



Automatický upgrade aplikace ve VFP přes FTP server

Rudolf Jalovecký

Resume:

Automatická aktualizace aplikace je v současnosti skoro nezbytným doplňkem každého kvalitního programu. Realizace jakékoliv aktualizace aplikace přináší programátoru náročnou práci v podobě udržování informací o mnoha předchozích verzích, především databázových strukturách, neboť nemůže vědět, kterou verzi právě uživatel používá. Jistou nevýhodou v tomto okamžiku je, že vývojového prostředí VFP pracuje s mnoha soubory, tím se případná změna struktur těchto souborů v aktualizacích složitěji udržuje.

Předložený příspěvek si dává za cíl ukázat programátorům ve VFP, jak relativně jednoduše vytvořit funkční a dostatečně kvalitní automatickou aktualizaci aplikace, aniž by výrazně opustili vývojové prostředí VFP, přičemž využívá FTP přístup na server s uloženou aktualizací.

Curriculum vitae:



Rudolf Jalovecký je vysokoškolským učitelem a dlouholetým vývojářem databázových aplikací, začínal na dBase II. ještě na 8-mi bitových počítačích, pak přešel na dBase III., v pravou chvíli přešel na FoxPro pro DOS, tehdy verze 1.02. Prošel i verzí FoxPro 2.5a, FoxPro pro Windows se mu moc nelíbila a tak posečkal až na VFP, kde začal na verzi 5.0, které zůstal věren přes 10 let. Vývoj jde však dopředu a proto v posledních letech přesedlal na VFP 9.0. Programování ve VFP je jeho velkým koníčkem, profesně se ale věnuje letecké elektronice, konkrétněji palubním počítačům a automatickému řízení letu, které vyučuje na Univerzitě obrany. Profesně i zájmově programuje i v prostředí Delphi a jednočipové mikroprocesory pro palubní vybavení letadel programuje v assembleru.

1. Obsah

1.	Obsah.....	3
2.	Proč automatický upgrade aplikace přes internet.....	4
3.	Představení vlastního řešení	5
3.1.	Idea řešení automatické aktualizace	5
3.2.	Popis programových modulů a jejich funkce	5
3.2.1.	Program FTPUpgrade.EXE.....	7
3.2.2.	Program NewSetupDevCon	11
3.2.3.	Program UpGradeDevCon	13
3.2.4.	Chování aktualizované aplikace DevCon2007	14
3.3.	Návaznost modulů aneb jak uvést celý automatický upgrade do provozu.....	18
4.	Komentář k předaným souborům	19
5.	Literatura a odkazy na web	19

2. Proč automatický upgrade aplikace přes internet

Problematika aktualizace vytvořeného programového díla vzniká prakticky v okamžiku, kdy Vaše dílo začne být uživateli připomínkováno ať již s požadavkem na rozšiřování a doplňování nebo s nutností provedení oprav chyb v programu.

Možností, jak provádět aktualizaci, je vždy několik. Od prostého ručního provádění aktualizace za přítomnosti autora aplikace, přes opět ruční aktualizaci, ale se zasláním aktualizčního souboru uživateli s popisem, jak si má aktualizaci provést. Další možností je poloautomatická aktualizace, zasláním informačního dopisu uživatelům, kde si aktualizční soubor stáhnou a jak si jej nainstalují až po plně automatickou aktualizaci, kdy se skoro o vše postará sama aplikace a po uživateli toho moc nechce.

S vývojovým prostředím Visual FoxPro dostává programátor k dispozici i slušný instalační program InstallShield Express, s nímž bezproblémově vytvoří velmi dobře pracující instalační program své aplikace. Jeho nevýhodou ovšem je, že možnosti upgrade aplikace jsou v „express“ verzi uzamčeny a je nutné zakoupit plnou verzi programu InstallShield. To v řadě případů pro velké firmy asi smysl má, avšak menší firmy či samostatní programátoři zase tak moc nevyužijí jeho vlastnosti.

I velký propagátor Visual FoxPro Doug Hennig měl loni přednášku na 13th European Visual FoxPro DevCon 2006 na téma "proč používám InnoSetup a ne InstallShield a co se mi na tom druhém nelíbí" [1].

Také z tohoto pohledu vznikla myšlenka realizace automatického upgrade aplikace s využitím programu InnoSetup (licence GNU GPL). O „velkou“ a správnou instalaci RT prostředí Visual FoxPro se postará InstallShield Express a „dodatky“ v aktualizované verzi pak má programátor velmi dobře pod kontrolou. Při tomto přístupu vzniká samozřejmě základní problém, že není možné běžící aplikaci „vyměnit“ na disku za aktualizovanou. Musíme tedy zajistit vhodné „předání řízení“ jinému programu, který umožní provést přepis aplikace a případně také změnit všechny požadované úpravy při přechodu od staré k nové verzi.

V následujícím textu je vysvětlen postup zcela automatické aktualizace aplikace od plně automatického zjištění, zda nová verze je někde na internetu k dispozici, přes její stažení (oboje se děje v pozadí běžící aplikace) až po spuštění aktualizace po zásahu uživatele a dokončení všech požadovaných změn struktury aplikace.

3. Představení vlastního řešení

Každý programátor přistoupí k realizaci aplikace určitým a snad i osobitým způsobem. Také realizace automatické aktualizace aplikace je tímto postižena.

3.1. Idea řešení automatické aktualizace

V podstatě vše probíhá tak, že po spuštění aplikace u uživatele se spustí externí program (ve VFP) běžící na pozadí, který otestuje připojení k internetu. Pokud připojení do internetu je, tak se aplikace připojí na definovaný adresář na webu (FTP protokolem) a otestuje, zda je tam nová verze (porovnáním čísel překladu nebo podobně). Při existenci nové verze ji stáhne k uživateli do PC. Vlastní běžící uživatelská aplikace pak v průběhu vlastního běhu otestuje, zda už byla nová verze stažena a uživateli sdělí, že je k dispozici upgrade programu. Pokud uživatel chce, tak si spustí aktualizaci. Před spuštěním aktualizace se ukončí aktualizovaná aplikace, spustí se vlastní upgrade (komprimovaná instalace v InnoSetup), která rozbalí upgrade (novou verzi aplikace), všechny další potřebné soubory či knihovny a malý exe program ve VFP, který je na konci upgrade spuštěn a v příjemném foxovém prostředí provede potřebné změny v tabulkách či dalších souborech. Tím je aktualizace provedena a jen se vymaže stažený instalační soubor s aktualizací a případné další pomocné (dočasné) soubory.

Na obr. 3.1 je naznačena vzájemná souvislost funkcí jednotlivých programů. V přiložené prezentaci je pak tento obrázek rozfázován, aby součinnost programů byla názorná.

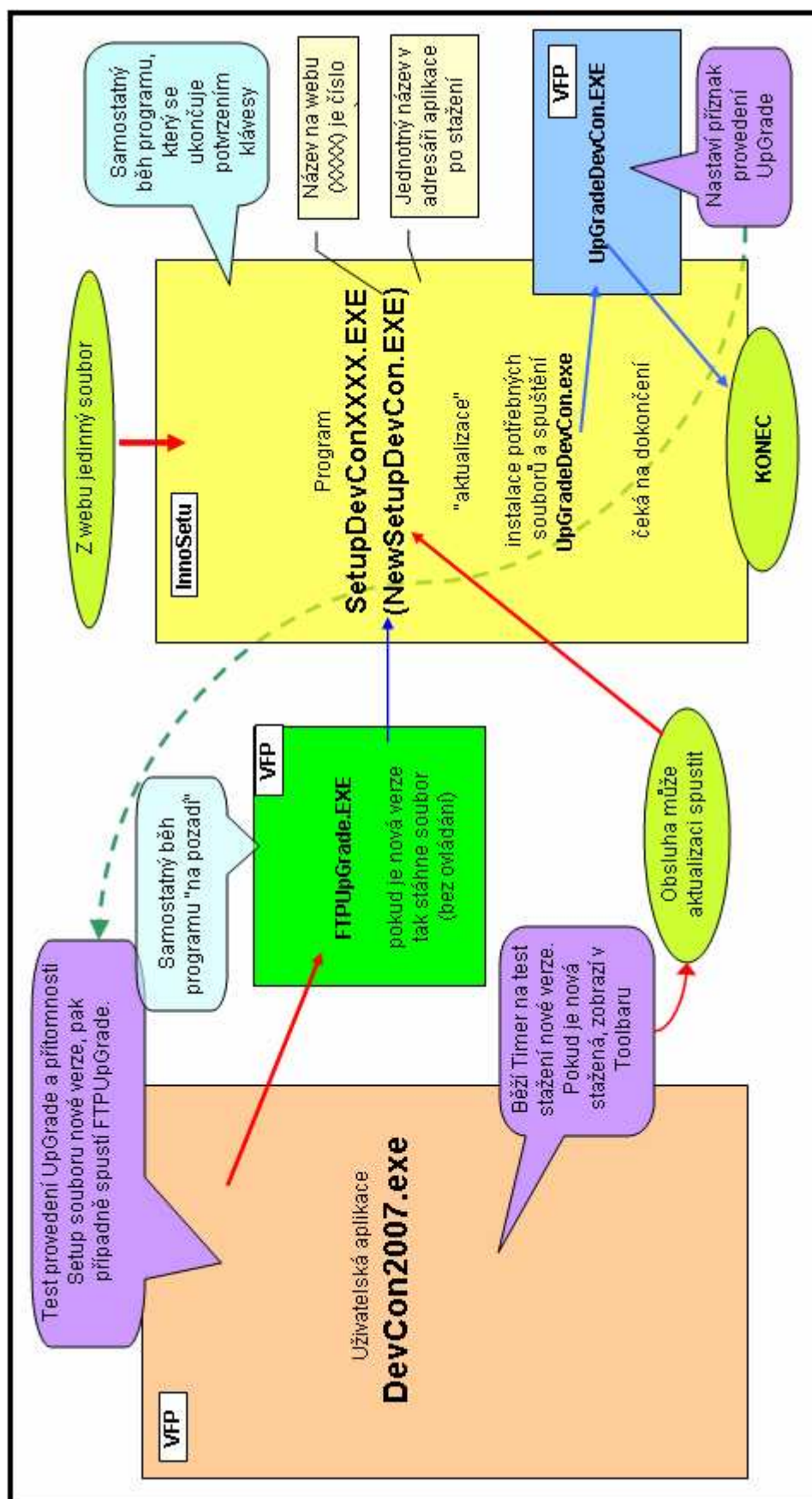
3.2. Popis programových modulů a jejich funkce

Jednotlivé programové moduly (celkem 4 samostatně běžící aplikace) mají na sebe návaznost, pro lepší orientaci v předloženém textu jsou shrnuty do následující tabulky. V ní je také poznamenáno, v čem je daný „EXE“ program vytvořen.

Název souboru	Jeho význam pro funkci Upgrade	Vývoj
SetupDevConXXXX.EXE	Soubor s aktualizací aplikace DevCon2007 stažený z FTP serveru – původní název (XXXX – číslo verze)	InnoSetup
NewSetupDevCon.EXE	Soubor s aktualizací aplikace DevCon2007 stažený z FTP serveru – přejmenovaný po stažení	InnoSetup
FTPUpGrade.EXE	Zjišťuje přítomnost nové verze programu	VFP
DevCon2007.EXE	Vlastní aplikace – bude se aktualizovat!	VFP
UpGradeDevCon.EXE	Aktualizační program, provádějící změny v aplikaci	VFP
UpGrade.inf	Pomocný soubor pro indikaci provedení UpGrade (testuje se)	VFP

Poznámky:

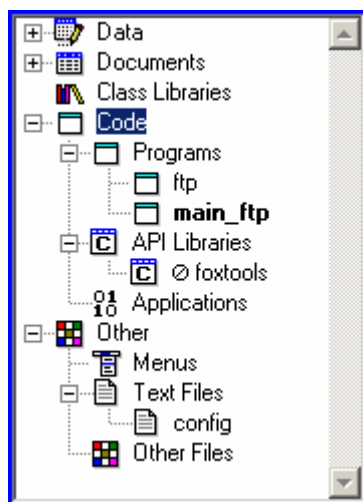
1. Oba programy vytvořené kompilací v prostředí InnoSetup jsou obsahově stejné, jen se různě jmenují, což je vysvětleno v dalším textu.
2. Jednotlivé výpisy programů obsahují odkazy [...] na místa kódu v předaných příkladech, kde jsou také tato čísla uvedena!



Obr. 3.1. Idea řešení automatického UpGrade aplikace ve VFP přes internet

3.2.1. Program FTPUpgrade.EXE

Je realizován v prostředí VFP a pracuje ve skrytém režimu (`_screen.hide()`). Jeho hlavním úkolem je otestovat přítomnost nové verze aktualizovaného programu na zvoleném serveru. Program, po nastavení potřebných počátečních podmínek, realizuje následující úkoly (Main_FTP.prg):



Project Manager FTPUpgrade

- Testuje přístup do internetu [**FTP01**], pokud je internet dostupný, pokračuje v činnosti, jinak skončí.
- Připojí se přes FTP k serveru s umístěnou aktualizací [**FTP02**], při neúspěšném přihlášení program skončí. Je-li přítomen soubor \$DWL\$.txt, připojí se do „testovacího“ adresáře.
- Stáhne informaci o verzi aktualizace [**FTP03**] a porovná ji s verzí uživatele [**FTP04**].
- Pokud je na FTP serveru novější aplikace, stáhne se aktualizací soubor [**FTP05**] a uloží se na disk uživatele (do pracovního adresáře), kde ji přejmenuje tak, aby ji běžící aplikace poznala.
- V případě nastaveného příznaku [**FTP06**] (volitelné v aplikaci) se vytvoří soubor s informacemi o provozu aplikace a uloží jej

Testuji existenci nové verze programu DevCon2007
 - aktuální verze programu: 14
 - síťový přístup není k dispozici
 Konec programu

Testuji existenci nové verze programu DevCon2007
 - aktuální verze programu: 9
 - FTP server připojen, přepínám adresář ...
 - přepnutí se povedlo ...
 - nová verze programu: 50
 - stahuji novou verzi SW ...

- aktuální verze programu: 9
 - FTP server připojen, přepínám adresář ...
 - přepnutí se povedlo ...
 - nová verze programu: 50
 - stahuji novou verzi SW ...
 - nová verze SW připravena ...
 - FTP spojení ukončeno ...
 Konec programu

na server.

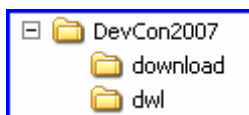
- Program se odpojí od serveru [**FTP07**] a ukončí svou činnost.

Přesto, že běh programu je realizován skrytě, lze doporučit, především pro období ladění, provádět informační výpisy do okna (nejlépe formou `wait window nowait`). Vzhledem k tomu, že těchto informací by mohlo být poměrně hodně a dopředu nevíme, kolik znaků se bude vypisovat, je vytvořená pomocná rutina `WAITWINDOW` [**FTP08**],

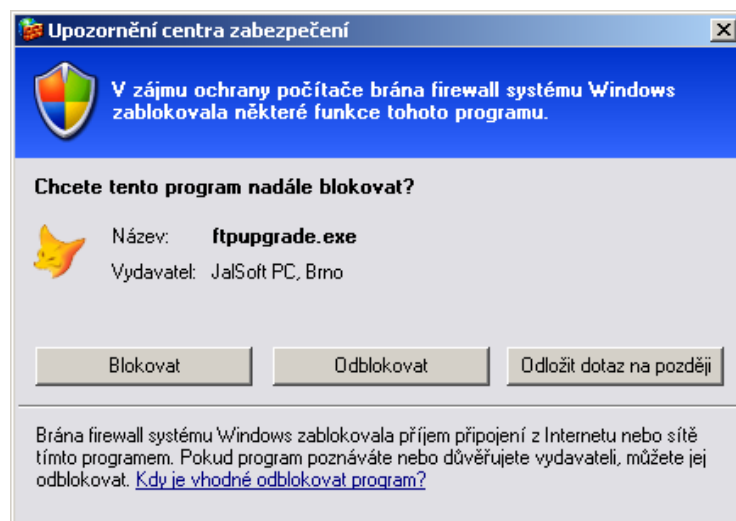
kteřá kontroluje délku vypisovaných údajů a případně začne umazávat řádky „historie“ zobrazovaného textu. Navíc je v této rutině i možnost [**FTP09**] úplného vypnutí zobrazení, jedinou změnou kódu.

Struktura programu **FTPUpgrade** je také bezprostředně vidět v okně Project Manager. Následující tabulka pak udává význam jednotlivých souborů.

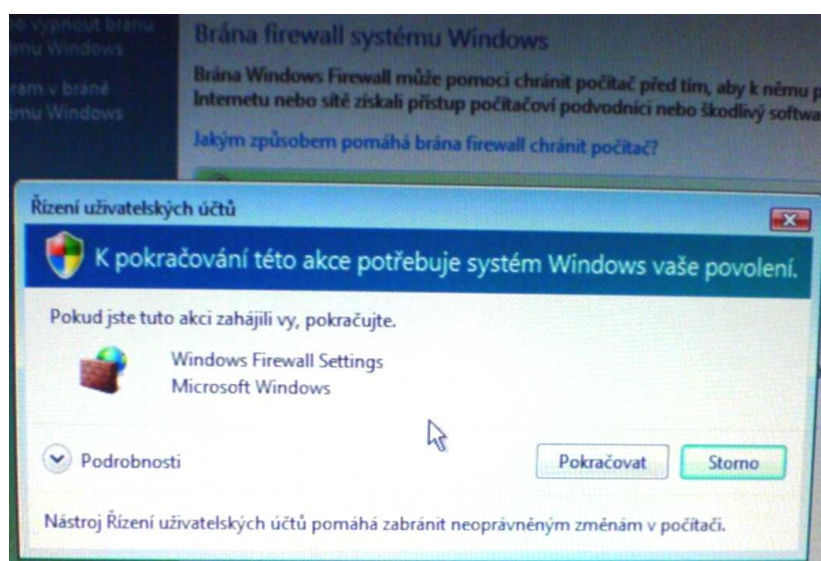
Název souboru	Jeho význam pro vlastní aplikaci	FTPUpgrade.EXE
FTUpgrade.PJX,PJT	Projekt aplikace pro kontrolu a stažení nové verze	
Main_FTP.prg	Hlavní program aplikace	
Config.fpw	Konfigurační soubor aplikace ve vztahu k VFP	
FTP.prg	FTP Services for Visual FoxPro	
\$DWL\$.txt	Textový soubor – pouze test přítomnosti	



Na FTP serveru je vhodné vytvořit strukturu dvou adresářů, z nichž jeden bude sloužit pro „testování“ nového upgrade a druhý pro ostrou verzi, tedy všechny uživatele. Pouze vy jako vývojáři budete vědět, jak se „automaticky“ připojit do testovacího adresáře (přítomnost souboru \$DWL\$.txt).



První spuštění programu **FTPUpgrade** ve většině případů a v OS Windows XP vyvolá hlášení brány firewallu (Centra zabezpečení) na upozornění přístupu z internetu. Zde je nutné povolit přístup, tedy „odblokovat“ bránu firewall. V OS Windows Vista se navíc zobrazí další hlášení na nejnižší úrovni zabezpečení a tu je nutné také povolit. V obou případech jde vždy pouze o první situaci při upgrade aplikace.



Hlášení z nejnižší úrovně Windows Vista nelze sejmout obrazovky, je proto vyfoceno!!

```
*****
*   MAIN
*   Hlavní program pro test existence nové verze aplikace
*   -----
*   autor:   doc.Ing. Rudolf Jalovecký, CSc., Brno, vyvoj@jalsoft.iol.cz
*
*****
_Screen.Hide()           && skrytý běh aplikace
LOCAL  cTxtWait          && proměnná pro texty hlášení
LOCAL  cNew_verze        && číslo nové verze (char)
LOCAL  nNew_verze        && číslo nové verze
LOCAL  nOld_verze        && číslo staré verze
LOCAL  MyUzivatelTXT     && soubor echa uživatele
LOCAL  MyText            && text do souboru
LOCAL  cFTPDirectory     && umístění Upgrade aktualizace
```



```

#DEFINE CRLF          CHR(13)
*                    přístup na server je nutno konkretizovat!
#DEFINE cFTPJmeno    "Rudolf"
#DEFINE cFTP Heslo    "Rudolf"
#DEFINE cFTPweb      "127.0.0.1"
#DEFINE cNewSoubor   "SetupDevCon*.EXE"
#DEFINE cNewSetupTMP "NewSetupDevConTMP.EXE"
#DEFINE cNewSetup    "NewSetupDevCon.EXE"
DIMENSION laDirArray[1,1]
laDirArray = ""
* ----- odkud bereme UpGrade
MyText="$DWL$.txt"
IF FILE("&MyText")
* ..... testovací
  cFTPDirectory="DevCon2007\DWL"
ELSE
* ..... ostrá
  cFTPDirectory=" DevCon2007\Download"
ENDIF
* ----- komentář do okna
cTxtWait="Testuji existenci nové verze programu DevCon2007"+CRLF
cTxtWait= WaitWindow(cTxtWait,0)
* ----- zjištění aktuální verze
set library to foxtools.fll additive
Public aFiles[12]
GetFileVersion("DevCon2007.EXE",@aFiles)
nOld_verze = VAL(SUBSTR(aFiles[4],AT(".",aFiles[4],2)+1))
set library to
RELEASE aFiles

* ----- komentář do okna
cTxtWait=cTxtWait+" - aktuální verze programu: "+STR(nOld_verze,4)+CRLF
cTxtWait= WaitWindow(cTxtWait,0)

* ----- Make it so Foxpro can find our FTP Class
LOCAL sz_ftp
SET PROCEDURE TO ftp.prg ADDITIVE
sz_ftp = CREATEOBJECT('ftp_service')
IF sz_ftp.OpenInternet(cFTPJmeno, cFTP Heslo, cFTPweb, "21")      && [FTP01]
* ----- komentář do okna
  cTxtWait=cTxtWait+" - FTP server připojen, přepínám adresář ... "+CRLF
  cTxtWait= WaitWindow(cTxtWait,0)
* ----- přepnutí do adresáře
  IF sz_ftp.ChangeFtpDirectory(cFTPDirectory)                    && [FTP02]
* ----- přepnutí do adresáře
    cTxtWait=cTxtWait+" - přepnutí se povedlo ... "+CRLF
    cTxtWait= WaitWindow(cTxtWait,0)
* ----- načteme soubory
    IF sz_ftp.GetFtpDirectoryArray(@laDirArray, cNewSoubor )    && [FTP03]
* ----- zde se musí poznat soubor s číslem a musí
* ----- se porovnat, zda je nová verze !!!
      FOR i= 1 TO ALEN(m.laDirArray, 1)
        cNew_verze =SUBSTR(m.laDirArray[i,1],AT(".",m.laDirArray[i,1])-4,4)
        IF ISDIGIT(cNew_verze)
* ----- pokud je zde číslo, máme upgrade
          nNew_verze = VAL(cNew_verze)
* ----- komentář do okna
          cTxtWait=cTxtWait+" - nová verze programu: "+STR(nNew_verze,4)+CRLF
          cTxtWait= WaitWindow(cTxtWait,0)
* ----- testujeme, zda je nova verze novejší
          IF nNew_verze > nOld_verze                               && [FTP04]
            cTxtWait=cTxtWait+" - stahuji novou verzi SW ..."+CRLF
            cTxtWait= WaitWindow(cTxtWait,0)
            * ----- && [FTP05]
            * ----- .F. - přepíše existující soubor !!
            IF sz_ftp.GetFtpFile(m.laDirArray[i,1],cNewSetupTMP,.F.)
              * přejmenovat staženy soubor!!
              IF FILE(cNewSetup)
                DELETE FILE cNewSetup
              ENDIF
              RENAME cNewSetupTMP TO cNewSetup
              cTxtWait=cTxtWait+" - nová verze SW připravena ..."+CRLF
              cTxtWait= WaitWindow(cTxtWait,0)
              && [FTP06]
            * ----- oznámení, kdo si stáhl data
            MyUzivatelTXT="Info_" +TTOC(DATETIME(),1)+".txt"
            MyText=SYS(0)+" "+TTOC(DATETIME())
          ENDIF
        ENDIF
      ENDFOR
    ENDIF
  ENDIF

```

```

        SET SAFETY OFF
        STRTOFILE(MyText,MyUzivatelTXT)
        SET SAFETY ON
        * ----- a ulozeni na FTP server
        sz_ftp.DeleteFtpFile(MyUzivatelTXT)
        sz_ftp.PutFtpFile(MyUzivatelTXT,MyUzivatelTXT)
        DELETE FILE "&MyUzivatelTXT"
    ELSE    && GetFtpFile
        cTxtWait=cTxtWait+" - stažení SW se nepovedlo ..."+CRLF
        cTxtWait= WaitWindow(cTxtWait,0)
    ENDIF    && GetFtpFile
ELSE    && new_verze
    cTxtWait=cTxtWait+" - není nová verze ..."+CRLF
    cTxtWait= WaitWindow(cTxtWait,0)
ENDIF    && new_verze
ENDIF    && isdigit
ENDFOR    && alen
ENDIF    && GetFtpDirectoryArray
cTxtWait=cTxtWait+" - FTP spojení ukončeno ..."+CRLF
ELSE    && ChangeFtpDirectory
    * -----                                přepnutí se nepovedlo
    cTxtWait=cTxtWait+CRLF+" - přepnutí se nepovedlo ..."+CRLF
ENDIF    && ChangeFtpDirectory
sz_ftp.CloseInternet()                                && [FTP07]
cTxtWait= WaitWindow(cTxtWait,3)
ELSE    && openInternet
    * -----                                spojeni se nepovedlo
    sz_ftp.CloseInternet()
    && chyba=sz_ftp.GetErrorcode(.T.)
    cTxtWait= cTxtWait+" - síťový přístup není k dispozici " +CRLF
    cTxtWait= WaitWindow(cTxtWait,0)
ENDIF    && openInternet

* -----                                komentář do okna
cTxtWait=cTxtWait+"Konec programu"
cTxtWait= WaitWindow(cTxtWait,0)

*****
*
*   Ulozeni potřebných dat před ukončením vlatní aplikace
*
*****
WaitWindow(cTxtWait,2)
RELEASE sz_ftp,cTxtWait,laDirArray
close databases all
*_Screen.Show()
*RETURN
QUIT

*****
*   Použití okna wait window na dlouhé texty (odmazává historii)
*       lze použít v okamžiku skrytého běhu VFP aplikace                [FTP08]
*****

PROCEDURE WaitWindow
PARAMETERS cVstup,nTime
*RETURN cVstup                                && bez echa do okna    [FTP09]
IF LEN(cVstup)>=255
    cVstup=SUBSTR(cVstup,AT(CRLF,cVstup)+1,255)
ENDIF
IF nTime = 0
    WAIT WINDOW NOWAIT cVstup
ELSE
    WAIT WINDOW TIMEOUT nTime cVstup
ENDIF
RETURN cVstup
*****

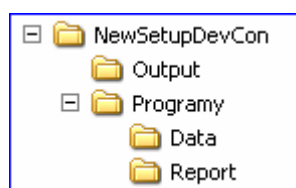
```

3.2.2. Program NewSetupDevCon

Instalační program NewSetupDevCon je sestaven v programu INNOSETUP [2]. V tomto instalačním programu je nutné sestavit řetězec instalací nových nebo inovovaných souborů, umístěných do potřebných adresářů a na závěr spustit program UpGradeDevCon, který zajistí potřebné změny v aplikaci (ve smyslu foxových konvencí). Po jeho ukončení je instalační program ukončen a také je zvládnuta automatická aktualizace aplikace. Velkou řadu informací, jak realizovat celou instalaci foxovské aplikace jen s pomocí programu INNOSETUP lze najít v [1], i přesto si Vám v této kapitole dovoluji představit, jaké úseky instalačního programu je nutné využít, ať již je základní instalace uživatelské aplikace libovolná.

Obsah instalačního programu (NewSetupDevCon.iss) je v podstatě klasický, jen v něm nejsou zastoupeny všechny sekce, neboť není potřebné, aby tento program příliš moc věcí měnil či doinstalovával. Komentář si zaslouží:

- Oblast [**Setup**] řádek „OutputBaseFilename“ je název výstupního instalačního souboru a podle zavedených konvencí tohoto příspěvku je nutné mít na konci čtyři číslice, udávající verzi nové aktualizace.
- Oblast [**Files**] obsahuje seznam vložených souborů pro aktualizaci. Podle uvedeného komentáře nesmí chybět soubory UpGradeDevCon.EXE, DevCon2007.EXE. Aby bylo možné mazat pracovní a dále již nepoužívané soubory je vhodné do TMP adresáře nainstalovat soubor DeleteMe.EXE a s ním pak v části [Run] pracovat. Zde také musí být umístěny všechny další nové soubory, které aktualizovaná aplikace bude používat. Pro přehlednost je vhodné si tyto soubory členit do adresářů (Data, Report, Label, Library a další) s tím, že k uvedeným souborům právě v této části instalačního programu uvedete správné cesty. Nezapomeňte, že řada reportů či labelů obsahují dva soubory.
- Oblast [**Run**] obsahuje prakticky pouze dva příkazy. První spouští program UpGradeDevCon na úpravu prostředí aplikace dle foxovských konvencí, přičemž příznak „waituntilterminated“ čeká na dokončení tohoto programu. Druhý příkaz pak slouží na vymazání právě skončeného programu.



Struktura adresářů pro projekt v InnoSetupu

Pro uvedený příklad je vhodné mít připravenou strukturu adresářů podle obrázku. Z adresáře „Program“ a jeho podadresářů se získávají všechny soubory pro instalaci aktualizace a do adresáře „Output“ je pak výsledný aktualizací soubor „NewSetupDevCon.EXE“ uložen.

Název souboru	Jeho význam pro vlastní aplikaci	NewSetupDevCon.EXE
NewSetupDevCon.iss	Projekt instalace automatické aktualizace pro InnoSetup	
UpGradeDevCon.EXE	Pomocný soubor, upravující stávající aplikaci na novou verzi	
DevCon2007.EXE	Nová verze aktualizované aplikace	
DeleteMe.EXE	Program pro mazání souboru v příkazové řádce	
Informace.txt	Textový soubor s informacemi o nové verzi aplikace (není povinný)	

```
; -----  
;   UpGrade programu DevCon2007       (doc.Ing.Rudolf Jalovecky,CSc.)  
; -----  
; - relativní cesta ke všem produktu, nezávisle na disku a umístění  
; - využívá strukturu adresářů předváděného příkladu  
; -----  
  
[Setup]  
AppName           =NewSetupDevCon  
AppVerName        =NewSetupDevCon 2007.01  
AppPublisher      =doc.Ing.Rudolf Jalovecký,CSc.  
AppPublisherURL   =http://www.jalsoft.iol.cz  
AppSupportURL     =http://www.jalsoft.iol.cz  
AppUpdatesURL     =http://www.jalsoft.iol.cz/sw_vfp/DevCon2007  
DefaultDirName    ={pf}\DevCon2007  
OutputBaseFilename=SetupDevCon0010  
Compression       =zip  
SolidCompression  =yes  
AlwaysRestart     =NO  
  
[Files]  
;   mazání UpGradeDevCon.EXE po ukončení práce  
Source: "Programy\DeleteMe.exe";      DestDir: "{tmp}"  
;   aktuální programy  
;   - aktualizace pro změny prostředí aplikace  
Source: "Programy\UpGradeDevCon.EXE";  DestDir: "{app}"; Flags: ignoreversion  
;   - nova verze aplikace  
Source: "Programy\DevCon2007.EXE";     DestDir: "{app}"; Flags: ignoreversion  
;   aktuální informace  
Source: "Programy\Informace.txt";      DestDir: "{app}"; Flags: ignoreversion  
;   nové (upravené) databaze...  
;Source: "Programy\Data\....";        DestDir: "{app}\Data"; Flags: ignoreversion  
;   všechny reporty  
;Source: "Programy\Report\....";      DestDir: "{app}\Report"; Flags: ignoreversion  
;   a další potřebné soubory, knihovny apod.  
  
[Run]  
; Spuštění programu UpGradeDevCon a úpravy prostředí aplikace dle foxovské konvence  
Filename: "{app}\UpGradeDevCon.EXE";  StatusMsg: "Provádím UpGrade dat!";  
                                           Flags: shellexec waituntilterminated  
; Mazání programu UpGradeDevCon  
Filename: "{tmp}\DeleteMe.exe";       Parameters: "{app}\UpGradeDevCon.EXE";  
                                           StatusMsg: "Provádím UpGrade dat"  
  
; ----- END
```

3.2.3. Program UpGradeDevCon

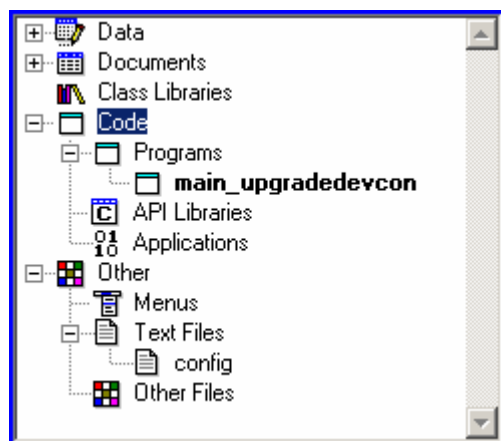
Program UpGradeDevCon je napsán ve VFP a zajišťuje veškeré potřebné změny v aktualizované aplikaci, tedy především změny struktury tabulek a databází podle foxovských konvencí. Samozřejmostí je, že programové změny nesmí způsobit ztrátu dat v těchto tabulkách.

Projekt manager může obsahovat jen jeden *.prg soubor, v němž se vše připraví a provede. Předváděný příklad v tomto případě nic s databázemi nedělá a obsahuje nutné části programu:

- Úvodní část pro definici prostředí dle potřeby [[UPG01](#)].
- Nezbytný test na spuštěnou aplikaci [[UPG02](#)], jinak by při změnách databází mohl vzniknout problém sdílení souborů.
- Oblast [[UPG03](#)], kde se vše odehraje, za předpokladu uvedeného v upozornění.
- Oblast [[UPG04](#)], která zajistí informovanost aktualizované aplikace, že byl upgrade proveden např. paměťovou proměnnou s následným uložením do souboru.
- Slušností je ohlásit, že program proběhl bez závad [[UPG05](#)].

Zde si dovolím drobné upozornění.

Protože vývojář nemůže dopředu vědět, zda si uživatel stáhl předchozí aktualizaci aplikace, musí být rutiny pro úpravu struktury tabule a databází napsány tak, aby na konci tohoto programu měly tabulky požadovanou strukturu, bez ohledu jakou měly před spuštěním programu UpGrade. Také při kopírování nových souborů je nutné mít na paměti, že už je může uživatel používat a modifikovat (tedy testovat jejich existenci před přepsáním). Tento problém je možné vyřešit např. tím, že instalační program NewSetupDevCon bude obsahovat požadovaný soubor, ale přejmenovaný na dočasný (např. TMPMojeDatabase.dbf) a program UpGradeDevCon provede test existence souboru (MojeDatabase.dbf) a pokud už je, tak vymaže dočasný soubor (TMPMojeDatabase.DBF) a pokud ještě není, tak přejmenuje dočasný soubor na správný název.



Project Manager UpGradeDevCon

Název souboru	Jeho význam pro vlastní aplikaci	UpGradeDevCon.EXE
UpGradeDevCon.PJX,PJT	Projekt aplikace pro úpravy databází aktualizované aplikace	
Main_UpGradeDevCon.prg	Hlavní program aplikace	
Config.fpw	Konfigurační soubor aplikace ve vztahu k VFP	
UpGrade.inf	Pomocný soubor pro indikaci provedení upgrade (vytváří se)	

```

*****
*
*   Upgrade programu DevCon2007
*
*****

WITH _Screen                                && [UPG01]
  .Hide()                                    && schováme VFP okno
  .WindowState = 0                          && normální velikost
  .AutoCenter = .T.                         && dprostřed
  .Caption    = "UpGrade programu DevCon2007" && hlavička
  .BackColor  = RGB(168,168,168)           && pozadí
ENDWITH

PUBLIC NameEXE,NameAppEXE
NameUpGradeEXE = 'UpGradeDevCon'
NameAppEXE     = 'DevCon2007.EXE'

*****
*
*   Definice prostředí VFP ... dle potřeby
*
*****

_Screen.show()

*****
*   cykl s testem na ukončení programu DevCon2007, aby
*   mohl být realizován UpGrade
*****
DO WHILE .T.                                && [UPG02]
  pom = FOPEN('NameAppEXE',12)             && otestujeme otevření programu
  =FCLOSE(pom)                             && zavřeme co nejdříve
  IF pom < 0                                && program běží
    IF 2=MESSAGEBOX("Program DevCon2007 je spuštěn!"+CHR(13)+;
                    "Nelze provést UpGrade!",5+16," P O Z O R !")
      CANCEL
    ENDIF
  ELSE
    EXIT                                    && můžeme pokračovat
  ENDIF
ENDDO

*****
*   Provádění změn struktury tabulek dle potřeby aktualizované verze
*****
*   UpGrade z xx.xx.2007 && komentář
*
* -----
*   .....                                && [UPG03]
*   .....
WAIT WINDOW TIMEOUT 1 "Provádím úpravy databází..."

WAIT WINDOW TIMEOUT 1 "Provádím úpravy struktury ..."

*****
*   Informace o spuštěném aktualizacním programu
*****
IF FILE("UpGrade.inf") && [UPG04]
  DELETE FILE "UpGrade.inf"
  SET MESSAGE TO " "
ENDIF
STRTOFILE("Proveden Upgrade ", "UpGrade.inf")

&& [UPG05]
MESSAGEBOX("UpGrade programu DevCon2007 byl proveden bez problémů!",0+32,;
           "Informace!",2000)
QUIT

```

3.2.4. Chování aktualizované aplikace DevCon2007

Aktualizovaná aplikace obsahuje poměrně málo nutných programových sekvencí pro realizaci automatické aktualizace aplikace. Uvedený příklad aplikace DevCon2007 nedělá vůbec nic, jen se spustí, zobrazí svoje číslo verze a čeká na ukončení. Do této aplikace jsou dopsány úseky programu,

keré zajistí funkčnost automatické aktualizace. Znamená to, že kdykoliv uživatel spustí uvedený program, provedou se činnosti, které bezprostředně směřují k zjištění nové verze, její stažení, ohlášení uživateli a po jeho rozhodnutí spuštění celého aktualizacího procesu.

V úvodu aplikace, tedy v době, kdy se aplikace spouští a připravuje si své prostředí, se doplní hlavní program krátkou sekvencí, která má za úkol (Main_Devcon.prg):

- Otestovat, zda proběhla od minulého spuštění programu aktualizace [DC01] (existence souboru UpGrade.inf).
- Pokud aktualizace proběhla, vymaže aktualizací i testovaný soubor a dále se program zabývá svými úkoly [DC02].
- Pokud aktualizací soubor není, tak spustí externí program FTPUpgrade [DC03].
- Zjištění verze programu [DC04] je spíše formální pro tento příklad, nemá žádnou další návaznost na funkčnost automatické aktualizace.

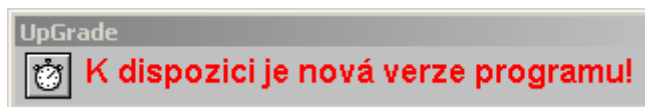
Jako způsob zobrazení uživateli, že má k dispozici novou verzi programu, byl zvolen samostatný Toolbar, který se zviditelní pouze v okamžiku přítomnosti souboru s novou verzí programu. Použití samostatného Toolbaru má několik důvodů:



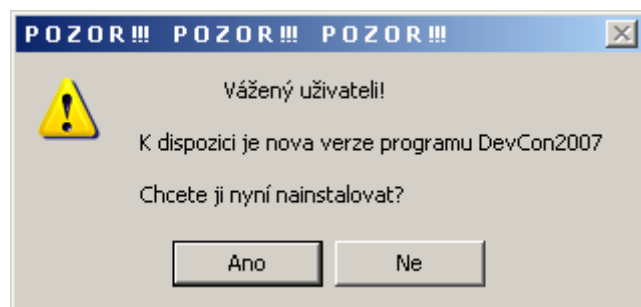
- Prakticky při jakékoliv činnosti je Toolbaru vidět a tak má uživatel tuto informaci k dispozici ihned jakmile dojde k identifikaci nové verze programu.
- Uživatele tato informace nikterak nezatíží v podobě nutného odkliknutí okna s hlášením, že je nová verze.

Ještě v hlavním programu (Main_Devcon.prg) se dopíše část programové sekvence:

- Definice Toolbaru [DC05] jako objektu, na němž je Timer a Label.
- Pokud nebyla aktualizace realizována, program zjistí, zda je přítomen aktualizací soubor a pokud ano, tak uživateli ohlásí, že má k dispozici aktualizaci [DC06] a zastaví časovač pro aktivaci pozdějšího testu, pokud ne, tak se Toolbar nezobrazí.

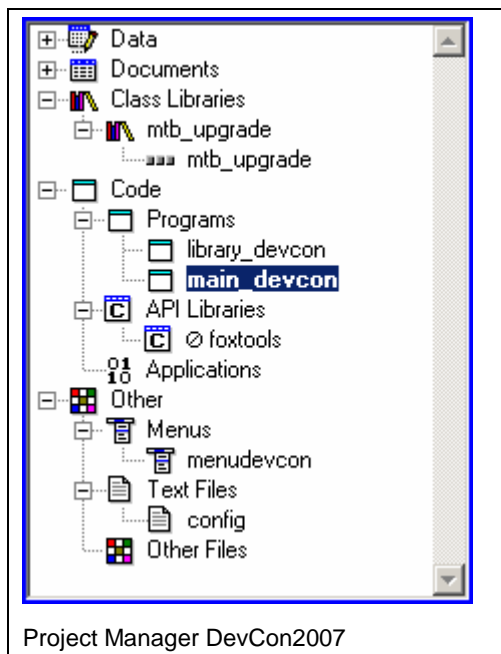


Nyní vlastně celá aplikace zcela normálně pracuje a po nastaveném čase v časovači Toolbaru (MTB_UpGrade.Timer_UpGrade.Timer) [DC07] se spustí činnost, která po uplynutí nastavené doby realizuje test na existenci aktualizacího souboru a při jeho existenci následně zobrazí uživateli hlášení o nové verzi.



Činnost programu v části Toolbaru po kliknutí na text „K dispozici je nová verze programu!“ zajistí dotaz a podle odpovědi i případné spuštění aktualizacího programu (MTB_UpGrade.lblMessage.click) [DC08].

Je samozřejmé, že součástí této programové sekvence musí být volaná procedura, která řádně ukončí běh aktualizované aplikace.



Název souboru	Jeho význam pro vlastní aplikaci	DevCon2007.EXE
DevCon2007.PJX,PJT	Projekt vlastní aplikace	
Main_DevCon.prg	Hlavní program vlastní aplikace	
MenuDevCon.mpr	Menu vlastní aplikace	
Config.fpw	Konfigurační soubor aplikace ve vztahu k VFP	
Library_DevCon.prg	Knihovna funkcí a procedur vlastní aplikace	
MTB_UPGRADE.VCX,VCT	Třída Toolbaru pro zobrazení informace o nové verzi	
FoxTools.fll	Knihovna VFP	

```

*(Main_DevCon.prg)

*****
*   Test na existenci nove verze programu
*****
#IF .T.
DECLARE INTEGER ShellExecute IN shell32.dll ;
    INTEGER hndWin, STRING cAction, STRING cFileName, ;
    STRING cParams, STRING cDir, INTEGER nShowWin
* ----- pokud proběhl UpGrade programu, tak smažeme Setup soubor
IF FILE("UpGrade.inf")                                && [DC01]
    DELETE FILE "UpGrade.inf"                          && [DC02]
    DELETE FILE "NewSetupDevCon.EXE"
ENDIF
* ----- vždycky spustíme kontrolu a stažení
IF NOT FILE("NewSetupDevCon.EXE")
    ShellExecute(0,"open","FTPUpGrade.EXE","",",",1)  && [DC03]
ENDIF
#ENDIF
*****
*   Zjištění verze programu                                && [DC04]
*****

```



```

set library to foxtools.fll additive
local aFiles[12]
GetFileVersion("DevCon2007.EXE",@aFiles)
PUBLIC SW_Verze
SW_Verze= aFiles[4]
RELEASE aFiles
RELEASE LIBRARY foxtools.fll
*****

```

```

*****
*   Definice a případné zobrazení Toolbaru s informacemi
*****
set classlib to MTB_UpGrade
MTBUpGrade=createobj('MTB_UpGrade')           && [DC5]

MTBUpGrade.Dock(0,INT(_screen.Width-MTBUpGrade.width),0)
* podle stavu
IF FILE("NewSetupDevCon.EXE")                 && [DC06]
    MTBUpGrade.Show()
    MTBUpGrade.Timer_Update.enabled = .F.
ELSE
    MTBUpGrade.hide()
ENDIF
*****

```

```

(MTB_UpGrade.Timer_UpGrade.Timer)

*****
*   Test na novou verzi programu                 && [DC07]
*****
IF FILE("NewSetupDevCon.EXE")
    MTBUpGrade.Show()
ENDIF
* v tomto běhu programu už nová verze nebude!!!
this.Enabled = .F.

```

```

(MTB_UpGrade.lblMessage.click)

*****
*   Informace uživateli o nové verzi a případné
*   spuštění aktualizacího programu             && [DC08]
*****
IF 6=MESSAGEBOX("                Vážený uživateli!"+CHR(13)+CHR(13)+;
                "K dispozici je nova verze programu DevCon2007  "+CHR(13)+CHR(13)+;
                "Chcete ji nyní nainstalovat?",4+48," P O Z O R !!!   P O Z O R !!!   P O Z
O R !!!")
    *   spustime instalaci
    DECLARE INTEGER ShellExecute IN shell32.dll ;
                INTEGER hndWin, STRING cAction, STRING cFileName, ;
                STRING cParams, STRING cDir, INTEGER nShowWin

    MTBUpGrade.Hide()
    ShellExecute(0,"open","NewSetupDevCon.EXE","",",",1)
    RELEASE WINDOWS   && umi jen jedno aktivni okno
    CLEAR EVENTS
ENDIF

```

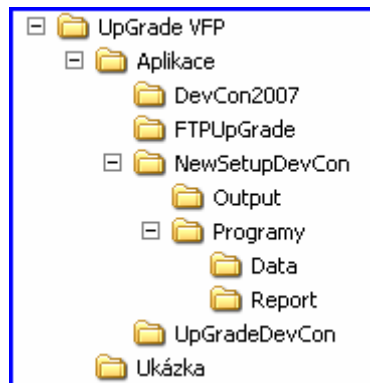
3.3. Návaznost modulů aneb jak uvést celý automatický upgrade do provozu

Podle uvedených popisů jde o několik vzájemně provázaných programů, jejich zprovoznění může činit drobné problémy, neboť v řadě míst na sebe navazují nebo se podmiňují. Proto doporučuji přistoupit k programové realizaci (doplnění stávající aplikace) následujícím způsobem:

1. Sepsat si názvy nových programů a jejich vzájemnou souvislost (např. s využitím obr.3.1.).
2. Připravit si FTP přístup na server (adresa, přístupové jméno a heslo) a na něm vytvořit adresář pro novou verzi programu. Pro úplně první testování si na něj umístnit malý EXE program, který bude mít název v konvencích uvedených v tomto popisu, tedy především čtyři číslice před tečkou i kdyby prozatím to byly tři nuly a jedna číslice (v našem příkladu SetupDevConXXXX.EXE).
3. Upravit a zprovoznit program na stahování nové verze z FTP serveru (pro nás FTPUpGrade.EXE). Při jeho ladění pouze odzkoušíme otestování nové verze na FTP serveru a případně její stažení.
4. Začlenit nové funkce na stahování nové verze do stávající aplikace (pro nás DevCon2007.EXE) a její odladění. Znamená to odzkoušení, zda se spouští externí program FTPUpGrade, zda se stahuje nový soubor, zda se po určité době (podle časovače) ohlásí přítomnost nové verze a zda se tato dá spustit (i když pro etapu ladění vlastně spustíme jiný program, který s naší celou strukturou nemá mnoho společného).
5. Sestavit instalační program v prostředí InnoSetup (NewSetupDevCon). Patří sem:
 - Příprava adresářové struktury instalačního programu.
 - Zkopírování všech souborů, které tvoří sestavu automatického upgrade aktualizované aplikace.
 - Zkopírování souborů, které budou nové v aktualizované aplikaci.
 - Sestavení zdrojového textu kompilace *.iss.
 - Přeložení a odzkoušení funkčnosti (kontrola, zda se všechny soubory doopravdy dostanou do aktualizované aplikace)
6. Zavést projekt aktualizacího programu UpGradeDevcon. Sestavit a odladit všechny nutné struktury programu tak, aby prováděla požadované změny v databázích a dalších souborech dle potřeby nové verze aktualizované aplikace.
7. Sestavit komplet projekt automatické aktualizace v jednom adresáři, se správnými názvy souborů a odladit celý cyklus aktualizace na „lokálním“ serveru, následně pak i na serveru v internetu. Otestovat si schopnost „výběru“ ostrého a testovacího adresáře na serveru v internetu.
8. Doporučení (z vlastní zkušenosti) - přenést výslednou sestavu automatické aktualizace na PC, kde není instalované vývojové prostředí, ale byla instalována aplikace, která s ebude aktualizovat.

4. Komentář k předaným souborům

V archivním souboru UpGrade.ZIP jsou umístěny všechny ukázkové soubory, které jsou v tomto dokumentu použité. Po rozbalení se Vám vytvoří adresářová struktura podle obrázku, kde v základním adresáři jsou vytvořeny dva podadresáře („Aplikace“, „Ukázka“).



Adresářová struktura příkladu na funkci automatického upgrade aplikace

Adresář „**Aplikace**“ obsahuje v dalších čtyřech adresářích všechny ukázkové programy (jejich vývojové podoby), s nimiž lze automatický upgrade realizovat. Názvy adresářů jsou shodné s názvy aplikací, které obsahují a které jsou také v tomto textu komentovány.

Adresář „**DevCon2007**“ obsahuje „malou“ aplikaci, kterou teoreticky aktualizujeme. Tato ukázková aplikace obsahuje jen ty nejnnutnější funkce, aby fungovala (otevřelo se okno) a bylo možné na ni činnost aktualizace předvést.

Adresář „**FTPUpgrade**“ obsahuje úplný FTP přístup k definovanému místu v internetu (pro ukázkou je zvolen vlastní počítač s adresou 127.0.0.1, na němž musí být vytvořen „lokální“

server).

Adresář „**NewSetupDevCon**“ obsahuje kompilační program pro sestavení souboru s aktualizací v programu InnoSetup a samozřejmě všechny „nové“ soubory, které chceme do aktualizace zahrnout.

Adresář „**UpGradeDevCon**“ obsahuje malou VFP aplikaci, která zajistí požadované změny struktury nebo obsahu aktualizované aplikace.

Adresář „**Ukázka**“ obsahuje ukázkovou aplikaci se všemi nutnými soubory pro demonstraci chodu automatického upgrade (připojuje se na server 127.0.0.1 s přihlašovacími právy podle textu v kap. 3.2.1)

5. Literatura a odkazy na web

- [1] Doug Hennig, Installing Applications Using Inno Setup, 13th European Visual FoxPro DevCon 2006
- [2] Stránky programu InnoSetup www.jrsoftware.org