

# Lotus Script: Domino objekty

---

## 2. DIEL



Základné postupy využitia jazyka Lotus Script pri programovaní aplikácií pre systém Lotus Domino/Notes.

Miro Uhlár, 2010

---



## Úvod

V 1. časti seriálu sme sa zorientovali v problematike LotusScriptu (LS), vyskúšali si používanie Domino Designera a LS Debuggera pri tvorbe a odľadňovaní LS sekvencií. Dnes sa pozrieme na základné stavebné prvky LS i jeho využitie pri sprístupnení Notes databázových prvkov.

## PREMENNÉ

Začnime základným stavebným prvkom kódov – premennou. Ide to pamäťovú hodnotu, ktorú je možné meniť počas behu kódu meniť. Premenná je určená:

- Názvom (nezáleží na veľkosti znakov, v názve však nesmie byť medzera)
- Dátovým typom (napr. String - reťazec znakov, Integer – celé čísla, Double – desatinné čísla,...)

Samotná deklarácia (teda priradenie dátového typu premennej) môže byť:

- Explicitná - príkazom *DIM* názov *AS* dátový typ
- Implicitná – uvedením definovanej prípony v názve premennej (napr. LName\$ priradí premennej Lname dátový typ String)

### TIPY:

- Explicitná deklarácia premennej nie je povinná (len v prípade použitia direktívy *Option Declare* v danom kóde) a môže sa vyskytovať na ľubovoľnom mieste scriptu (pred použitím premennej).
- Hodnota sa do premennej priraduje znakom = (rovná sa).
- Textový reťazec je uzatvorený v úvodzovkách "... " (alebo |...|, resp. {...}).

## PRAX: PREMENNÉ

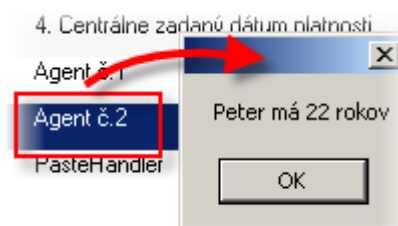
Využitie deklarácie premenných si vyskúšame na nasledujúcom kóde (vytvorte agenta č.2 analogicky k č.1 a vložte kód):

```
Dim Vek As Integer
```

```
Vek=22
```

```
Meno$="Peter"
```

```
MessageBox Meno$+" má "+Cstr(Vek)+" rokov|
```



Vysvetlenie kódu:

1. riadok: Explicitná deklarácia premennej *Vek*
2. riadok: Priradenie hodnoty do premennej *Vek*
3. riadok: Implicitná deklarácia premennej *Meno* s priradením hodnoty
4. riadok: Zobrazenie textového reťazca, ktorý vznikol zložením z viac častí, pričom funkcia *Cstr* konvertuje číselnú hodnotu do reťazca

K problematike premenných logicky patria aj polia (*Array*), zoznamy (*List*) či triedy (*Class*). Vzhľadom k limitovanému rozsahu seriálu sa týmto prvkom venovať nebudeme, rozhodne sú však potrebné pri „väčšejšom“ využití LS.

## VETVENIE KÓDU

LS poskytuje pre vetvenie kódu nasledovné základné možnosti:

Spôsob	Vysvetlenie
IF...THEN...ELSE	Klasické vetvenie kódu do 2 vetiev (splnená resp. nesplnená podmienka)
IF...THEN...ELSEIF...THEN....	Viacnásobné vnorenie vetvenia kódu
IF...GOTO...ELSE	Odskok kódu na definované návěstie v prípade splnenia podmienky
SELECT CASE...	Vetvenie kódu do viacerých vetiev

Bližšie syntaktické detaily nájdete v príručke k Domino Designeru (DD).

## CYKLY V kóde

Opakované vykonávanie kódu je možné v LS zrealizovať nasledovnými spôsobmi:

Spôsob	Vysvetlenie
FOR...TO...STEP NEXT	Klasické opakovanie cyklu od – do. Cyklus je možné opustiť príkazom EXIT FOR (pokračuje za príkazom NEXT) alebo GOTO (pokračuje sa na definovanom návěstí).
FORALL...IN... END FORALL	Spracovanie jednotlivých prvkov poľa. Tento cyklus sa často využíva pri spracovaní jednotlivým hodnôt v tzv. „multi-value“ poliach Notes dokumentu.
WHILE <i>podmienka</i>	Cyklus sa opakuje kým je splnená podmienka, definovaná pri vstupe.

WEND	
DO LOOP	Nekonečné opakovanie, ktoré sa opúšťa príkazom EXIT DO. Tento cyklus má aj modifikácie s podmienkou na začiatku či konci.

Cykly i vetvenie využijeme v ďalších LS kódach, syntaktické detaily sú uvedené v DD príručke.

## Domino OBJEKTY

Hlavný zmysel využitia LS v prostredí Domino/Notes systému je v prístupe k tzv. „Domino objektom“. Tieto objekty popisujú vlastnosti a metódy rozličných častí prostredia (databáza, ACL, pohľad, otvorený dokument,...) a delia sa do 2 základných kategórií:

- FRONT-END (základom je *NotesUIWorkspace*)
- BACK-END (základom je *NotesSession*)

**TIP:** Celý diagram (teda zoznam objektov aj so vzájomnými vzťahmi) je dostupný v úvodnej stránke DD (home page). Z tohto miesta je zároveň možné preskúmať ich všetky dostupné vlastnosti a metódy.

Vytvorenie novej inštancie triedy sa realizuje dvomi spôsobmi :

- Priamo pri deklarácii: *DIM* premenna *AS NEW* trieda
- Po deklarácii: *SET* premenna = *NEW* trieda

Sprístupnenie vlastností či metód tried sa realizuje cez „bodku“: premenna.Vlasnost

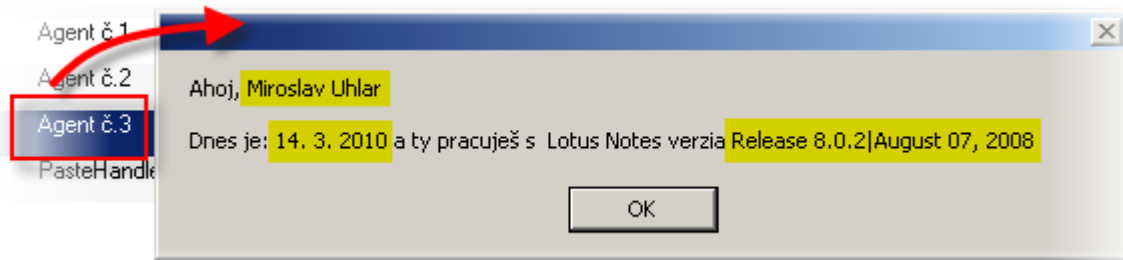
## PRAX: NOTESSESSION

Trieda *NotesSession* je základom „back-end“ tried a umožňuje prístup k aktuálne bežiacemu kontextu scriptu. Jej využitie si overíme na nasledujúcom kóde (vytvorte agenta č.3):

```
Dim session As New NotesSession
```

```
MessageBox "Ahoj, "+session.CommonUserName+" "+Chr(13)+Chr(13)_
```

```
+"Dnes je: "+ Cstr(Today)+" a ty pracuješ s Lotus Notes verzia "+session.NotesVersion
```



1. riadok: Explicitná deklarácia inštancie triedy *NotesSession*
2. riadok: Zobrazenie informácií z aktuálneho kontextu, teda:
  - Meno aktuálne prihláseného používateľa (*CommonUserName*)
  - Dnešný dátum z funkcie *Today* konverziou na textový reťazec (*Cstr(Today)*)
  - Verziu klienta Lotus Notes (*NotesVersion*)

## PRAX: NOTESDATABASE

Trieda *NotesDatabase* predstavuje „back-end“ Notes databázu a dá sa odvodiť priamo z aktuálneho kontextu - *NotesSession*. Nižšie uvedený kód použite na vytvorenie agenta č.4:

```
Dim session As New NotesSession
```

```
Dim db As NotesDatabase
```

```
Set db = session.CurrentDatabase
```

```
MessageBox |Názov databázy: "| & db.Title & "|.
```

```
MessageBox |Cesta k databáze: "| & db.FilePath & "|.
```

```
MessageBox |Veľkosť databázy: | & Format(db.Size,"#,#" & | [B].|
```

1. riadok: Explicitná deklarácia inštancie triedy *NotesSession*
2. riadok: Deklarácia triedy *NotesDatabase*
3. riadok: Vytvorenie inštancie pre aktuálnu datbázu (*CurrentDatabase*)
4. riadok: Zobrazenie názvu databázy (*Title*)
5. riadok: Zobrazenie plnej cesty k databáze (*FilePath*)
6. riadok: Zobrazenie sformátovanej veľkosti databázy (*Size*)

Všimnite si, že časti textového reťazca je možné spájať aj pomocou znaku &.

## Zistenie adresných kníh

---

LS v spojení s NotesSession môže pomôcť odpovedať aj na otázku: „Aké adresné knihy aktuálne využíva môj Notes klient?“ (Upresnenie tejto informácie je často riešením mnohých problémov, súvisiach s rozoznávaním adries pri e-mail správach). Nižšie uvedený script uložte do agenta č. 5:

```
Dim session As New NotesSession
```

```
Dim ABooks As Variant
```

```
ABooks = session.AddressBooks
```

```
Forall db In Abooks
```

```
    Call db.Open( "", "" )
```

```
    If db.Server="" Then
```

```
        MessageBox "Lokálna: " + db.Title
```

```
    Else
```

```
        MessageBox "Server db: " + db.Title
```

```
    End If
```

```
End Forall
```

Kód tentokrát tentokrát nebude vysvetlený, to už bude vašou „domácou“ úlohou.