

Lotus Script: Základy

1. DIEĽ



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

Miro Uhlár, 2010



Úvod

IBM Lotus Domino/Notes. Software, ktorý je od roku 1996 strategickou súčasťou IBM portfólia a v súčasnosti „de-facto“ štandard pre „collaboration“ systémy. K jeho silným stránkam patrí aj možnosť rýchleho vývoja aplikácií. Práve preto sme na túto tému počas roka 2006 publikovali 10 dielny seriál. V jednotlivých častiach sme odhalili jednotlivé komponenty Notes aplikácie a zároveň vytvorili jednoduchý dokument management systém (DMS).

Dnes pokračujeme využitím ďalšieho, mimoriadne silného programátorského prostriedku: LotusScriptu (LS). Počas 4 pokračovaní ukážeme jeho možnosti pri vylepšovaní Notes aplikácií, pričom sa v záverečnej časti zmienime aj o ďalších moderných technológiach, ktoré pripravila IBM pre platformu Lotus Notes.

Seriál nemá ambíciu a priestor, aby poučal skúsených programátorov LS. Naopak, je primárne určený pre tých, čo:

- Zvládajú základy vývoja aplikácii v Lotus Notes, ale LS je pre nich veľkou neznámou,
- Poznajú Visual Basic a chceli by využiť LS,
- Programujú v iných systémoch a chceli by zistiť, ako na to v Lotus Notes.

Nech už patríte k ľubovolnej skupine, na konci seriálu budete nielen „múdrejší“, ale budete mať aj svoju vlastnú aplikáciu s LS modulmi a pri troche šťastia aj cenu zo súťaže... Na začiatok už len zopár poznámok:

- Ako vývojové prostredie bude popisovaný Domino Designer R7/8 (v súčasnosti najrozšírenejší). Moderná, „Eclipse based“ verzia Domino Designera 8.5x bude zmienená v záverečnej časti.
- Z dôvodu prehľadnosti bude popisovaný len jeden „programátorský“ postup, nie všetky alternatívne možnosti.
- Na stránkach www.edevelopment.sk (časť Tipy) nájdete predchádzajúci vývojársky seriál.

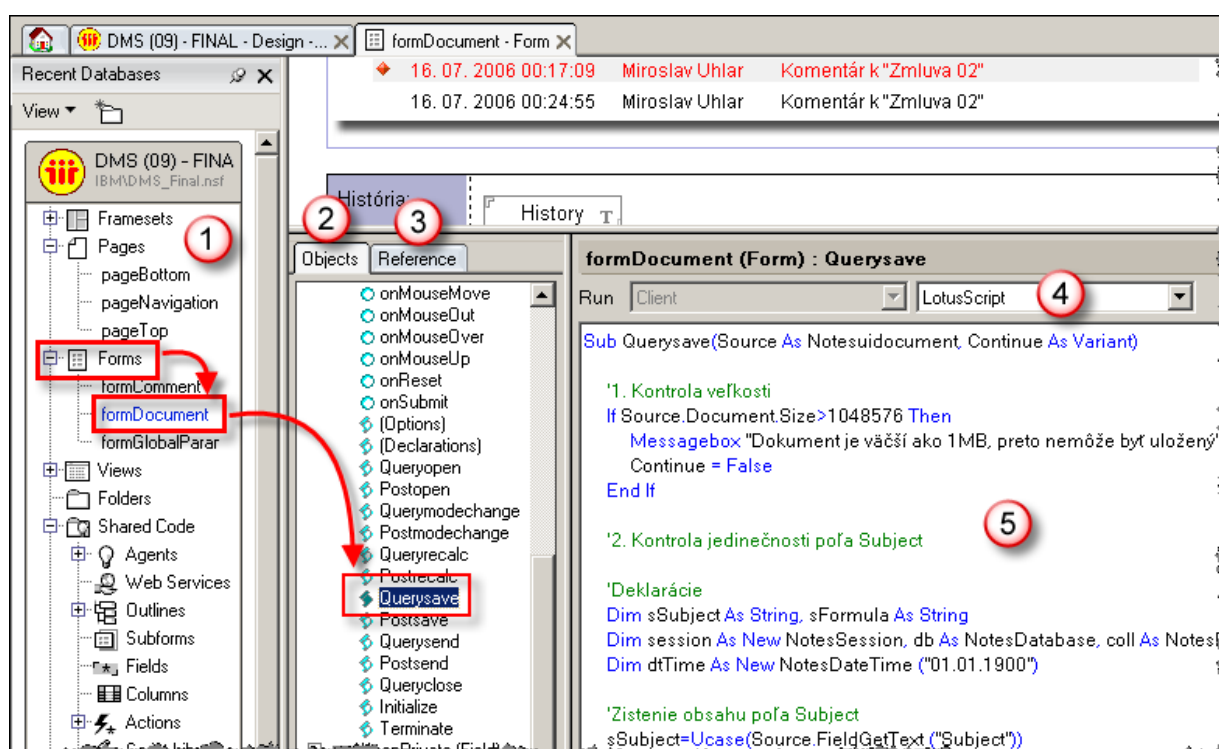
ČO JE TEDA LOTUS SCRIPT

LS je programovací script jazyk, ktorý:

- Je syntakticky podobný Visual Basicu
- Je použiteľný v určitých častiach Notes aplikácie (teda je platformne nezávislý)
- Sprístupňuje sadu objektov Notes aplikácie s ich vlastnosťami a metódami
- Je súčasťou integrovaného vývojového prostredia – Lotus Domino Designera
- Výrazne prekračuje možnosti tzv. „@Formula language“
- Umožňuje zoskupovať vlastné funkcie a procedúry do tzv. LS knižníc

ORIENTÁCIA V PROSTREDÍ DOMINO DESIGNERA

- Z <http://www.edevelopment.sk> si do dátového adresára Lotus Notes klienta stiahneme aplikáciu *DMS_Final.NSF* (časť Tipy)
- Spustíme program DD
- Zvolíme *File-Database-Open*, vyhľadáme aplikáciu a otvoríme ju.
- V skupine *Forms* otvoríme fomulár *formDocument* a v objektoch vyhľadáme udalosť *QuerySave*, kde sa nachádza LS sekvencia:



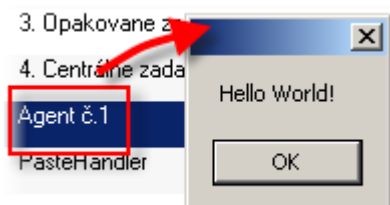
Jednotlivé časti charakterizuje tabuľka:

Časť	Vysvetlenie
1. Main Resource Window	Výber z objektov Domino databáz (aplikácií)
2. Object Tab	Výber udalosti, ktorú je potrebné programovať. Ponuka sa kontextovo prispôsobuje typu zvoleného objektu.
3. Reference Tab	Informácie o jednotlivých triedach (či už Lotus Notes alebo iných programoch)
4. Language Combo Box	Výber programovacieho jazyka (Notes formulas, Java, JavaScript a LotusScript) pre programovanie zvolenej udalosti daného objektu. Možnosti výberu sú dané konkrétnym typom zvolenej udalosti.
5. Main Window	Editor pre pre písanie LS kódu. Jeho vlastnosti (štýl písma, dopĺňanie parametrov, odsadzovanie odstavcov, farby pre príkazy, komentáre,.....) sú definované v Programmer's Pane (pod pravým tlačítkom myši).

Prax: MESSAGEBOX

Teraz už nastal čas pre demonštračné použitie LS, ktorý nám do dialógového okna vypíše text. Pre tento účel je v LS pripravená funkcia *MessageBox*.

- V aplikácii vytvoríme nového agenta, pričom:
 - Ako názov zadáme *Agent č.1*
 - Vlastnosť *Target* zvolíme na *None*
- Do udalosti *Initialize* zadáme kód: `MessageBox "Hello, World!"`
- Agentu uložíme (čím prebehne kompilácia kódu) a zatvoríme
- Nasledne agenta spustíme (menu *Agent - Run*) – na obrazovke sa objaví dialógové okno, vytvorené pomocou LS:



TIP1:

V časti *Reference – LotusScript Language* sú uvedené všetky LS funkcie. Po označení a stlačení *F1* sa zobrazí dokumentácia z Helpu.

TIP2:

Kompilácia kódu prebieha automaticky pri ukladaní, pričom dôjde k upozorneniu na syntaktické chyby. Chyby je možné rozdeliť na:

- Compile-time error: Chybu odhalí už kompilátor (napr. zlé parametre funkcie)
- Run-time error: Chyba vznikne až pri vykonávaní kódu (napr. delenie nulou)

SYNTAKTICKÉ PRAVIDLÁ písania LS kódu

Pred písaním ďalších LS kódov je vhodné pripomenúť s, že:

- Príkazy LS sú zložené z riadkov, pričom prázdny riadok neovplyvňuje vykonávanie scriptu.
- Text scriptu môže začínať úplne vľavo alebo byť odsadený podľa potreby.
- V rámci príkazu sú jeho jednotlivé prvky oddelené medzerou alebo tabelátorom. Bez problému je možné doplniť aj ďalšie prázdne znaky.
- Príkaz (s výnimkou blokových príkazov) musí byť na jednom riadku. Ak je príliš rozsiahly, možno ho spojiť s ďalším riadkom pomocou prázdneho znaku a podčiarkovníka: (_).

PRAX: Rozšírenie kódu

Kód agenta si teraz upravíme nasledovne:

```
sText="Hello, World!"
```

```
MessageBox sText
```

```
sText="Ahoj, svet!"
```

```
Print sText
```

Vysvetlenie kódu:

1. riadok: Do premennej *sText* je načítaný textový reťazec
2. riadok: Obsah premennej je zobrazený v dialógovom okne (MessageBox)
3. riadok: Obsah premennej *sText* je prepísaným novým reťazcom
4. riadok: Obsah premennej je vypísaný v dolnom stavovom riadku (Print)

LS Debugger

Krokovanie LS kódu a identifikácia chýb je úlohou LS Debuggera. Jeho základné funkcie si vysvetlíme na krokovaní kódu nášho agenta:

- LS Debugger aktivujeme menu príkazom *Tools – Debug Lotus Script*
- Spustíme agenta, čím sa zobrazí prostredie LS debuggera
- V spodnej časti si prepne okno *Variables*, ktoré zobrazuje hodnoty premenných
- Príkazom *Step Into/Step Over (F8)* kroкуjeme vykonávanie scriptu až po 3. riadok
- Overíme si možnosť zmeniť obsah premennej počas vykonávania tak, že do *sText* vložíme iný reťazec a po prechode na 4.riadok uvidíme v stavovej lište výpis nami zadaného reťazca

