

Vážení zákazníci,

dovolujeme si Vás upozornit, že na tuto ukázkou knihy se vztahují autorská práva, tzv. copyright.

To znamená, že ukáзка má sloužit výhradně pro osobní potřebu potenciálního kupujícího (aby čtenář viděl, jakým způsobem je titul zpracován a mohl se také podle tohoto, jako jednoho z parametrů, rozhodnout, zda titul koupí či ne).

Z toho vyplývá, že není dovoleno tuto ukázkou jakýmkoliv způsobem dále šířit, veřejně či neveřejně např. umístováním na datová média, na jiné internetové stránky (ani prostřednictvím odkazů) apod.

redakce nakladatelství BEN – technická literatura
redakce@ben.cz



Vytváření síťového diagramu z databáze: příklad

Vzorový program popisovaný v tomto oddílu ukazuje příklad, jak lze vytvořit jednoduchý síťový diagram na základě obsahu externí databáze. Úplný zdrojový kód programu naleznete v modulu NetDB1, uloženém v souboru \DVS\WBA SOLUTIONS\NETDB.VST. Databáze NETWORK.MDB (ve Visio 5.0), resp. NETDB.MDB (Visio 2000) použitá v tomto příkladu byla vytvořena v programu Microsoft Access a je rovněž umístěna ve složce dvs.

Program vytváří výkres Visio tak, že nejprve otevře zvolenou databázi a nastaví vlastnosti dokumentu a výkresové stránky. Poté na výkresovou stránku umístí vzorový obrazec Ethernet a pro každý záznam databáze vytvoří jeden uzel síťového diagramu.

Před vytvořením uzlu program zakáže obnovování obsahu obrazovky. Proces vytvoření uzlu se skládá z umístění vzorového obrazce uzlu do výkresové stránky, označení instance obrazce a z připojení obrazce Ethernet do uzlu. Program rovněž provádí formátování prvního řádku popisu obrazce. Obsah obrazovky se obnovuje až po vytvoření každého uzlu diagramu, uživatel tedy uvidí celkový výsledek procesu vytvoření uzlu síťového diagramu, nikoli jeho jednotlivé etapy.

Otevření databáze

Databázový soubor NETWORK.MDB (resp. NETDB.MDB) obsahuje jedinou tabulku pojmenovanou NetInfo. Každý záznam v tabulce obsahuje čtyři textová pole: Name (název), Node (uzel), Dept (oddělení) a Spec (dodatek). Pole Name a Dept se používají k označení uzlů diagramu. Pole Node obsahuje název vzorového obrazce Visio uloženého v knihovně NETDB.VSS, který má být použit pro grafické ztvárnění daného uzlu ve výkresu. Pole Spec obsahuje další informace využitelné ve výkresu.

Name	Node	Dept	Spec
AndyJ	Printer	Product Development	HPLJ III
CndyM	Machtosh	Creative Services	16MB 500MB
JohnF	Desktop PC	Marketing	12 MB 500MB
KerID	Desktop PC	Technical Support	8MB 320MB
MitchS	Workstation	Product Development	24MB 720MB

Program Network Diagrammer (Visio 5.0)

Program Network Diagrammer (NETDIAG.EXE) představuje složitější verzi programu popisovaného v tomto oddílu. Pokud vás zajímá, jak program Network Diagrammer pracuje, prostudujte si jeho zdrojový kód uložený v souboru MAIN.BAS ve složce \DVS\VB SOLUTIONS\NETDIAG.

Využitím metod objektů DAO (Data Access Objects), které zprostředkovávají přístup k externím datům program otevře databázi NETWORK.MDB a soubor záznamů uložený v databázi přiřadí do objektové proměnné nazvané *NetInfo*:

```

Dim NetBase as DAO.Database
Dim NetInfo as DAO.RecordSet
Set NetBase = OpenDatabase(Visio.Application.Path _
& "dvs\vb solutions\netdb\network.mdb", True, True)
Set NetInfo = NetBase.OpenRecordset("NetInfo", _
dpOpenShapshot)

```

Podrobnosti o přístupu k datům uloženým v databázi naleznete v dokumentaci Visual Basicu.

Umístění vzorového obrazce Ethernet

Program zahajuje tvorbu síťového diagramu převzetím vzorového obrazce Ethernet z kolekce Masters knihovního souboru a umístěním tohoto vzorového obrazce síťového diagramu. Instance bude později umístěna s větší přesností. Objekt Shape vrácený metodou Drop reprezentuje nový obrazec Ethernet ve výkresové stránce.

```

Set Master = NetStencil.Masters("Ethernet")
Set Ethernet = NetDiagram.Drop(Master, 1, 1)

```

Obrazec Ethernet patří do skupiny 1D obrazců, obsahuje tedy počáteční a koncový bod namísto uchopovacích bodů výběru. Program využívá metod **SetBegin** a **SetEnd** objektu reprezentujícího obrazec Ethernet k umístění počátečního bodu obrazce na pozici [0,5, 2,0] a koncového bodu na pozici [7,5,2,0].

```

Ethernet.SetBegin .5, 2
Ethernet.SetEnd 7.5, 2

```

Obnovování obsahu obrazovky

Po umístění obrazce Ethernet do výkresové stránky prochází program smyčkou, v níž je pro každý záznam databáze vložen do výkresu jeden uzel síťového diagramu. Nejprve se zaměříme na řídicí část smyčky, kód uvnitř smyčky bude podrobně rozebrán později. Před vložením vzorového obrazce pro uzel diagramu je vypnuto automatické obnovování obsahu obrazovky, takže celý proces vytváření uzlu probíhá skrytě a uživatel na obrazovce uvidí až celkový výsledek. Toto řešení navíc oproti režimu s povolenou volbou **ScreenUpdating** urychluje vykonávání příkazů programu. V případě velmi složitých výkresů je dokonce možné vyjmout obnovu obrazovky i z této smyčky a snížit tak počet obnovení obsahu obrazovky na minimum.

```

NetInfo.MoveFirst
XPos# = 1
Digit% = Asc("1")
While Not NetInfo.EOF
    Visio.ScreenUpdating = False
    ...
    'Umístit uzel diagramu
    'Označit uzel
    'Připojit obrazec Ethernet do uzlu
    'Zformátovat označení uzlu
    ...
    Visio.ScreenUpdating = True
    NetInfo.MoveNext
Wend

```

Umístění uzlu síťového diagramu

Program provádí umístění uzlu diagramu pomocí metody **Drop**. Pro každý uzel program umístí vzorový obrazec definovaný v poli Node příslušného databázového záznamu. Souřadnice předávané metodě **Drop** určují umístění pinu vzorového obrazce. V případě obrazců použitých v tomto příkladu představuje pin obrazce jeho střed. Pro větší srozumitelnost je v následujících příkladech zdrojových kódů opakovaně užíván počáteční a koncový příkaz řídicí smyčky.

```

While Not NetInfo.EOF
    ...
    NodeType$ = NetInfo.Fields("Node")
    Set Master = NetStencil.Masters(NodeType$)
    Set Shape = NetDiagram.Drop(Master, XPos#, .875)
    XPos# = XPos# + 1.5
    ...
Wend

```

Označení uzlu

Program vytváří štítek s označením každého uzlu využitím polí Name a Dept databázové tabulky. Jakmile je označení uzlu vygenerováno, je text přiřazen do vlastnosti **Text** příslušného obrazce.

```

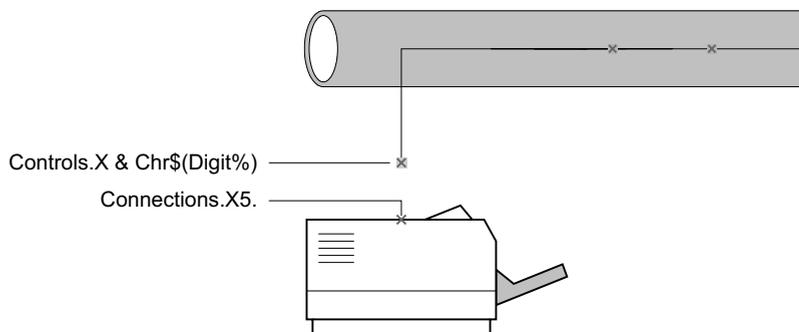
While Not NetInfo.EOF
...
Label$ = NetInfo.Fields("Name")
Label$ = Label$ & Chr$(13) & Chr$(10)
Label$ = Label$ & NetInfo.Fields("Dept")
Shape.Text = Label$
...
Wend

```

Připojení uzlu k obrazci Ethernet

Přímky obrazce Ethernet jsou opatřeny řídicími body, které mohou být připojeny k uzlům. Program může připojit libovolný řídicí bod obrazce Ethernet do připojovacího bodu uzlu.

V uvedeném příkladu je v objektové proměnné *ControlCell* uložen odkaz na řídicí bod *Controls.Xn* obrazce Ethernet. Proměnná *Digit%* vybírá jeden řídicí bod pro každý uzel. Každý uzel je opatřen několika připojovacími body, pátý připojovací bod je umístěn na vrcholu obrazce a použití tohoto řídicího bodu je tedy nejvýhodnější.



Připojení obrazce Ethernet k uzlu síťového diagramu

Program nastavuje objekt *ConnectCell* na buňku pátého připojovacího bodu obrazce představujícího uzel diagramu (*Connections.X5*). Poslední příkaz užívá metodu *GlueTo* k připojení řídicího bodu do tohoto bodu připojení.

```

While Not NetInfo.EOF
...
Set ControlCell = Ethernet.Cells("Controls.X" & _
Chr$(Digit%))
Digit = Digit + 1
Set ConnectCell = Shape.Cells("Connections.X5")
ControlCell.GlueTo ConnectCell
...
Wend

```

Formátování označení uzlu

V závěru program formátuje písmo v prvním řádku každého štítku uzlu jako tučné. Tato akce vyžaduje poněkud hlubší znalosti prostředí Visio, neboť vyžaduje získání obrazce, který je součástí skupiny, a navíc pracuje pouze s částí textu zvoleného obrazce.

Získání prvku skupiny. Obrazec uzlu diagramu představuje skupinu Visio. Pokud byl k uzlu přiřazen štítek, nebyl přiřazen skupině, ale jistému členu skupiny, a to prvku skupiny uloženému zcela nahoře. Aby mohl program pracovat s textem přiřazeným k uzlu, musí získat přístup k tomuto prvku skupiny.

Skupina Visio je z pohledu Automation představována objektem Shape s kolekcí Shapes. Aby mohl program pracovat s obrazcem uloženým ve skupině zcela nahoře, musí nejprve získat informaci o počtu obrazců v kolekci Shapes a poté získat odkaz na obrazec s nejvyšším pořadovým.

```

While Not NetInfo.EOF
...
Index = Shape.Shapes.Count
Set TextShape = Shape.Shapes(Index)
...
Wend

```

Práce s částí textu obrazce. Program potřebuje pro práci s určitou částí textu obdržet objekt Characters. Tento objekt je možné získat pomocí vlastnosti **Characters**. Hodnoty vlastností **Begin** a **End** objektu Character určují počáteční a koncový znak vybrané části textu. V našem případě je třeba vybrat počátek textu až do výskytu prvního znaku nového řádku. Inicializační hodnota vlastnosti **Begin** je rovna nule, takže tuto hodnotu není třeba nastavovat, hodnota vlastnosti **End** je nastavena na jeden znak před znakem nového řádku (Chr\$(13)).

Po nastavení objektu Characters na požadovanou část textu provede program formátování znaků na tučné písmo. Konstanty **visCharacterStyle** a **visBold** jsou definovány v typové knihovně Visio.

```
While Not NetInfo.EOF
...
Set Chars = TextShape.Characters
Chars.End = InStr(TextShape.Text, Chr$(13)) - 1
Chars.CharProps(visCharacterStyle) = visBold
...
Wend
```

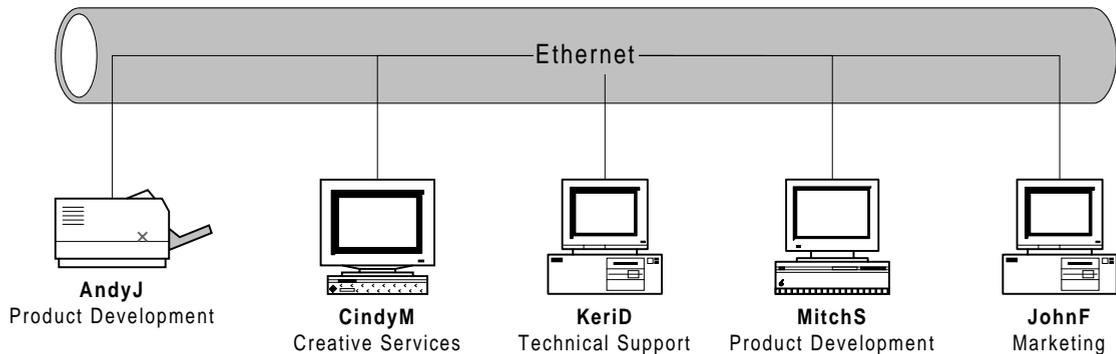


Diagram vytvořený z databáze NETWORK.MDB (Visio 5.0), resp. NETDB.MDB (Visio 2000).