

MS ACCESS A MS WORD V KAŽDODENNÍ PRAXI

Martin Fussek

VL Servis s.r.o., Ruská 24/83, 706 02 Ostrava – Vítkovice, fussek@lahvarna.cz

Abstrakt

Príspevek volne navazuje na mój príspevek na TS 1999, pričomž nyní je zaměřen na tvorbu databází a jednoduchého evidenčního systému firmy zcela založeného na technologii MS Access a MS Word. V příspěvku jsou zhodnoceny dosavadní zkušenosti s tvorbou databázových aplikací využitelných v každodenní práci malé firmy. Na příkladech demonstruje jak jednoduché je navrhnout a realizovat a následně i spravovat různé aplikace propojující „velký“ IS společnosti s vlastními systémy. Příspěvek poukazuje na rozdíly mezi různými verzemi balíku Office ve vztahu k navržených aplikacím a jak zajistit maximální přenositelnost.

1. Úvod

Většina běžných uživatelů kancelářského balíku Office využívá ke své práci textový procesor Word (pro psaní a úpravy textů), zkušenější tabulkový procesor Excel (pro analýzy a tabulky), někteří možná i prezentační nástroj PowerPoint (pro prezentaci výsledků) a velmi malá část databázový program Access.

Toto rozdělení uživatelů vychází z každodenních potřeb firmy. Většinu času zabere psaní různých dopisů, objednávek, nabídek a jiné „obchodní“ korespondence. V minulém příspěvku jsem se věnoval tomu, jak ulehčit práci s vyplňováním hlavičkových dokumentů za pomoci jednoduchých formulářů a maker. Toto uvedené řešení již přináší značné zjednodušení, ale dalšího kvalitativního posunu dosáhneme napojením formuláře na databázi. Přinejmenším můžeme využít databáze kontaktů.

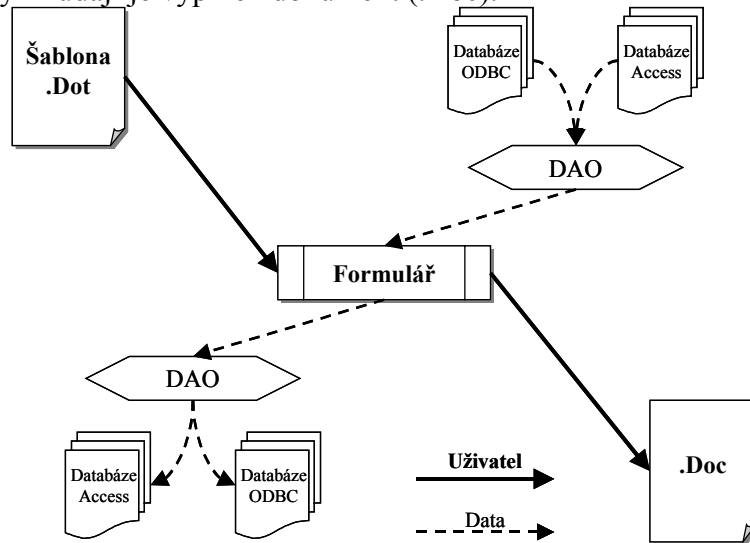
Programovací nástroje programu Word (a právě využití programu těchto nástrojů) nám umožňují postupné zlepšování a zvyšování možností aplikace podle dodatečných požadavků uživatelů („testováno na lidech“)

Považuji za nutné zdůraznit, že aplikace založené na programovacích možnostech balíku Office, které nemohou nikdy v oblasti programování nahradit „plnohodnotné“ programovací jazyky (C++, Delphi, Visual Basic, Java), jsou výhodné zejména z těchto důvodů:

1. Umožňují vytvářet aplikace „na míru“ pouhým přizpůsobením již hotové hostitelské aplikace (Word) a úpravy uživatelských modulů (šablony s programovým kódem);
2. Pracují rychle, protože mohou využívat standardních funkcí hostitelské aplikace;
3. Umožňují uživatelům zůstat ve známém pracovním prostředí a využívat jim známé nástroje a postupy;
4. Umožňují vytvářet vlastní aplikace i zkušeným uživatelům s minimální znalostí programování;
5. Není cílem nahradit plnohodnotný IS společnosti.

2. Základní model

Základní model znázorňuje chování programu, uživatele a tok dat. Na začátku je šablona dokumentu obsahující programový kód a definující uživatelské rozhraní (formuláře). Uživatel pomocí prostředků hostitelské aplikace zadá požadavek na vytvoření nového dokumentu (menu Soubor/Nový...) založeného na zvolené šabloně (.Dot). Šablona otevře formulář, který při svém otevření vytvoří spojení na potřebné databáze (pomocí komponenty DAO, přes rozhraní ODBC, nebo přímo na databázi programu Access). Uživatel vyplní požadované údaje (na jejichž základě mohou být dotazovány databáze, která vrátí požadovaná data) a uzavře formulář. V tu chvíli jsou jím zadaná data uložena do databáze, uzavřeno spojení na databáze a zadanými údaji je vyplněn dokument (.Doc).



Obrázek: Základní model

Model je možno podle potřeby modifikovat. Tedy formulář může/nemusí načítat data z databáze, může/nemusí ukládat data do databáze. Vše záleží na konkrétních podmínkách a požadavcích.

V dalším textu budu demonstrovat praktickou realizaci systému usnadňující vytváření a evidenci objednávek, který plně využívá uvedeného modelu.

3. Popis konkrétního problému

Předtím, než byl realizován systém „Sledování objednávek“ (dále SSO), sloužil program Word pouze jako „inteligentní psací stroj“ pro vyplňování předtištěných objednávek (hlavičkových papírů) a každá objednávka byla evidována písemnou formou v knize objednávek.

Prvním krokem k dnešnímu SSO bylo vytvoření „pasivní“ šablony s formulářem usnadňující vyplňování příslušných kolonek předtištěné objednávky (popis takového formuláře viz příspěvek TS 1999).

Druhým krokem bylo rozšíření formuláře o vyhledávání adres dodavatelů z IS společnosti (tabulka na SQL serveru). Tento krok se osvědčil a výrazně zrychlil vytváření objednávek, protože odboural nutnost vyhledávat adresy jiným způsobem.

Třetím krokem bylo rozšíření formuláře o evidenci objednávek v databázi. Tento krok byl klíčový, protože odboural nutnost evidovat objednávky písemnou formou. Bohužel bylo potřeba vytvořit nástroj pro vyhledávání a editaci uložených objednávek (opět na bázi Word).

Čtvrtým krokem bylo vytvoření aplikace v Accessu, která umožňuje plný uživatelský komfort vyhledávání, úpravy a správy objednávek, tisk seznamů atp.

Celá koncepce systému je postavena využití programů Word (uživatelské rozhraní) a Access (v podobě databáze dat a uživatelského rozhraní). Zadávání nových objednávek do systému je umožněno pouze z formuláře šablony Word.

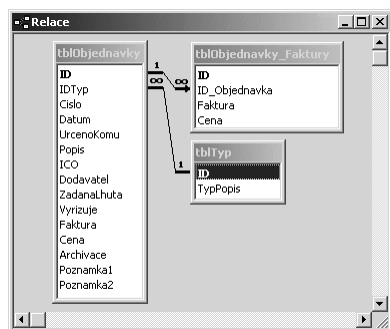
4. Programujeme Word

4.1 Základní formulář

Formulář objednávky obsahuje položky typ objednávky (seznam z databáze MDB), číslo objednávky (automaticky generované z databáze MDB), žádaná lhůta (uživatel), určeno komu (seznam z databáze MDB + uživatel), popis (uživatel), pole adresát (uživatel + vyhledávání z databáze ODBC), vyřizuje (registr), datum (aktuálně), telefon (registr), fax (registr).

Formulář je plně použitelný i bez databází. Použití databází přináší nové funkce a možnosti – vyhledávání v adresáři a vytvoření nové objednávky (jediný způsob jak vytvořit v databázi objednávku je přes tento formulář). Uživateli je také umožněn tisk již existujících a v databázi uložených objednávek.

4.2 Databáze



Databáze je typu MDB (Microsoft Access) vytvořená v programu Microsoft Access. Databáze obsahuje tři tabulky: tblObjednavky (údaje o objednavce), tblObjednavky_Faktury (tabulka přidaná na základě požadavků uživatelů, není k ní přístup přes Word), tblTyp (tabulka typů objednávky – určuje číselnou řadu). Tabulky jsou ve vzájemné relaci.

4.2.1 Typy polí tabulek

S ohledem na jednoduchost a možnosti programu Word doporučuji většinu polí typu text, nebo memo a to i v případě, že obsahují číselnou hodnotu. S číselnou hodnotou (zvláště pokud je desetinná) se velmi špatně pracuje – problém desetinná tečka/čárka. V případech, kdy to je to

Název pole	Datový typ	Popis
ID	automatické číslo	
IDTyp	text	Typ Strojní/Elektro
Cislo	číslo	Číslo objednávky
Datum	datum/čas	Datum vystavení
UrcenoKomu	text	Určeno komu
Popis	memo	Text objednávky
ICO	text	ICO dodavatele
Dodavatel	text	Název dodavatele
ZadanaLhuta	text	Požadovaná lhůta dodání
Vyřizuje	text	Vyřizuje
Faktura	text	Číslo faktury (nepoužito)
Cena	měna	Cena (nepoužito)
Archivace	datum/čas	Datum archivace objednávky
Poznámka1	text	Poznámka 1
Poznámka2	text	Poznámka 2

Vlastnosti pole	
Obecné	Vyhledávání
Velikost pole	50
Formát	
Vstupní maska	
Titulek	Určeno komu
Výchozí hodnota	
Ověřovací pravidlo	
Ověřovací text	
Je nutno zadat	ne
Povolit nulovou délku	ano
Indexovat	ne
Kompresce kódu Unicode	ano

možné je vždy lepší používat textová pole a v případě potřeby obsah pole převádět na hodnoty pomocí příkazů Visual Basic. Pokud to z nějakého důvodu není možné, je potřeba naprogramovat si vlastní obslužné rutiny převádějící tečku na čárku a obráceně.

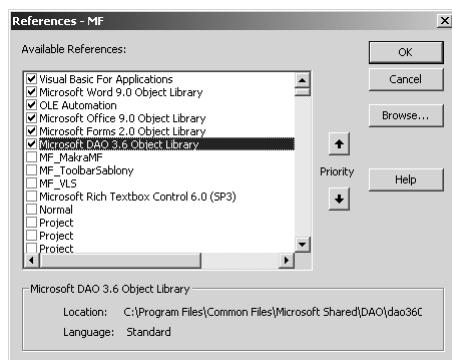
S polem typu datum/čas problémy nevzniknou, protože komponenta DAO umožňuje ve vztahu k databázi MDB zápis SQL příkazu takový, že datum zapsaný jako text (mezi apostrofy) v českém formátu (např.: „14.3.2002“) je korektně převeden do tvaru používaného databází.

Dále je potřeba (v rámci usnadnění) všem polím nastavit vlastnost „Je nutno zadat“ na NE (aby pole mohla obsahovat hodnotu NULL) a „Povolit nulovou délku“ na ANO (aby pole mohla obsahovat prázdný znak). Opět podotýkám, že pokud to z nějakého důvodu není možné, je potřeba naprogramovat vlastní rutiny, obsluhující kontrolu zadání a převod prázdných znaků na hodnotu NULL atp.

Zde uvedené se týká pouze případu, pokud chceme do databáze typu MDB zapisovat pomocí DAO. Pokud ji budeme pouze číst neplatí žádná omezení. Zápis do databází typu ODBC jsem z programu Word nikdy neprováděl a tudíž s nimi nemám žádné zkušenosti.

4.3 Propojení na databáze

Propojení na databáze je realizováno pomocí komponenty Microsoft DAO, která je standardní součástí instalace programového balíku Office. Pokud není instalována, je nutno ji doinstalovat instalačním programem balíku a volbou „Komponenty pro přístup k datům“.



Aby bylo možno s komponentou v projektu Word pracovat, je potřeba na ni vytvořit odkaz. V prostředí editoru jazyka Visual Basic (menu Nástroje/Makro/Editor jazyka Visual Basic Alt+F11) zvolíme menu Tools/References... a zaškrtneme volbu „Microsoft DAO 3.6 Object Library“. Verze komponenty se může lišit v závislosti na verzi balíku Office a jiných instalacích v systému. Standardně je pro Office 97 verze 3.5 a pro Office 2000 verze 3.6. Obrázek ukazuje nastavení všech voleb, které by měly být zaškrtnuty.

Komponenta DAO využívá pro propojení na databáze pracovní prostory dvou typů. ODBCdirect workspace pro spojení s databází přes rozhraní ODBC a Microsoft Jet workspace pro spojení s databázemi Access (.mdb). Pro bližší podrobnosti překračující rámec tohoto příspěvku odkazuji na nápovědu, která je poměrně rozsáhlá a zájemce v ní najde všechny požadované informace.

V mých programech vytvářím spojení na databázi v metodě UserForm_Initialize formuláře . Pracovní prostory a spojení jsou uloženy v globálních proměnných formuláře.

```
Dim wrkODBC As Workspace
```

```
Dim wrkJet As Workspace
```

```
Dim conPubs As Connection
```

```
Dim dbsPubs As Database
```

4.3.1 Spojení s databází ODBC

```
Set wrkODBC = CreateWorkspace("NewODBCWorkspace", strUID, strPWD,
dbUseODBC)
Set conPubs = wrkODBC.OpenConnection("MFConnection1", dbDriverNoPrompt,
"ODBC;DSN=" & strDSN & ";DATABASE=" & strDATABASE & ";UID=" & strUID &
";PWD=" & strPWD)
```

4.3.2 Spojení s databází MDB

```
Set wrkJet = CreateWorkspace("NewJetWorkspace", "admin", "", dbUseJet)
Set dbsPubs = wrkJet.OpenDatabase(strDATABASE, False, False, "")
```

4.4 Dotazování databáze

Databáze je dotazována SQL příkazy, které vrací záznamy ve formě množiny záznamů (Recordset). Dotazování MDB i ODBC databáze je shodné a provádí se metodou OpenRecordset příslušného spojení.

4.4.1 Dotaz na databázi ODBC

Databázi ODBC (uloženou na SQL serveru) dotazují při změně obsahu textového pole Adresát. Programový kód sestaví SQL dotaz, dotáže databázi, vrácené záznamy uloží do pole (Array) a následně přiřadí obsah pole seznamu (List mající 7 sloupců) na formuláři. Uložení záznamů do vícerozměrného pole a jeho následné přiřazení seznamu je výrazně rychlejší variantou, než do pole přidávat jednotlivé záznamy postupně (je totiž nutné nejprve přidat nový řádek seznamu a poté jednotlivě přidávat další sloupce). Jedná-li se o seznam o jednom sloupci, je možno přímé přidávání řádků. Je důležité na konci procedury uzavřít recordset.

```
strWhere1 = "WHERE Nazev1 like '%" & strText & "%' OR " & "Nazev2 like '%" &
strText & "%' OR " & "ODPOS like '%" & strText & "%'"
strSQL1 = "SELECT NAZEV1, NAZEV2, ULICE, PSC + ' ' + MESTO, ODPOS, ICO,
FAX FROM ICO " & strWhere1
strOrderBy = " ORDER BY Nazev1, Nazev2 DESC"
Set rstTemp = conPubs.OpenRecordset(strSQL1 & strOrderBy, dbOpenSnapshot)
intRecordCount = rstTemp.RecordCount
ReDim MyArray(rstTemp.RecordCount - 1, 6)
...
With rstTemp
  i = 0
  Do While Not .EOF ' Enumerate records.
    For j = 0 To 6
      MyArray(i, j) = Trim(.Fields(j).Value)
    Next j
    .MoveNext
    i = i + 1
  Loop
  .Close ' Zavření record setu
End With
Me.ListBox1.List = MyArray
Me.ListBox1.Selected(0) = True
```

```

    End If
End If
4.4.2 Dotaz na databázi MDB
Následující kód dotazuje databázi MDB a ze záznamů vyplní rozbalovací seznam (Combo o jednom sloupci) určeno komu. Je důležité na konci procedury uzavřít recordset.
SQL = "SELECT DISTINCT UrcenoKomu FROM tblObjednavky ORDER BY UrcenoKomu"
Set rstTemp = dbsPubs.OpenRecordset(SQL)
If Err.Number <> 0 Then Exit Sub
With rstTemp
    Do While Not .EOF ' Enumerate records.
        If Not IsNull(rstTemp.Fields("UrcenoKomu")) Then
            If rstTemp.Fields("UrcenoKomu") <> "" Then
                Me.ComboBox_UrcenoKomu.AddItem Trim(rstTemp.Fields("UrcenoKomu"))
            End If
        End If
        .MoveNext
    Loop
    .Close ' Zavření record setu
End With

```

4.5 Uložení objednávky do databáze

Uložení objednávky do databáze se provádí metodou Execute příslušného spojení, které je předán SQL příkaz. S ohledem na způsob zápisu SQL příkazu je potřeba v textových proměnných (kde by se mohly vyskytnout) nahradit apostof jiným znakem, jinak bude zápis SQL neplatný a metoda vrátí chybu.

```

strPopis = Replace(strPopis, "'", "'")
strDodavatel = Replace(strDodavatel, "'", "'")
SQL = "INSERT INTO tblObjednavky (IDTyp, Cislo, Datum, UrcenoKomu, Popis, ICO,
Dodavatel, ZadanaLhuta, Vyrizuje) " & _
"VALUES (" & strIDTyp & "," & strCislo & "," & strDatum & "," & strUrcenoKomu
& "," & strPopis & "," & strICO & "," & strDodavatel & "," & strZadanaLhuta & "," &
strVyrizuje & ")"
dbsPubs.Execute SQL

```

4.6 Vyplnění dokumentu

Vyplnění dokumentu je prováděno pomocí záložek a postup je popsán v mém příspěvku TS 1999.

4.7 Zavření spojení a pracovních prostorů

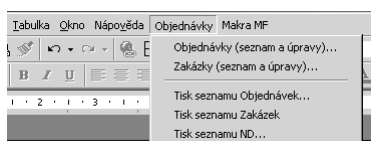
Důležitým krokem je uzavření spojení na otevřené databáze a pracovních prostorů nejlépe v metodě UserForm_QueryClose formuláře.

```

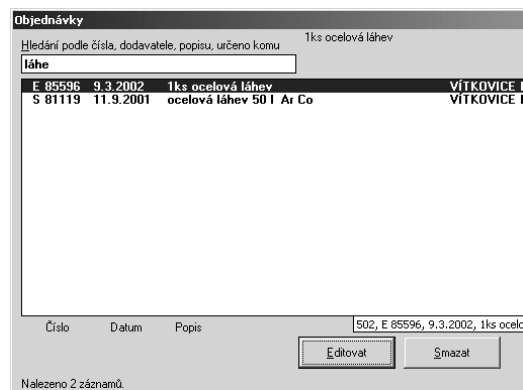
conPubs.Close
wrkODBC.Close
dbsPubs.Close
wrkJet.Close

```

4.8 Vyhledávání objednávek v databázi z programu Word



Program Word nedisponuje žádnými nástroji pro prohledávání databáze. Pomocí šablony doplňků můžeme do aplikace přidat nové menu a v něm vytvořit položky pro volání formuláře vyhledávacího záznamu a umožňujícího úpravy stávajících záznamů. Funkce formuláře je shodná se všemi uvedenými případy. Využívá vyhledávání, zobrazení záznamů, uložení záznamu (UPDATE) a smazání záznamu (DELETE).



Využití formuláře je omezeno pouze počtem vrácených záznamů. V případě, že databáze vrátí více jak 200 záznamů, je vyplnění seznamu (List) časově neúnosné (řádkově vteřiny). V takovém případě je lepší buď omezit počet vrácených záznamů v SQL příkazu, nebo naprogramovat uživatelské prostředí v programu Access a provádět vyhledávání a úpravy záznamů v prostředí Access. Dojde tím k výraznému zrychlení vyhledávání. Navíc Access disponuje možností tvorby sestav a je proto snadné tisknout různé seznamy. Tisk seznamů v prostředí Word je také možný, ovšem je nutno

celou proceduru naprogramovat a platí zde stejné omezení s počtem záznamů. Při překročení počtu 300 záznamů (formátovaných v tabulce v dokumentu) je časové zpoždění vytvoření dokumentu již velmi značné.

5. Zhodnocení aplikace

Navržená a realizovaná aplikace je již více než rok zcela funkční a plně pokrývá potřeby firmy. Umožňuje komfortní vytváření nových objednávek, vyhledávání a díky pokročilým možnostem i prohledávání objednávek, okamžitou kontrolu stavu objednávek (archivace), tisk různých seznamů. Navržené řešení je plně funkční, vykazuje téměř nulovou chybovost a je rozšiřitelné.

Po zavedení SSO byl stejný model použit i pro evidenci přijatých objednávek (zakázek), náhradních dílů a nabídek.

Nezanedbatelné je, že aplikace může vznikat „za chodu“, stejně jako přidávání nových funkcí. Systém nevyžaduje žádnou speciální instalaci (instalaci představuje nakopírování souborů .dot a .mdb do adresáře a nastavení cest k ní).

6. Přenositelnost

Přenositelnost šablon programu Word mezi verzí 97 a 2000 je téměř 100%. Při převodu z verze 97 do 2000 se může stát, že aplikace zhavaruje na nemožnosti najít komponentu DAO příslušné verze (3.6). V tom případě je potřeba otevřít Editor jazka Visual Basic a zrušit neplatný odkaz na verzi 3.6 a změnit ji na verzi 3.5. Po této změně již by měla být aplikace plně funkční.

S přenositelností databází Access je to horší. Verze 97 využívá jiný formát než verze 2000. Je samozřejmě možná téměř bezproblémová konverze mezi těmito formáty, ovšem problémem je jak přistupovat z programu Word 97 do databáze formátu Access 2000. Pravděpodobně je potřeba mít nainstalovány ovladače schopné pracovat s novým formátem databáze. Tento problém jsem nikdy neřešil. Vždy jsem pracoval pouze v prostředí verze 97 nebo 2000.

Obecně lze konstatovat, že nejjednodušším řešením je programovat v té verzi, která bude používána cílovým spotřebitelem. Vytvořil jsem již několik aplikací založených na kombinaci Word a Access (vždy stejné verze) a tyto aplikace jsou plně funkční již několik let bez nutnosti jakéhokoliv zásahu ze strany programátora.