instancję klasy, która zarejestrowała okno programu
- dzięki temu nie musimy dbać o ręczne wyrejestrowanie instancji
(UnregisterClassEx())







Styl klasy okna

- kombinacja flag bitowych określająca styl okna aplikacji
- CS_DBLCLKS - obsługuje podwójne kliknięcia
- CS_OWNDC - posiada własny kontekst urządzenia

Styl klasy okna

- `CS_CLASSDC` - wszystkie okna współdzielą kontekst urządzenia
- `CS_(H/V)REDRAW` - wymuszenie odrysowania okna po skalowaniu
- `CS_SAVEBITS` - zachowuje wizualną zawartość okna (!!!)

Ikony

- IDI_APPLICATION - 
- IDI_ASTERISK - 
- IDI_ERROR - 
- IDI_WARNING - 
- IDI_QUESTION - 
- IDI_WINLOGO - 

Kursory

– IDC_ARROW -



– IDC_WAIT -



– IDC_APPSTARTING -








– IDC_HELP -







– IDC_CROSS -



Kursory

- IDC_NO - 
- IDC_IBEAM - 
- IDC_HAND - 
- IDC_UPARROW - 
- IDC_SIZEALL - 

Kursor

- IDC_SIZENESW - 
- IDC_SIZENS - 
- IDC_SIZENWSE - 
- IDC_SIZEWE - 

Można wczytać własny - LoadImage(...)

Tło okna

W rzeczywistości to co nazywamy tłem okna jest **obszarem klienta!**

Styl okna

grupa	flagi	nazwa stylu	uwagi
rodzaj okna	WS_CHILD WS_CHILDWINDOW	okno potomne	Jedyną cechą wyróżniającą okno potomne od zwykłego jest brak możliwości posiadania przez nie menu.
	WS_OVERLAPPED WS_TILED	okno trwałe	Trwałe okno to takie, które może być wyświetlane jako niemodalne.
	WS_POPUP	okno „wyskakujące”	Takie okno może być wyświetlane tylko jako modalne.
krawędź okna	WS_BORDER	stałe obramowanie	Krawędź okna z takim stylem jest narysowana cienką linią i nie reaguje na przeciąganie (nie można więc zmienić rozmiaru okna, ciągnąc za jego brzeg).
	WS_THICKFRAME WS_SIZEBOX	zmiennie obramowanie	Zmienne obramowanie okna jest narysowane grubą linią i pozwala na zmianę rozmiaru okna poprzez przeciąganie krawędzi.
	WS_DLGRAME	obramowanie okna dialogowego	Ustala obramowanie typowe dla okien dialogowych; okno z tym stylem nie może mieć paska tytułu (stylu WS_CAPTION).
pasek tytułu	WS_CAPTION	pasek tytułu	Okno z tym stylem posiada widoczny pasek tytułu.
	WS_SYSMENU	menu sterujące	Styl ten powoduje obecność ikonki w lewym górnym rogu okna. Kliknięcie na tą ikonkę powoduje pokazanie menu sterującego. By styl ten zadziałał, należy podać także flagę WS_CAPTION.
	WS_MAXIMIZEBOX	przycisk maksymalizacji /przywracania	Tworzy okno z obecnym przyciskiem maksymalizacji. Flaga WS_SYSMENU jest również wymagana.
	WS_MINIMIZEBOX	przycisk minimalizacji	Tworzy okno z obecnym przyciskiem minimalizacji. Flaga WS_SYSMENU jest również wymagana.
początkowy stan okna	WS_VISIBLE	widoczne okno	Okno z tym stylem jest widoczne na ekranie od razu po utworzeniu (nie trzeba stosować dlań funkcji ShowWindow()).
	WS_MAXIMIZE	okno zmaksymalizowane	Po utworzeniu okno jest widoczne i zmaksymalizowane.
	WS_ICONIC WS_MINIMIZE	okno zminimalizowane	Okno z tym stylem jest z początku zredukowane do przycisku na pasku zadań.
	WS_DISABLED	nieaktywne okno	Styl ten dezaktywuje okno - tak, że nie reaguje ono na działania użytkownika. Okno można uaktywnić z powrotem za pomocą funkcji EnableWindow().
paski przewijania	WS_HSCROLL	poziomy pasek przewijania	Tworzy okno z poziomym paskiem przewijania.
	WS_VSCROLL	pionowy pasek przewijania	Tworzy okno z pionowym paskiem przewijania.

Tabela 30. Flagi bitowe stylu okna

Styl okna

chcąc wyłączyć jedną z niechcianych flag w przypadku użycia predefiniowanego stylu wystarczy zastosować sztuczkę:

```
WS_STYL_PREDEF & ~WS_FLAGA
```

Pokazywanie / ukrywanie okna

```
BOOL ShowWindow(HWND hWnd, int  
nCmdShow);
```

- SW_SHOW - pokazuje okno
- SW_MAXIMIZE - zmaksymalizowane
- SW_HIDE - ukrywa okno

Sprawdzanie widoczności okna

- sprawdza czy jest widoczne
BOOL IsWindowVisible(HWND hWnd);
- sprawdza, czy jest zminimalizowane
BOOL IsIconic(HWND hWnd);
- sprawdza, czy jest powiększone
BOOL IsZoomed(HWND hWnd);

Ustawianie tytułu okna

Jedna z podstawowych operacji na klasie okna. Realizuje się poprzez:

```
BOOL SetWindowText(HWND hWnd,  
LPCTSTR lpString);
```

Odczytanie tytułu okna

```
char szTytułOkna[256];  
GetWindowText (hOkno, szTytułOkna,  
256);
```

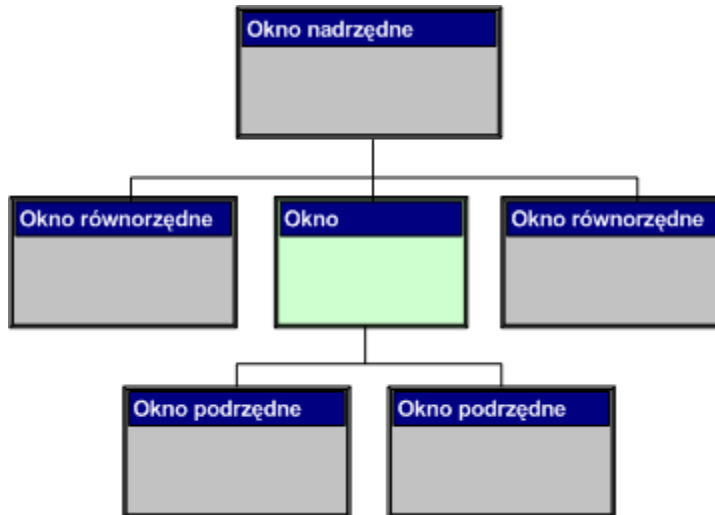
Inne dane...

```
BOOL GetWindowInfo(HWND hWnd,
PWINDOWINFO pwi);
```

<i>typ</i>	<i>pola</i>	<i>opis</i>
DWORD	cbSize	Podobnie jak w WNDCLASSEX, pierwszy pole struktury WINDOWINFO określa jej rozmiar w bajtach . Musimy ustawić je na <code>sizeof(WINDOWINFO)</code> , zanim zechcemy wywołać funkcję <code>GetWindowInfo()</code> .
RECT	rcWindow rcClient	Te dwa pola określają prostokąty zawierające (kolejno): całe okno oraz jego obszar klienta .
DWORD	dwStyle dwExStyle	Z tych pól możemy odczytać styl oraz rozszerzony styl okna, czyli wartości, które zostały ongiś przekazane do <code>CreateWindow[Ex]()</code> podczas tworzenia okna.
DWORD	dwWindowStatus	Pole to określa status okna , tzn. to, czy jest ono aktywne, czy też nie. Wartość stałej <code>WS_ACTIVECAPTION</code> w tym polu oznacza pierwszą sytuację, zero - drugą.
UINT	cxWindowBorders cyWindowBorders	Ta para pól zawiera szerokość oraz wysokość obrzeża okna w pikselach.
ATOM	atomWindowType	W tym polu zapisany zostaje atom identyfikujący klasę okna . Jak (mam nadzieję) pamiętasz, atom ten zwraca funkcja <code>RegisterClass[Ex]()</code> po rejestracji klasy okna, a wartość ta może zostać użyta w miejsce nazwy tejże klasy w niektórych funkcjach, jak np. <code>CreateWindow[Ex]()</code> .
WORD	wCreatorVersion	Określa windowsową wersję aplikacji, która stworzyła okna.

Tabela 35. Pola struktury WINDOWINFO

Uchwyty!



- HWND GetParent(HWND hWnd);
- HWND GetWindow(HWND hWnd, UINT uCmd);
- HWND FindWindow(LPCTSTR lpClassName, LPCTSTR lpWindowName);

C.D.N...