# Powłoka Windows (Windows Shell)

- Hierarchiczna przestrzeń nazw obiektów wykonujących operacje na plikach i folderach
- Wybrane możliwości:
  - uruchamianie aplikacji
  - 🗆 drag & drop
  - definicja rozszerzenia plików, powiązanie plików z aplikacją
  - modyfikacja ikon plików
  - modyfikacja wyglądu folderu
  - □ automatyczne uruchamianie aplikacji z CD-ROM
  - tworzenie skrótów
  - użycie paska zadań (taskbar) i pasków narzędzi na pulpicie
  - użycie Active Desktop
  - aplikacje panelu sterowania

## Wersje bibliotek

- Wykorzystywane biblioteki: Comctl32.dll, Shell32.dll, Shlwapi.dll
  - różna zawartość w zależności od wersji
  - zawarte w instalacjach Windows i Internet Explorer'a
  - wersję każdej biblioteki trzeba sprawdzać oddzielnie

```
hinstDll = LoadLibrary(lpszDllName);
if (hinstDll) {
    DLLGETVERSIONPROC pGV = NULL;
    pGV = (DLLGETVERSIONPROC)
        GetProcAddress(hinstDll,"DllGetVersion");
    if (pGV) {
        DLLVERSIONINFO dvi;
        ZeroMemory(&dvi, sizeof(dvi));
        (*pGV)(&dvi);
    }
    FreeLibrary(hinstDll);
}
```

# Przestrzeń nazw powłoki

- Organizacja wszystkich obiektów zarządzanych przez powłokę w jedną hierarchię drzewiastą
  - m.in.: pliki i foldery, drukarki, komputery połączone siecią, aplikacje panelu sterowania, śmietniczka
  - elementy hierarchii:
    - pulpit pień
    - foldery węzły drzewa (foldery systemowe lub wirtualne)
    - pliki liście drzewa (pliki dyskowe)
- Identyfikacja obiektów
  - pełna ścieżka do pliku brak dla obiektów wirtualnych
  - identyfikator (item ID) struktura SHITEMID
  - lista identyfikatorów struktura ITEMIDLIST
  - **PIDL wskaźnik na ITEMIDLIST** 
    - względne i bezwzględne (rozpoczynające się od pulpitu)

Krzysztof Mossakowski Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej

### Foldery

- Interfejs IMalloc alokacja pamięci dla PIDL
  - SHGetMalloc()
  - IMalloc::Alloc(), IMalloc::Free()
- Foldery specjalne identyfikowane stałymi CSIDL
- Interfejs IShellFolder informacje o folderze
   zawartość, nazwa, właściwości
- Funkcje:
  - □ SHBrowseForFolder() okno dialogowe z wyborem foldera
  - SHGetPathFromIDList() pobranie ścieżki dla PIDL
  - SHGetFolderLocation() PIDL specjalnego foldera
  - SHGetFolderPath() pobranie ścieżki specjalnego foldera
  - SHGetFileInfo() informacje o pliku podanym przez PIDL
  - SHBindToParent() pobranie PIDL rodzica obiektu

Krzysztof Mossakowski Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej

#### Przykład - SHBrowseForFolder()

```
LPITEMIDLIST pidlRoot = NULL, pidl = NULL;
BROWSEINFO bi = \{0\};
LPMALLOC pMalloc = NULL;
SHGetMalloc(&pMalloc);
SHGetFolderLocation(hWnd, CSIDL PROGRAM FILES,
                    NULL, NULL, &pidlRoot);
bi.hwndOwner = hWnd;
bi.pidlRoot = pidlRoot;
bi.pszDisplayName = buf;
bi.lpszTitle = "Choose a folder";
if ((pidl = SHBrowseForFolder(&bi)) != NULL) {
    SHGetPathFromIDList(pidl, buf);
    pMalloc->Free(pidl);
pMalloc->Free(pidlRoot);
pMalloc->Release();
```

# **Uruchamianie aplikacji**

- ShellExecute(), ShellExecuteEx()
  - plik lub folder do uruchomienia
  - operacja (polecenie)
- Operacje
  - najczęściej wykorzystywane:
    - edit uruchomienie edytora i otwarcie dokumentu
    - find szukanie począwszy od wskazanego katalogu
    - open uruchomienie aplikacji
    - print wydrukowanie dokumentu
    - properties wyświetlenie właściwości
  - zdefiniowane dla obiektu w rejestrze HKEY\_CLASSES\_ROOT\CLSID\{object\_clsid}\Shell\operacja

ShellExecute(hwnd, "find", "C:\\Program Files", NULL, NULL, 0);

# System plików

- Cechy specjalne:
  - połączenia między dokumentami (np. plikami używanymi ze strony HTML)
  - ograniczenia dostępu pomiędzy użytkownikami
  - lista ostatnio używanych dokumentów
- Operacje na plikach SHFileOperation()
  - □ kopiowanie, przenoszenie, usuwanie, zmiana nazwy
- Powiadomienie o zmianach SHChangeNotify()
- Dodanie dokumentu do listy ostatnio używanych -SHAddToRecentDocs ()
- Dostęp do katalogów użytkownika: CSIDL\_MYDOCUMENTS

SHGetFolderLocation(NULL, CSIDL MYDOCUMENTS, NULL, 0, &pidlDocFiles);

### **Przenoszenie danych**

- Przenoszone dane
   obiekty implementujące interfejs IDataObject
- Schowek
  - umieszczenie danych OleSetClipboard() (dane jako obiekt implementujący IDataObject)
  - pobranie danych ze schowka OleGetClipboard()
  - dostęp do danych IDataObject::GetData()
- Drag & drop
  - inicjalizacja OleInitialize()
  - źródło przenoszonych danych IDropSource
  - cel przenoszonych danych IDropTarget

```
HKEY CLASSES ROOT
  .ZID
     Default)=WinZip
    ContentType=application/x-zip-compressed
PersistentHandler
       (Default)={098f2470-bae0-11cd-b579-08002b30bfeb}
    ShèllEx
       {00021500-0000-0000-c000-00000000046}=
          {E0D79307-84BE-11CE-9641-444553540000}
    ShellNew
       NullFile
  CLSID
    {E0D79306-84BE-11CE-9641-444553540000}
       (Default)=WinZip
InProcServer32
          (Default)=C:\PROGRA~1\WinZip\WZSHLSTB.DLL
          ThreadingModel=Apartment
  WinZip
    DefaultIcon
       (Default)=C:\PROGRA~1\WinZip\winzip32.exe,0
    shell
       open
          [Default]=Open with &WinZip
          command
             (Default)=C:\PROGRA~1\WinZip\winzip32.exe "%1"
    shellex
       DropHandler
          (Default)={E0D79306-84BE-11CE-9641-444553540000}
```

## Menu kontekstowe

- Polecenia
  - □ standardowe:

open

print

printto - polecenie niewyświetlane w menu kontekstowym

explore - otwiera Windows Explorer

find - otwiera Windows Search

openas - otwiera Open With

properties

własne

```
HKEY_CLASSES_ROOT

.myp

(Default)=MyProgram.1

MyProgram.1

(Default)=My Program Application

Shell

(Default)=doit

doit

(Default)=&Do it

command

(Default)=C:\MP\MP.exe /d "%1"
```

Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej

# Skróty

- Obiekty, do których mogą istnieć skróty: pliki, foldery, dyski, drukarki
- Obsługa programistyczna interfejs IShellLink
  - umiejscowienie SetPath(), GetPath()
  - L katalog roboczy SetWorkingDirectory(), GetWorkingDirectory()
  - parametry SetArguments(), GetArguments()
  - parametr funkcji ShowWindow SetShowCmd(), GetShowCmd()
  - ikona SetIconLocation(), GetIconLocation()
  - Opis SetDescription(), GetDescription()
  - hot key SetHotKey(), GetHotKey()
  - identyfikator SetIDList() , GetIDList()

### **Tworzenie skrótu**

Krz

Wν

```
HRESULT CreateLink (LPCSTR lpszPathObj,
     LPCSTR lpszPathLink, LPCSTR lpszDesc) {
  HRESULT hres; IShellLink* psl;
  hres = CoCreateInstance(CLSID ShellLink,
             NULL, CLSCTX INPRO\overline{C} SERVER,
             IID IShellLink, (LPVOID*)&psl);
  if (SUCCEEDED(hres)) {
    IPersistFile* ppf;
    psl->SetPath(lpszPathObj);
    psl->SetDescription(lpszDesc);
    hres = psl->QueryInterface(
             &IID IPersistFile, (LPVOID*)&ppf);
    if (SUCCEEDED(hres))
      WCHAR wsz[MAX PATH];
      MultiByteToWideChar(CP ACP, 0,
              lpszPathLink, -1, wsz, MAX PATH);
      hres = ppf->Save(wsz, TRUE);
      ppf->Release();
    psl->Release();
  return hres;
```

# Pasek zadań *(taskbar)*

- Menu start (start menu)
  - dodawanie przed NT 4.0 i 95 wykorzystać DDE
  - dodawanie od NT 4.0 i 95 wykorzystać IShellLink i SHGetSpecialFolderLocation() Z CSIDL\_PROGRAMS
- Quick Launch, Address, Links, Desktop toolbars
   możliwość tworzenia własnych pasków
- Taskbar buttons
  - □ okna bez rodzica ze stylem **ws\_ex\_appwindow**
  - FlashWindow() "zamruganie"
- Status area (taskbar notification area, tray)
  - ikony symbolizujące stan lub zdarzenie
  - Shell\_NotifyIcon() dodawanie, modyfikacja, usuwanie
- powiadomienia o akcjach myszy nad ikoną (lewy, prawy przycisk, dwuklik, ruch) ysztof Mossakowski

Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej

### **Application Desktop Toolbar**

• Cechy:

możliwość doczepienia do brzegu ekranu

- zwykle zawiera przyciski uruchamiające akcje
- poza obszarem dostępnym dla okien aplikacji

Obsługa:

```
SHAppBarMessage()
```

ABM\_NEW, ABM\_REMOVE, ABM\_SETPOS, ABM\_SETAUTOHIDEBAR, ABM\_ACTIVATE ABM\_QUERYPOS, ABM\_WINDOWPOSCHANGED

powiadomienia ABN\_POSCHANGED, ABN\_STATECHANGED, ABN\_FULLSCREENAPP, ABN\_WINDOWARRANGE

### Automatyczne uzupełnianie

- Automatyczne podpowiadanie zakończenia tekstu wpisywanego w kontrolce edycyjnej (edit control, ComboBoxEx)
- SHAutoComplete() wykorzystanie mechanizmu dla podanej kontrolki edycyjnej
- Ręczne wykorzystanie interfejs IAutoComplete
  - możliwość korzystania z jednego lub z wielu źródeł podpowiedzi: historia, lista ostatnio używanych, przestrzeń nazw powłoki
  - dwa tryby pracy
    - autoappend dopisanie podpowiedzi do tekstu jako zaznaczonego fragmentu
    - autosuggest rozwijalna lista z podpowiedziami

# Własne paski

- Explorer Bar
  - dokowalny pasek w przeglądarce
  - dopisany do menu View / Explorer Bar
  - do stworzenia konieczne jest zaimplementowanie i zarejestrowanie obiektu *band*
- Tool Band
  - kontrolka rebar zawierająca kilka kontrolek toolbar
  - wykorzystywana w Internet Explorer'ze
- Desk Band
  - dokowalne okna na pulpicie (dostępne z menu kontekstowego dla paska zadań)
- Obsługa programistyczna IDeskBand
  - wpis w rejestrze HKEY\_LOCAL\_MACHINE \ Software \ Microsoft \ Internet Explorer \ Toolbar

Krzysztof Mossakowski

Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej

### Active Desktop

- Umieszczenie dokumentów HTML i ich zawartości (m.in. kontrolki ActiveX, aplety Javy, animacje Flash) na pulpicie
- Obsługa programistyczna interfejs IActiveDesktop
  - tworzenie CoCreateInstance() dla CLSID\_IActiveDesktop
  - dodawanie elementu AddDesktopItem(),
    AddDesktopItemWithUI(), AddUrl()
  - wyliczenie elementów GetDesktopItemCount(),
    GetDesktopItem()
  - □ tapeta GetWallpaper(), SetWallpaper(), GetWallpaperOptions(), SetWallpaperOptions()

Krz

#### Active Desktop – przykład w C++

```
#include "wininet.h"
#include "shlobj.h"
CoInitialize(NULL);
HRESULT hr;
IActiveDesktop *pActiveDesktop;
hr = CoCreateInstance(CLSID ActiveDesktop,
           NULL, CLSCTX INPROC SERVER,
           IID IActiveDesktop,
           (void**) &pActiveDesktop);
pActiveDesktop->SetWallpaper(
           L"C:\\MyWallPaper.gif", 0);
pActiveDesktop->ApplyChanges(AD APPLY ALL);
pActiveDesktop->Release();
```

### Active Desktop – przykład w C#

```
IActiveDesktop ad = (IActiveDesktop)new ActiveDesktop();
ad.SetWallpaper(@"C:\MyWallpaper.gif", 0);
ad.ApplyChanges(AD APPLY.ALL);
```

### Windows Themes

- Wprowadzone w Microsoft Plus! dla Windows 95
- Określone w plikach .themes
  - prosty format tekstowy, podobny do plików .ini
  - mogą być tworzone i rozpowszechniane niezależnie od systemu
  - widoczne w oknie ustawień wyświetlania
- Mogą zmodyfikować tapetę, kursory, fonty, dźwięki i ikony

# Style wizualne (Visual Styles)

- Wprowadzone w Windows XP
- Definiują wygląd kontrolek
- UxTheme API umożliwia użycie stylu w aplikacji w przypadku potrzeby ręcznego rysowania
- Stosowane są do:
  - obszarów non-client okien
  - common controls





### ComCtl32.dll i UxTheme.lib

#### ComCtl32.dll

- włączone do Windows XP w wersji 5 (dla kompatybilności) i 6 (dla nowych stylów)
- w wersji 6 dołączone są nowe kontrolki
- do zastosowania stylów w aplikacji konieczne jest użycie manifestu

#### UxTheme.lib

- biblioteka rysująca kontrolki, pozwalająca na oddzielenie elementów wizualnych od funkcjonalności kontrolek
- używana przez common controls
- wymagana przez ComCtl32.dll

## Manifest

 Plik XML dodawany jako zasób lub jako zewnętrzny plik w katalogu aplikacji

MANIFEST ID RT MANIFEST "YourApp.exe.manifest"

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <assembly xmlns="urn:schemas-microsoft-com:asm.v1"</pre>
      manifestVersion="1.0">
  <assemblyIdentity version="1.0.0.0"</pre>
      processorArchitecture="X86"
      name="CompanyName.ProductName.YourApp"
      type="win32" />
  <description>Your application description here.</description>
  <dependencv>
    <dependentAssembly>
      <assemblyIdentity type="win32"</pre>
          name="Microsoft.Windows.Common-Controls"
          version="6.0.0.0"
          processorArchitecture="X86"
          publicKeyToken="6595b64144ccf1df"
          language="*" />
    </dependentAssembly>
  </dependency>
</assembly>
```

# Wskazówki dla stylów wizualnych

- Pliki manifestów są ignorowane przez poprzednie wersje Windows
- Nie linkować statycznie biblioteki UxTheme.lib, lecz wczytywać ją dynamicznie
- Obsługiwać błędy wynikające z innego niż oczekiwane działania stylów
- Przetestować swoją aplikacji, by mieć pewność, że nie wymaga ComCtl32.dll w wersji 6
  - jeśli aplikacja wykorzystuje specyficzne cechy tej wersji, należy napisać alternatywny kod dla wcześniejszych systemów
  - nie można zainstalować ComCtl32.dll w wersji 6 w poprzednich wersjach Windows

## Użycie UxTheme API

```
HTHEME hTheme = OpenThemeData(hwndButton, "Button");
DrawMyControl(hDC, hwndButton, hTheme, iState);
if (hTheme) {
   CloseTheme(hTheme);
void DrawMyControl(HDC hDC, HWND hwndButton, HTHEME hTheme,
   int iState)
{
  if (hTheme) {
    hr = DrawThemeBackground(hTheme, hDC, BP BUTTON,
             iState, &rc, 0);
    hr = GetThemeBackgroundContentRect(hTheme, BP BUTTON,
             iState, &rc, &rcContent);
    hr = DrawThemeText(hTheme, hDC, BP BUTTON,
             iState, szButtonText, cch, DT CENTER
             DT VCENTER | DT SINGLELINE, 0, &rcContent);
  } else {
    // Rysowac bez uzycia stylow wizualnych
```

### **Desktop Window Manager**



- Desktop Window Manager (DWM) jest nowym komponentem wprowadzonym w Windows Vista
- DWM pozwala na uzyskania charakterystycznych dla Windows Vista efektów graficznych na oknach, m.in.:
  - □ glass window frames
  - 3-D window transition animations
  - Windows Flip i Windows Flip3D
  - obsługa wysokich rozdzielczości

http://weblogs.asp.net/kennykerr/archive/2006/08/10/Windows-Vista-for-Developers- 1320 -Part-3- 1320 -The-Desktop-Window-Manager.aspx

Can you see me?

Sample

Krzysztof Mossakowski

🗅 🌈 📕 🧐 (\*

Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej

http://www.mini.pw.edu.pl/~mossakow

x

[Vista]

### **Style wizualne w Windows Forms 2.0**

- Włączenie użycia stylów wizualnych
   Application.EnableVisualStyles()
- Kontrolki rysowane bez użycia stylów wizualnych: Label, LinkLabel, DomainUpDown, NumericUpDown, CheckedListBox
- Znane problemy z kontrolkami:
  - MonthCalendar zaznaczenie zakresu dat
  - TabControl zakładki rysowane na dole lub z boku
  - TextBox surrogate fonts (pary znaków Unicode)

# Użycie stylów wizualnych [V



- Klasa ControlPaint rysowanie elementów standardowych kontrolek systemowych
- Uniwersalne klasy, których można użyć niezależnie od ustawień systemowych:
  - ButtonRenderer, CheckBoxRenderer, GroupBoxRenderer, RadioButtonRenderer
- Klasy, których można użyć wyłącznie przy włączonej obsłudze stylów wizualnych w systemie:
  - ComboBoxRenderer, ProgressBarRenderer, ScrollBarRenderer, TabRenderer, TextBoxRenderer, TrackBarRenderer

## Przykład stylów wizualnych [WinForms]

### **Rysowanie elementów**

# [WinForms]

```
private VisualStyleRenderer renderer = null;
private readonly VisualStyleElement element =
    VisualStyleElement.StartPanel.LogOffButtons.Normal;
public CustomControl()
  if (Application.RenderWithVisualStyles &&
      VisualStyleRenderer.IsElementDefined(element)) {
    renderer = new VisualStyleRenderer(element);
protected override void OnPaint(PaintEventArgs e)
  if (renderer != null) {
    renderer.DrawBackground(e.Graphics,
                            this.ClientRectangle);
  } else {
    this.Text = "Visual styles are disabled.";
    TextRenderer.DrawText(e.Graphics, this.Text,
         this.Font, new Point(0, 0), this.ForeColor);
```

[WPF]

# Użycie wybranej kompozycji

- 1. Dodać referencję do odpowiedniego assembly :
  - PresentationFramework.Aero, PresentationFramework.Classic, PresentationFramework.Luna, PresentationFramework.Royale

#### 2. Dodać **ResourceDictionary** do zasobów aplikacji:

<Application.Resources> <ResourceDictionary> <ResourceDictionary.MergedDictionaries> <!--<ResourceDictionary Source="/PresentationFramework.Royale;component/themes/Royale.NormalColor.xaml" />--> <!--<ResourceDictionary Source="/PresentationFramework.Luna;component/themes/Luna.NormalColor.xaml" />--> <!--<ResourceDictionary Source="/PresentationFramework.Luna;component/themes/Luna.NormalColor.xaml" />-->

<!--<ResourceDictionary Source="/PresentationFramework.Luna;component/themes/Luna.Metallic.xaml" />-->

<ResourceDictionary Source="/PresentationFramework.Aero;component/themes/Aero.NormalColor.xaml" />

<!--<ResourceDictionary Source="/PresentationFramework.Classic;component/themes/Classic.xaml" />--> </ResourceDictionary.MergedDictionaries> </ResourceDictionary> </Application.Resources>

Krzysztof Mossakowski Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej

### Kompozycje Windows dostępne w WPF



# Kompozycje WPF

#### • Open source:

- http://wpf.codeplex.com/Wiki/View.aspx?title=WPF%20Themes
- □ <u>http://www.codeplex.com/wpfthemes</u>





#### Komercyjne:

- □ <u>http://www.xamltemplates.net</u>
- □ <u>http://www.nukeation.com/reuxables.aspx?id=catalog</u>

#### Krzysztof Mossakowski

Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej