

Jak na průzkumníka Windows



V této kapitole si ukážeme, jak lze v Delphi realizovat některé funkce, které používá Průzkumník Windows. Dozvíme se, jak zaregistrovat vlastní příponu do systému Windows. Ale předtím se podíváme na to, jaké nové zajímavé komponenty se objevily v Delphi 6.

Jistě všichni znáte Průzkumníka Windows, který slouží ke správě souborů. Na následujících několika stránkách si ukážeme, jak postupovat v případě, že byste si chtěli podobný program naprogramovat sami. A začneme hned tím nejjednodušším způsobem, a tím je použití hotových komponent.

„Průzkumné“ komponenty

Pro procházení a zobrazování složek a souborů se v Delphi 6 poprvé objevila sada komponent, které tuto činnost značně zjednodušují. Tyto komponenty naleznete na paletě komponent **Sample** a jsou celkem čtyři:

- **ShellTreeView** – Komponenta, která zobrazuje stromovou strukturu disků a složek.
- **ShellListView** – Komponenta obsahující seznam souborů a vnořených složek ve zvolené složce.
- **ShellComBox** – Rozvírací nabídka, která obsahuje seznam dostupných diskových jednotek a složek.
- **ShellChangeNotifier** – Nevizuální komponenta, která vyvolá událost, pokud se změní obsah zvolené složky.

Použití jednotlivých komponent je velice jednoduché a základní funkčnost je zajištěna i bez nutnosti doplňovat vlastní kód. Postačí nastavit jen některé vlastnosti a získáte tím plně funkční prohlížeč souborů a složek. Bohužel nápověda Delphi o těchto komponentách zarytě mlčí, a proto se na ně podíváme postupně podrobněji a ukážeme si, jak s nimi pracovat.

Komponenta ShellTreeView

Po umístění komponenty na formulář můžete ihned procházet soubory a složky na disku.

Následující tabulka obsahuje přehled nejdůležitějších vlastností a metod, které tato komponenta poskytuje.

Název	Typ	Popis
AutoContextMenu	Vlastnost	Pokud je nastavena na true, po stisku pravého tlačítka se objeví standardní místní nabídka.
AutoRefresh	Vlastnost	Nastavením na hodnotu true se zajistí automatická aktualizace obsahu komponenty.

Název	Typ	Popis
FullCollapse	Procedura	Procedura skryje všechny vnořené úrovně.
FullExpand	Procedura	Procedura zobrazí všechny vnořené úrovně.
HideSelection	Vlastnost	Pokud má tato vlastnost hodnotu false , bude vybraná položka zvýrazněná i tehdy, pokud komponenta ztratí zaměření.
ObjectTypes	Vlastnost	Sada příznaků, které rozhodují o tom, zda se mají zobrazovat složky, soubory a zástupci nebo skryté objekty.
RightClickSelect	Vlastnost	Rozhoduje o tom, zda je možné vybírat položky pravým tlačítkem myši.
Root	Vlastnost	Zde je možné zadat, ve které složce zobrazovaná stromová struktura začíná.
SelectedFolder	Funkce	Funkce vrací hodnotu typu TShellFolder , která obsahuje vybraný adresář.
ShellComboBox	Vlastnost	Komponentu je možné napojit na ShellComboBox tak, že se zobrazuje složky vybrané v nabídce zadané v této vlastnosti.
ShellListView	Vlastnost	Je-li zde zadána komponenta ShellListView , potom tato komponenta automaticky zobrazuje obsah vybrané složky.
ShowButtons	Vlastnost	Pomocí této vlastnosti je možné přepínat zobrazení rozvíracích tlačítek vedle položek obsahujících vnořené položky.
ShowLines	Vlastnost	Pokud je tato vlastnost nastavena, zobrazují se vodící čáry ve stromové struktuře.
ShowRoot	Vlastnost	Vlastnost určuje, zda je u kořenové složky k dispozici "rozbalovací" tlačítko.
UseShellImage	Vlastnost	Pokud má hodnotu true , jsou u jednotlivých položek zobrazeny systémové ikony.



Funkce **SelectFolder** vrací hodnotu typu **TShellFolder**. To je třída, která uchovává informace o složkách. Většinu z nich nebudete nikdy potřebovat, ale například jméno složky lze zjistit pomocí členské metody **DisplayName** a jméno složky včetně kompletní cesty metodou **PathName**.

Komponenta ShellListView

Tato komponenta zobrazí obsah zvolené složky ve zvoleném formátu. Je možné použít velké ikony, malé ikony nebo seznam položek. Komponenta může snadno spolupracovat s předchozí komponentou nebo s komponentou **ShellComboBox**. V následující tabulce můžete opět najít přehled nepoužívanějších vlastností a metod:

Název	Typ	Popis
AutoNavigate	Vlastnost	Pokud je tato hodnota true, a poklepete myší na vybranou položku tak komponenta zobrazí buď obsah vybrané složky, nebo spustí vybraný soubor.
Back	Procedura	Zobrazí se předchozí seznam.
DeleteSelected	Procedura	Vymaže vybranou položku.
FlatScrollBars	Vlastnost	Nastavením této vlastnosti na true se použijí ploché posuvníky.
GridLines	Vlastnost	Jednotlivé řádky seznamu jsou od sebe odděleny mřížkou.
HotTrack	Vlastnost	Pokud je tato vlastnost nastavena na true, není nutné na položku, která se má vybrat, klepat, ale stačí na ni pouze najet myší.
IconOption	Vlastnost	Sada příznaků, které určují, jakým způsobem se budou v seznamu zobrazovat ikony.
MultiSelect	Vlastnost	Vlastnost určuje, zda je možné současně vybrat více než jednu položku.
Refresh	Procedura	Aktualizuje obsah komponenty.
ShellComboBox	Vlastnost	Pokud je v této vlastnosti přiřazena nějaká hodnota, seznam se automaticky aktualizuje podle výběru v zadané komponentě ShellComboBox.
ShellTreeView	Vlastnost	Pokud je zde zadána komponenta ShellTreeView obsah seznamu je automaticky aktualizován podle vybrané složky.
ShowColumnHeader	Vlastnost	Přepíná zobrazení popisků sloupců v detailním zobrazení.
Sorted	Vlastnost	Určuje, zda bude seznam setříděný nebo ne.
ViewStyle	Vlastnost	Nastavením této vlastnosti je možné měnit způsob zobrazení seznamu. Volit lze mezi velkými ikonami, malými ikonami, seznamem a podrobnostmi.



*Předchozí tabulka neobsahuje některé vlastnosti a metody, které jsou stejné jako v případě komponenty **ShellTreeView** a jsou umístěny v tabulce věnované této komponentě.*

ShellComboBox

ShellComboBox je komponenta odvozená od třídy **TComboBox**, od které zdědila i většinu svých vlastností. Protože rozvírací nabídka této komponenty obsahuje seznam složek dostupný na daném počítači, přibylы ke zděděným vlastnostem ještě některé další, z nichž nejdůležitější jsou následující čtyři:

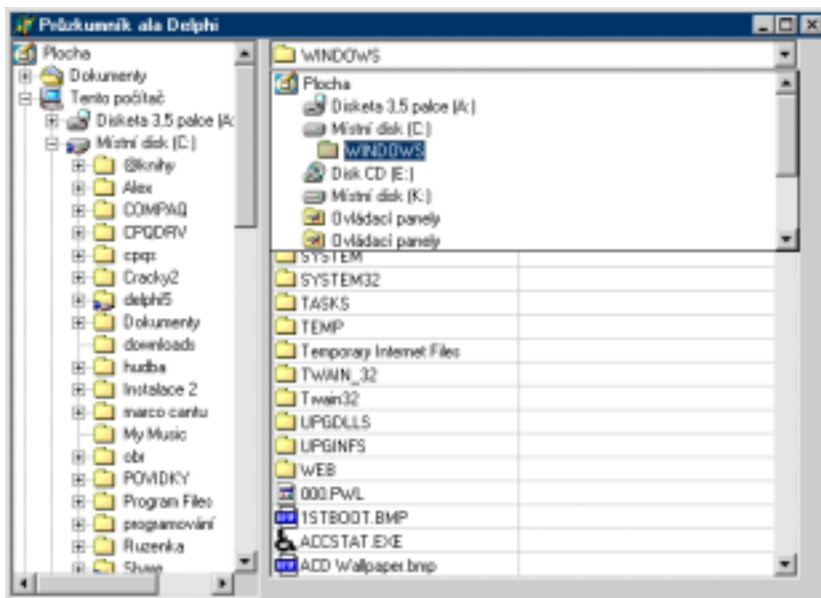
- **Root** – Pomocí této vlastnosti je možné nastavit křenovou složku zobrazovaného seznamu.
- **UseShellImage** – Pokud je tato vlastnost nastavena na hodnotu true, potom jsou k jednotlivým položkám v seznamu přiřazeny ikony používané systémem.
- **ShellListView** – Pomocí této vlastnosti je možné propojit komponentu **ShellComboBox** s komponentou **ShellListView**
- **ShellTreeView** – Nastavením této vlastnosti se komponenta propojí s komponentou **ShellTreeView**.

ShellChangeNotifier

Poslední ze zkoumané sady komponenty je nevizuální a plní úlohu hlídacího psa, který štěkáním informuje svého majitele, když dojde ke změně v hlídané oblasti. Hlídanou oblastí je v tomto případě zvolená složka a štěkání je nahrazeno vyvoláním události **onChange**.

Kromě jména má tato komponenta jen tři zajímavé vlastnosti:

- **Root** – Jméno kořenové složky, která se má kontrolovat.
- **NotifyFilters** – Sada hodnot, které udávají, jaké změny se mají hlásit. Možné jsou následující hodnoty, nebo jejich kombinace:
- **nfFileNameChange** – Událost je vyvolána při změně jména souboru ve sledované složce.
- **nfDirNameChange** – Přejmenování složky vyvolá událost.
- **nfAttributeChange** – Událost reaguje na změnu atributů souboru.
- **nfSizeChange** – Komponenta hlásí změnu velikosti souboru.
- **nfWriteChange** – Událost je vyvolána při zápisu do souboru.
- **NfSecurityChange** – Událost je vyvolána při změně zabezpečení dané složky.
- **WatchSubTree** – Pokud je tato vlastnost nastavena na hodnotu true, potom se hlásí i změny uvnitř vnořených složek.



Věřili byste, že toto není skutečný průzkumník?

Následující obrázek ukazuje, jak může vypadat aplikace s použitím uvedených komponent.



Všechny komponenty uvedené v této kapitole jsou obsaženy pouze v Delphi 6. Majitelé starších verzí Delphi se bez nich budou muset bohužel obejít.

Zaregistrování přípony souboru

Každý začínající uživatel *Windows* dobře ví, že pokud v *Průzkumníkovi Windows* poklepe na soubor s příponou TXT, otevře se vybraný soubor v zápisníku, soubor BMP v programu malování atd.

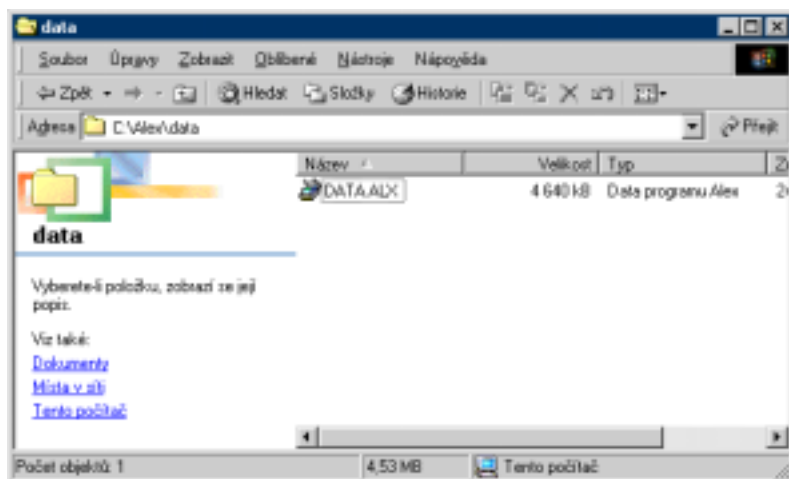
Nyní si ukážeme, jak postupovat, pokud chcete, aby se soubory se zvolenou příponou otvíraly ve vaší aplikaci. V první řadě je nutné, aby program, který chcete k otevírání používat, uměl pracovat s parametry příkazové řádky. Pokud nevíte, jak to zařídit, nalistujte si znovu kapitulu jedenáct.

Dále je nutné příslušnou příponu zaregistrovat do systému *Windows*. Seznam registrovaných přípon je uložen v systémovém registru v části **HKEY_CLASSES_ROOT**. Předpokládejme, že chceme zaregistrovat soubory typu ALX tak, aby je bylo možné otvírat pomocí programu alex.exe. Potom je nutné postupovat podle následujících bodů:

1. Spustíte program **Regedit**.
2. Vyberte větev **HKEY_CLASSES_ROOT** a stiskněte pravé tlačítko myši.

3. Z místní nabídky vyberte položku **Vytvořit → Nový klíč**.
4. Nový klíč pojmenujte **.alx**.
5. Do výchozí hodnoty tohoto klíče uložte text: **soubor ALX** (stačí zobrazit obsah klíče a z místní nabídky vybrat příkaz **Upravit**).
6. Vytvořte další klíč s názvem **soubor ALX**.
7. Uvnitř tohoto klíče vytvořte vnořený klíč s názvem **DefaultIcon**.
8. Jako hodnotu tohoto klíče zadejte jméno souboru, který obsahuje ikonu, jež se má asociovat se zvolenou příponou. Za jménem souboru je uvedeno pořadí ikony ve spustitelném souboru (například **c:\alex\alex.exe, 1**).
9. V klíči **soubor ALX** vytvořte další vnořený klíč s názvem **Shell**.
10. Uvnitř tohoto klíče vytvořte další vnořený klíč s názvem **Open**.
11. Do tohoto klíče vnořte ještě jeden vnořený klíč a nazvěte jej **Command**.
12. Jako výchozí hodnotu tohoto klíče zadejte příkaz, který se má provést po poklepnání na zvolený soubor. Například spuštění programu s vybraným souborem jako parametrem se zapíše takto: **c:\alex\alex.exe %1**.
13. Zavřete program **Regedit** a restartujte počítač.

Pokud jste postupovali přesně podle uvedených bodů, je nový soubor zaregistrován, a když se na něj podíváte v *Průzkumníkovi*, měl by být označen zvolenou ikonou, a pokud na něj poklepete myší, spustí se ve vybraném programu. Jestliže navíc zobrazíte podrobnosti souboru, uvidíte i popis, který jste uvedli v registrech, tak jak je to vidět na následujícím obrázku.



Zaregistrovaný soubor lze spustit ve zvoleném programu pouhým poklepnáním.

V případě, že chcete soubory používané vaší aplikací tímto způsobem zaregistrovat, a nechcete, aby si uživatelé museli sami provádět změny v registrech, můžete snadno registraci naprogramovat automaticky. Uvedený poměrně pracný postup lze například nahradit jediným stiskem tlačítka, pokud se po jeho stisknutí provede následující kód:

```

...
procedure TForm1.Button1Click(Sender: TObject);
var reg: TRegistry;
begin

  reg:=TRegistry.Create;
  try
    reg.RootKey := HKEY_CLASSES_ROOT;

    if reg.OpenKey('.alx', True) then
      begin
        reg.WriteString('', 'soubor ALX');
        reg.CloseKey;
      end;
    if reg.OpenKey('soubor ALX', true) then
      begin
        reg.WriteString('', 'Data programu Alex');
        reg.CloseKey;
      end;

    if reg.OpenKey('soubor ALX\DefaultIcon', true) then
      begin
        reg.WriteString('', 'c:\alex\alex.exe, 1');
        reg.CloseKey;
      end;
    if reg.OpenKey('soubor ALX\Shell\Open\Command', true)
then
      begin
        reg.WriteString('', 'c:\alex\alex.exe %1');
        reg.CloseKey;
      end;
    finally
      reg.Free;
    end;
end;
...

```



*Nezapomeňte na začátku programu připojit jednotku **registry**.*

Pokud chcete uložené informace z registru opět odstranit, je kód ještě jednodušší:

```

...
procedure TForm1.Button2Click(Sender: TObject);
var reg: TRegistry;

```

```
begin  
  
  reg:=TRegistry.Create;  
  try  
    reg.RootKey := HKEY_CLASSES_ROOT;  
  
    if reg.DeleteKey('.alx') then  
      //pokud se podařilo odstranit jeden klíč  
      reg.DeleteKey('soubor ALX') //odstranit i další  
    else  
      ShowMessage('Přípona není zaregistrována');  
  finally  
    reg.Free;  
  end;  
  
end;  
...
```

Pokud nechcete po každém zaregistrování nebo odregistrování přípony restartovat počítač, můžete použít funkci z jednotky **ShlObj**, která donutí systém k obnově informací o typech souborů. Stačí nakonec obou uvedených procedur přidat následující řádek:

```
...  
SHChangeNotify(SHCNE_ASSOCCHANGED, SHCNF_IDLIST, nil, nil);  
...
```

Nyní se informace o souboru zaktualizují ihned po provedené změně.