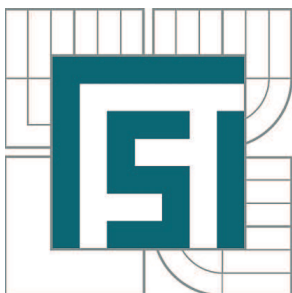


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

MS EXCEL A WORD ADD-IN PRO NAČÍTÁNÍ DAT Z DATABÁZE SQL SERVERU

MS EXCEL AND WORD ADD-IN FOR RETRIEVING DATA FROM SQL DATABASE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Ing. DAVID KAMRLA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ KOVÁŘ, Ph.D.

BRNO 2015

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Ing. David Kamrla

který/která studuje v **bakalářském studijním programu**

obor: **Aplikovaná informatika a řízení (3902R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

MS Excel a Word Add-In pro načítání dat z databáze SQL serveru

v anglickém jazyce:

MS Excel and Word Add-In for retrieving data from SQL database

Stručná charakteristika problematiky úkolu:

Načtení dat z databáze lze v programech MS Excel a Word provést pomocí datového připojení. Složitější zpracování a formátování lze docílit mmj. pomocí maker. Cílem práce je vytvořit Add-In, jehož úkolem bude spojit obě funkce. První etapou práce bude rešeršní studie na dané téma. Druhá část práce bude zaměřena na realizaci (programování) Add-Inu.

Cíle bakalářské práce:

Cílem práce bude naprogramovaný Excel a Word Add-In (v jazyce C#) pro načítání a zpracování dat z databáze SQL serveru.

Seznam odborné literatury:

[1] Troelsen, A. Pro C# 2010 and the .NET 4 Platform, 2010, ISBN: 1430225491

[2] Ben-Gan, I. Microsoft® SQL Server® 2008 T-SQL Fundamentals (PRO-Developer), 2008, ISBN: 8178531046

Vedoucí bakalářské práce: Ing. Jiří Kovář, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2014/2015.

V Brně, dne 21.11.2014

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
Děkan fakulty

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně, a že jsem uvedl všechny použité informační zdroje.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 31. května 2015

.....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Jiřímu Kováři, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování této bakalářské práce.

Bibliografická citace díla

KAMRLA, D. MS Excel a Word Add-In pro načítání dat z databáze SQL serveru. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 35 s. Vedoucí bakalářské práce Ing. Jiří Kovář, Ph.D..

Klíčová slova

C#, .NET Framework, Add-In, MS Office, SQL.

Anotace

Tato bakalářská práce se zaměřuje na realizaci Add-Inu v jazyce C# pro programy Microsoft Excel a Microsoft Word, jehož účelem je rychlý přístup do SQL databáze a vypisování přijatých výsledků do aktivního listu sešitu nebo dokumentu. Add-In zprostředkovává snadný přístup do Konferenční databáze Ústavu automatizace a informatiky pro obvyklé SQL dotazy. Add-In může být překonfigurován pro práci s jinou databází bez zásahu do zdrojového kódu.

Keywords

C#, .NET Framework, Add-In, MS Office, SQL.

Anotation

This bachelor thesis focus on development and implementation of the Add-In for Microsoft Excel and Microsoft Word programs for a straight-forward access to the SQL database and writing results into an active worksheet or document. The Add-In creates user-friendly access into the Conference database for Institute of Automation and Computer Science for frequently used SQL queries. The Add-In can be reconfigured to be aligned for work with another database without change to the source code.

Obsah

1	Úvod	7
2	Rozbor úlohy.....	8
3	Seznámení s jazykem C#	10
3.1	Nástroje .NET Framework	11
4	Rozbor kódu programu	13
4.1	Předávání dat mezi formuláři.....	13
4.1.1	Předávání dat pomocí konstruktoru.....	13
4.1.2	Předávání dat pomocí přímého přístupu k proměnným.....	14
4.1.3	Čtení a zápis v INI souboru	14
4.2	Připojení do SQL serveru	15
4.3	Vkládání dat do listu MS Excelu	17
4.4	Vkládání dat do MS Word dokumentu	18
5	Popis MS Office Add-Inu.....	20
5.1	Základní uspořádání ovládacích prvků	21
5.2	Formulář pro SQL Login	22
5.3	Volání SQL příkazů pomocí Add-Inu	24
5.4	Možnosti přenastavení Add-Inu	25
5.4.1	Nahrání nastavení	26
5.4.2	Změna nastavení.....	26
5.5	Obecné SQL dotazy	30
5.6	Přizpůsobení Add-Inu pro MS Word	31
6	Závěr	32

1 Úvod

Microsoft již vytvořil několik aplikací pro přístup k datům v MS-SQL databázi, jmenovitě např. Microsoft Access a Microsoft SQL Server Management Studio. Běžný uživatel však s těmito aplikacemi neumí zacházet a zpravidla je ani nemá k dispozici.

V dnešní době se pro běžného uživatele nejčastěji zprostředkovává komunikace s SQL databází přes webové rozhraní, zpravidla pomocí skriptovacího programovacího jazyka PHP nebo pomocí CLR-kompatibilního jazyka a ASP.NET knihoven. Webové rozhraní má mnoho nesporných výhod, ale před dalším zpracováním je často nutné data převést do tabulkového procesoru nebo textového souboru. Webové rozhraní může na výstup vygenerovat soubor s daty, který si uživatel může stáhnout a dále s ním pracovat, ale i toto řešení je zdlouhavé. Uživatel zpravidla chce mít data ihned k dispozici v tabulkovém procesoru nebo textovém editoru. Nejrozšířenější tabulkový procesor a textový editor je v balíku Microsoft Office, proto bude tato práce zaměřena na programy z tohoto balíku.

Cílem této práce je vytvořit uživatelsky přívětivý Add-In pro programy Microsoft Excel a Microsoft Word, který by sloužil ke čtení a zápisu dat do SQL databáze. V Microsoft Office již je implementována funkce pro získání dat z databáze, ale není zde rozumný správce přihlášení a možnost jednoduše vyvolat často používané dotazy. Tato bakalářská práce je zaměřena na možnosti programovacího jazyka C# a využití platformy .NET Framework pro tvorbu Add-Inu, který by umožňoval snadné a uživatelsky přívětivé čtení a zápis v SQL databázi.

2 Rozbor úlohy

Přístup do SQL databáze se zpravidla zprostředkovává přes webové rozhraní pomocí jazyka PHP nebo jazyka C# s využitím knihoven ASP.NET [1, 2]. Mnohdy je vhodné mít data v textovém dokumentu pro další publikaci, nebo v tabulkovém procesoru, kde lze data dále zpracovávat nebo analyzovat. Stahování dat z rozhraní webových stránek je časově náročná činnost, která sice může být usnadněna tím, že jedním z výstupů mohou být data stažitelná ve formě tabulkového souboru, např. ve formátu *.csv, *.txt nebo *.xls, ale i v takovém případě uživatel musí provést nezanedbatelný počet kroků, aby data stáhl, otevřel v tabulkovém procesoru a naformátoval je.

Mohlo by se zdát, že MS Access nebo podobný emulátor pro přístup do databáze je ideální volbou, ale v praxi uživatelé s takovým programem neumí zacházet, či jej dokonce ani nemají k dispozici. Pro běžné uživatele je často vhodnější, aby mohli snadno přistupovat k datům v databázi přímo z běžně používaného tabulkového procesoru, např. z MS Excelu, Lotus Symphony, LibreOffice nebo Apache OpenOffice. Nejrozšířenější je MS Office, proto je tento projekt zaměřen právě na tuto platformu.

Programy MS Office již mají implementované rozhraní pro načítání dat z SQL databáze [1], ale toto rozhraní vyžaduje znalosti tvorby dotazů a vnitřního uspořádání databáze. Pro běžného uživatele by bylo vhodnější rozhraní, které by v sobě již mělo implementované konkrétní často využívané SQL dotazy. Na základě těchto skutečností byl stanoven cíl této práce a to vytvoření Add-Inu pro snadný přístup do SQL databáze (viz list Zadání).

Pro další rozšíření použitelnosti bude implementován způsob pro přizpůsobení Add-Inu pro potřeby konkrétního uživatele, tím se Add-In stane univerzálním – bude umožňovat snadný přístup do více databází a bude plnit potřeby mnohem širšího spektra uživatelů. Toto přizpůsobení vyžaduje znalosti ohledně dotazů a vnitřní uspořádání tabulek SQL databáze. Znalý uživatel si může nastavení přizpůsobit svépomocí, méně zručný uživatel může k nastavení využít pomoc správce konkrétní SQL databáze nebo může importovat nastavení z konfiguračního souboru, který obdrží od správce databáze nebo Add-Inu.

Úlohu lze řešit pomocí .NET Framework za pomoci Visual Basic nebo jazyka C# [1]. Pro tento projekt jsem zvolil C#, protože s tímto jazykem mám zkušenosti z předchozího studia. Kapitola 3 uvádí stručný úvod do tohoto programovacího jazyka, ukázky programového kódu jsou uvedeny a rozebrány v kapitole 4. Popis funkcionalit Add-Inu ve formě blízké uživatelské příručky obsahuje kapitola 5.

3 Seznámení s jazykem C#

Programovací jazyk C# [sí šárp] je objektově orientovaný a je součástí vývojového prostředí Visual Studio .NET [dot net] Framework. Toto prostředí, včetně jazyka C#, bylo oficiálně uvedeno na trh v roce 2002 firmou Microsoft jako konkurence programovacímu jazyku Java.

Označení C# není jediné, se kterým se lze setkat. Někdy lze narazit i na zápis C# a v některých naučných textech i na zápis v textovém přepisu *C Sharp*. Význam tohoto označení nebyl nikdy oficiálně publikován, proto si lze jen domýšlet, co za něj autoři schovali.

Tento jazyk nevychází z jazyků C, C++ , ale byl jako první navržen přesně podle systému CLR (Common Language Runtime). Ideou CLR je vytvoření společného prostředí pro běh programů napsaných v různých jazycích. Díky CLR spolu mohou jednotlivé části programu (nebo celé programy) navzájem spolupracovat, přestože byly napsány každý v jiném programovacím jazyce. Aby byl programovací jazyk kompatibilní s CLR, musí vyhovovat specifikacím CLS (Common Language Specifications).

Přestože s jistými úpravami vyhovuje specifikaci CLS více programovacích jazyků, které jsou pro prostředí .NET upravovány, C# je tomuto prostředí šitý přesně na míru. Pokud se kompiluje program v C#, výsledkem není přímo strojový kód proveditelný procesorem, ale kód v jazyku CIL (Common Intermediate Language), který je teprve pomocí CLR přeložen do nativního kódu procesoru. CIL byl dříve označován jako MSIL (Microsoft Intermediate Language).

Nejedná se tedy o jazyk interpretovaný, jako třeba Java, ze které se C# hodně inspiroval, ale ve výsledku přímo o spustitelný kód, i když generovaný automaticky až těsně před spuštěním programu. Ačkoli je CIL velmi podobný assembleru, neexistuje v současné době žádný procesor, který by ho uměl nativně vykonávat. Proto se musí před spuštěním přeložit do strojového kódu za pomoci kompilátoru označovaného jako JITter. [3, 4, 5]

3.1 Nástroje .NET Framework

Seznam význačných vlastností jazyka C# a vývojového prostředí .NET Framework ukazuje následující přehled.

Jednoduchá dědičnost (inheritance) – Každá třída (class) může mít jednoho nebo více potomků, ale každá třída může být potomkem pouze jedné třídy, tzn. že třída může zdědit členy jednoho jiného objektu a pouze je rozšířit. Toto omezení je možno obejít přes rozhraní.

Násobná implementace rozhraní (interface) – každá třída může implementovat jedno nebo více rozhraní a každé rozhraní může být implementováno jednou nebo více třídami.

Garbage collector – je část CLE sloužící jako automatický správce paměti. Automaticky odstraňuje dynamicky alokovaný prostor objektů, které dále již nejsou používány. Samotné uvolnění paměti je označováno jako garbage collecting.

Metody – jsou funkce definované v rámci nějaké třídy. Pomocí přístupových modifikátorů lze nastavit, zda lze metodu použít i mimo třídu. Dalšími modifikátory jsou *static* a *abstract*.

Přístupové modifikátory – slouží k řízení přístupu ke členům (members) a metod mimo danou třídu:

- *public* – viditelné z dané sestavy (assembly) nebo jiné sestavy, která tuto odkazuje.
- *protected* – viditelné pouze z dané třídy nebo struktury či jeho potomka.
- *private* – viditelné pouze z dané třídy nebo struktury.
- *internal* – viditelné z dané sestavy.
- *protected internal* - viditelné z dané sestavy nebo z jiné sestavy skrze potomka.

Události (events) – umožňují třídě upozornit jinou třídu nebo více tříd, že v ní došlo ke změnám. Ostatní třídy pak mohou tuto událost zachytávat a příslušným způsobem na ni zareagovat.

Zpracování chyb pomocí výjimek (exception handling) – výjimka je objekt nesoucí informaci o chybě, která v rámci provádění programu nastala a zachycením tohoto objektu lze na daný chybový stav patřičným způsobem reagovat, aniž by došlo k pádu běhu programu.

Zajištění zpětné kompatibility v binární podobě – program je schopen pracovat i se starší verzí knihoven bez potřeby nového překladu.

Zajištění zpětné kompatibility v podobě zdrojového kódu – je vyžadován nový překlad, ale není třeba změn v kódu. [1, 5, 6]

4 Rozbor kódu programu

V této kapitole budou popsány vybrané části programového kódu, které jsou implementovány v Add-Inu.

4.1 Předávání dat mezi formuláři

Často je potřeba proměnné (variables) předávat mezi různými formuláři (form). Jednotlivé způsoby jsou vysvětleny na příkladu, kdy chceme údaj z textového pole `textBox1` formuláře `Form1` převést do štítku `Label1` okna formuláře `Form2`.

4.1.1 Předávání dat pomocí konstrukturu

Formulář se podobně jako jiné třídy (class) inicializuje pomocí konstrukturu, který může obsahovat jeden či více vstupů. Tento způsob je vhodný jen pro předávání nízkého počtu proměnných.

Zde je příklad inicializace formuláře `Form2` se vstupem textového řetězce (string) proměnné `Input`, který se vloží do popisu štítku `Label1`:

```
public Form2(string Input)
{
    InitializeComponent();
    Label1.Text = Input;
}
```

Při volání instance třídy `Form2` z formuláře `Form1` pak stačí zvolený datový typ (nebo kombinaci datových typů) zadat, např. takto:

```
Form2 frm2 = new Form2(textBox1.Text);
frm2.Show();
```

Řetězec z textového pole se ve druhém formuláři inicializuje jako `Input`.

4.1.2 Předávání dat pomocí přímého přístupu k proměnným

Tento způsob spočívá v tom, že se oba formuláře programově propojí. Ve vyvolávaném formuláři Form2 se v deklarační části vytvoří instance formuláře Form1 s názvem frm1:

```
private Form1 frm1 = new Form1();
```

V původním formuláři Form1 se nastaví propojení s instancí formuláře frm1, který je deklarován v druhém formuláři a poté se otevře nový formulář frm2:

```
Form2 frm2 = new Form2(textBox1.Text);  
Form2.frm1 = this;  
frm2.Show();
```

Předávání dat pak lze inicializovat ve formuláři Form2 voláním deklarované instance frm1, např. takto:

```
Label1.Text = frm1.TextBox1.Text;
```

Proměnné není nutné deklarovat a udržovat aktuální v obou formulářích, proto zde odpadá problém s nekonzistencí proměnných mezi oběma formuláři.

4.1.3 Čtení a zápis v INI souboru

V určitých případech je žádoucí uchovávat hodnoty proměnných i po ukončení běhu programu, nejvhodnější je použít ukládání do externího inicializačního souboru (dále jen INI souboru) nebo konfiguračního souboru na bázi XML [1, 3].

Do Add-Inu je importován jmenný prostor INIfiles s třídou INIfile, která obsahuje konstruktor, metodu IniWriteValue pro zápis do souboru a metodu IniReadValue pro čtení ze souboru. Hodnoty proměnných jsou ukládány pod klíčem, který je zpravidla shodný s názvem proměnné. Pro lepší přehlednost jsou klíče zatříděny do skupin klíčů. Třída INIfile je univerzální a lze ji použít v každém projektu, do kterého se importuje INIfiles.cs a kde se aktivuje jmenný prostor pomocí direktivy `using IniFiles;`.

Pro zvolený příklad by zdrojový kód ve formuláři Form1 mohl vypadat takto:

```

static private string Filename =
    AppDomain.CurrentDomain.BaseDirectory +
    "AddIn.ini";
IniFile ini = new IniFile();
ini.IniWriteValue("Form1", "textBox1", textBox1.Text);

```

V prvním kroku je určen řetězec `Filename`, který obsahuje plný název INI souboru včetně přípony a cesty k souboru. V ukázce je název souboru získán z privátní konstanty, ale lze jej získat např. i z dialogového okna pro uložení souboru nebo jiného INI souboru. V druhém kroku je zavolán konstruktor třídy `IniFile` a ve třetím je zapsána aktuální hodnota z textového pole `textBox1` do INI souboru metodou `IniWriteValue`. Do jednoho souboru lze uložit více proměnných. Pomocí cyklů `for`, `foreach` a `switch` lze ukládat složené proměnné typu pole (array), výčet (list) nebo indexovanou strukturu.

Hodnotu proměnné lze získat z INI souboru např. takto:

```

IniFile ini = new IniFile(Filename);
Label1.Text = ini.IniReadValue ("Form1", "textBox1");

```

Metoda `IniReadValue` přečte ze souboru `Filename`, sekce `"Form1"` hodnotu, která je uložena pod klíčem `"textBox1"`. Nahrát lze více proměnných, které jsou uloženy v daném INI souboru. Data jsou dostupná kdykoli a pro jakýkoli proces, který má přístup k INI souboru.

4.2 Připojení do SQL serveru

Pro připojení do databáze SQL serveru je třeba znát adresu SQL serveru, jméno databáze, uživatelské jméno a heslo. Tyto údaje jsou v řetězci údajů pro připojení (connection string), jak je uvedeno níže (kurzívou je proměnný text) [1]:

```

string connString =
    "Server=adresa_serveru; " +
    "Database=jméno_satabáze; " +
    "User Id=uživatelské_jméno; " +
    "Password=uživatelské_heslo; " +

```

V rámci důvěryhodného připojení k SQL serveru je možno použít přihlašovací údaje do operačního systému Windows. Potom v řetězci místo

User ID a Password může být Trusted Connection=True nebo Trusted Connection=SSPI nebo Integrated Security=SSPI [1]. Poslední varianta je preferovaná, protože je podporována vícero druhy databázových serverů [3].

Když jsou známy přihlašovací údaje i samotný dotaz (query), je možno se připojit k databázi za pomoci třídy SqlConnection a případné výsledky zachytávat třídou SqlDataAdapter do kolekce, např. typu seznam (List) nebo datová tabulka (DataTable) [1]. Příklad programového kódu je níže:

```
using SQL = System.Data;

string query = "SELECT * FROM * dbo.Table";

SQL.DataTable DT = new System.Data.DataTable();
SqlConnection conn = new SqlConnection(connString);
SqlDataAdapter da = new SqlDataAdapter(query, conn);
try {
    da.Fill(DT);
}
catch (SqlException E) {
    //kód pro ošetření vyjímky z SQL serveru
}
catch (Exception Ex) {
    //kód pro ošetření vyjímky
}
finally {
    conn.Close();
}
```

Sled bloků hlídaný blok (try), blok obsluhy (catch) a koncový blok (finally) slouží k prevenci pádu programu. Pokud chyba nastane v hlídaném bloku, tak program přeskočí zbytek hlídaného bloku a provede instrukce v bloku obsluhy. Jestliže chyba nenastane, blok obsluhy bude vynechán. Koncový blok se provede vždy, nezávisle na tom, zda chyba nastala, či nikoliv [5]. V Add-Inu se zpravidla blokem obsluhy vyvolá hlášení o chybě a probíhající funkce se předčasně ukončí. Předpokládaným místem, kde může dojít k chybě je např. tvorba spojení k SQL serveru, když uživatel zadá chybné heslo [2]. Pomocí cyklů lze převést data z adaptéru dat např. do listu aktivního sešitu nebo do tabulky dokumentu.

4.3 Vkládání dat do listu MS Excelu

Prvním krokem je deklarace objektu aktivního listu sešitu sheet:

```
using Excel = Microsoft.Office.Interop.Excel;

Excel.Worksheet sheet = ((Excel.Worksheet)Globals.
    ThisAddIn.Application.ActiveSheet);
```

V druhém kroku se vloží hlavička do prvního řádku aktivního listu a v posledním kroku se zaplní buňky listu daty přijatými z SQL serveru:

```
string[] ColumnNames = new string[DT.Columns.Count];
int col = 0;
foreach (SQL.DataColumn dc in DT.Columns) {
    ColumnNames[col++] = dc.ColumnName;
}
char lastColumn = (char)(65 + DT.Columns.Count - 1);
Excel.Range Header =
    sheet.get_Range("A1", lastColumn + "1");
Header.Value2 = ColumnNames;

SQL.DataRow[] dr = DT.Select();
string[,] rowData = new string[dr.Count<SQL.DataRow>(),
    DT.Columns.Count];
int RowCount = 0;
foreach (SQL.DataRow row in dr) {
    for (int col = 0; col < DT.Columns.Count; col++) {
        rowData[RowCount, col] = row[col].ToString();
    }
    RowCount++;
}
sheet.get_Range("A2", lastColumn +
    RowCount.ToString()).Value2 = rowData;
```

Pozn.: Programový kód Add-Inu navíc obsahuje bloky k ošetřování mezních stavů (např. stav, kdy žádné entity neodpovídají danému dotazu) a další prvky, viz soubor s programovým kódem Ribbon1.cs na CD příloze. Z důvodu zachování názornosti je zobrazený programový kód o tyto prvky zjednodušen.

4.4 Vkládání dat do MS Word dokumentu

Způsob vkládání dat do MS Wordu programovým kódem je odlišný. Každá tabulka je v MS Word dokumentu odlišný objekt, který má své pořadové číslo a zobrazí se v dokumentu na definované pozici [1]. Před vložením nové tabulky program metodou `Tables.Count` zjistí počet již existujících tabulek a nové tabulce přiřadí následující volný index. Velikost tabulky odpovídá rozměrům datové tabulky z SQL serveru DT a tabulka je vložena na pozici karetu (textového kurzoru) nebo aktivního výběru (selection). Tyto kroky jsou zahrnuty v ukázce programového kódu uvedeného níže:

```
using Word = Microsoft.Office.Interop.Word;

Word.Document doc =
    ((Excel.Document)Globals.ThisAddIn.
    Application.ActiveDocument);
Word.Range rng = doc.Application.Selection.Range;
Word.Range tableLocation =
    doc.Range(rng.Start, rng.End);
doc.Tables.Add(tableLocation,
    DT.Rows.Count + 1, DT.Columns.Count);
Word.Table newTable =
    doc.Tables[doc.Tables.Count];
```

V dalších krocích jsou hlavička a tělo tabulky za pomoci cyklů a typů `DataColumn` a `DataRow` naplněny daty z SQL a výsledná tabulka je naformátována. Toto je zachyceno v programovém kódu níže, jenž navazuje na kód z kapitoly 4.2:

```
int col = 0;
foreach (SQL.DataColumn dc in DT.Columns) {
    cell = newTable.Cell(1, col++ + 1);
    cell.Range.Text = dc.ColumnName;
}
if (col == 0) {
    //výjimka - 0 atributů přijato z SQL databáze
}
SQL.DataRow[] dr = DT.Select();
int RowCount = 0;
foreach (SQL.DataRow row in dr) {
    for (col = 0; col < DT.Columns.Count; col++) {
        cell = newTable.Cell(RowCount + 2, col + 1);
        cell.Range.Text = row[col].ToString();
    }
}
```

```
        RowCount++;
    }
    if (RowCount == 0) {
        //výjimka - 0 entit přijato z SQL databáze
    }
    newTable.ApplyStyleHeadingRows = true;
    newTable.AutoFormat();
    newTable.Columns.AutoFit();
```

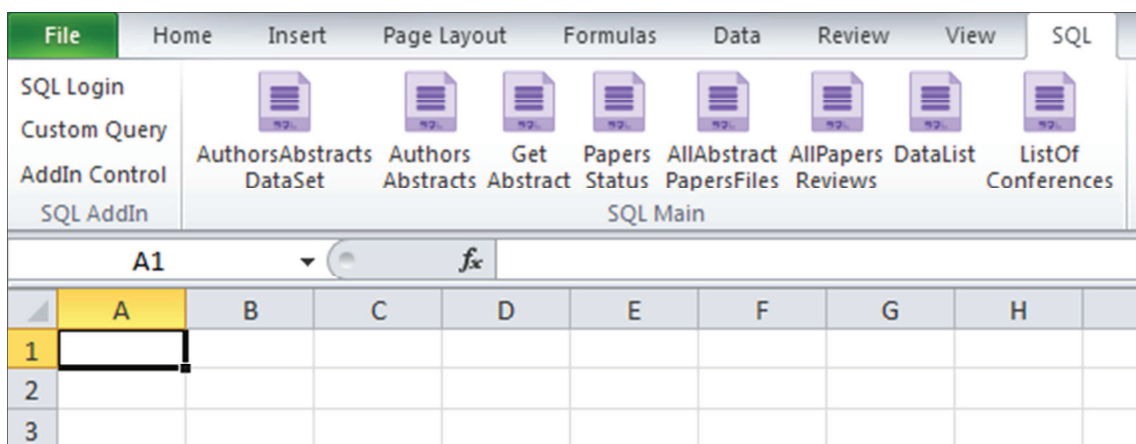
Pozn.: Zobrazený programový kód je uveden pro ukázkou, neobsahuje ošetřování všech výjimek.

5 Popis MS Office Add-Inu

Na následujících stránkách bude vysvětlena funkce MS Excel Add-Inu pro přístup do SQL databáze. MS Word Add-In je takřka totožný, vzájemné rozdíly jsou popsány v kapitole 5.6.

Add-In byl programován s ohledem na uživatelskou přívětivost. Ovládací prvky byly uspořádány tak, aby ovládání bylo intuitivní. Při neúspěchu konkrétní akce je uživatel informován tak, aby věděl kde se stala chyba a jaké kroky má provést jako nápravné opatření.

Add-In je podporován ve verzi Microsoft Excel 2010 nebo novější [1]. Pro přístup do SQL databáze je nutno znát přihlašovací údaje, tedy jméno SQL serveru a název databáze a pokud pro přístup do SQL serveru nelze použít přihlašovací údaje do Windows, pak je třeba ještě znát uživatelské jméno a heslo pro přístup do SQL serveru.



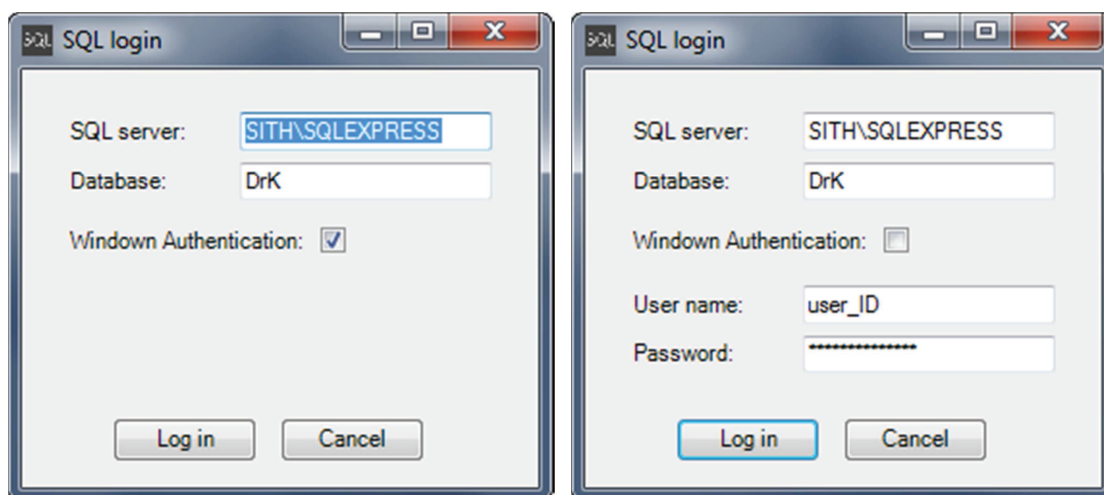
Obr. 1 - Add-In – Ukázka lišty

Při spuštění Excelu s Add-Inem bude k dispozici nová lišta s názvem „SQL“ (viz Obr. 1) ve které se nachází ovládací prvky, které byly naprogramovány na základě této práce.

5.1 Základní uspořádání ovládacích prvků

Všechny ovládací prvky programu jsou vyvolány z přídatné lišty s názvem „SQL“ nebo jsou vyvolávány z formuláře, který je spouštěn z této lišty.

Jak je vidět na Obr. 1, lišta Add-Inu obsahuje dvě skupiny ovládacích prvků – „SQL Add-In“ a „SQL Main“. Ve skupině „SQL Main“ jsou tlačítka pro rychlý přístup do databáze, která provedou přednastavený dotaz a případně ve vyskakovacím formulářovém okně požádají uživatele o doplnění proměnných. Např. tlačítko „Authors Abstracts“ odešle dotaz typu uložená procedura (stored procedure) do databáze Konferenčního systému Ústavu automatizace a informatiky a přijaté výsledky vepíše do aktivního listu Excelu. Velmi rychle tak lze získat seznam abstraktů všech prací, které jsou pro danou osobu registrované v konferenčním systému.



Obr. 2 - Add-In – Formulář pro přihlašovací údaje

Tlačítka v sekci „SQL Add-In“ slouží k uživatelskému nastavení Add-Inu, zde je jejich výčet se stručným popisem:

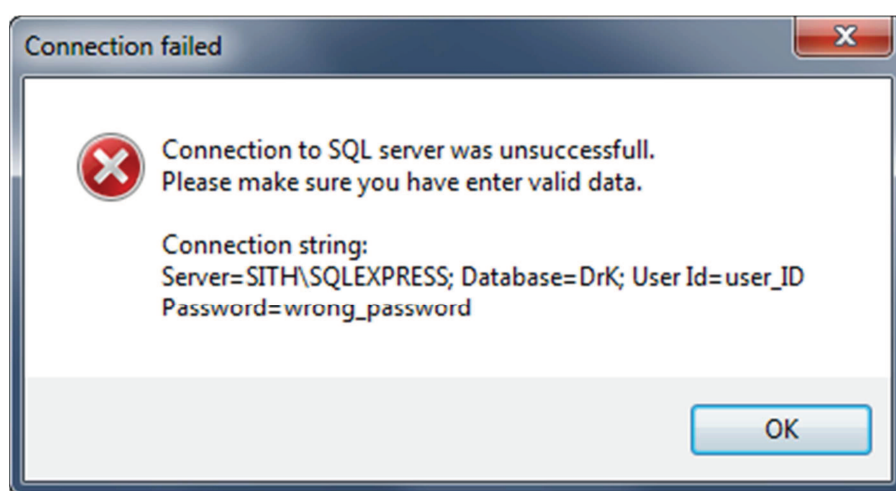
SQL Login – vyvolá formulář pro vložení přihlašovacích údajů do databázového serveru.

Custom Query – vyvolá formulář, do něj lze ručně napsat obecný dotaz, odeslat jej SQL serveru a vypsát výsledky do aktivního listu.

Add-In Control – vyvolá formulář, ve kterém lze upravit nastavení a parametry tlačítek, které jsou v sekci „SQL Main“.

5.2 Formulář pro SQL Login

Při prvním spuštění Add-Inu je třeba nastavit přihlašovací údaje do SQL databáze. Tlačítkem „SQL Login“ se vyvolá přihlašovací formulář, jak je znázorněno na Obr. 2 vlevo.

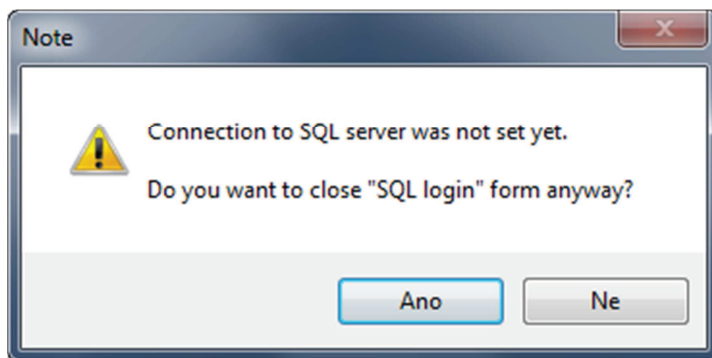


Obr. 3 - *Add-In* – Chybová hláška pro špatné přihlašovací údaje

Pokud se pro přístup do databáze nepoužívají přihlašovací údaje do operačního systému MS Windows, tak po odznačení kolonky „Windows Authentication“ je nutno vepsat také SQL uživatelské jméno a heslo viz Obr. 2 vpravo.

Pomocí tlačítka „Log in“ uživatel dá programu pokyn k testovacímu spojení se serverem a pokud toto spojení proběhne v pořádku, pak si program dané přihlašovací údaje zapamatuje. V případě neúspěchu je uživatel informován chybovou hláškou, viz Obr. 3.

Pokud uživatel předčasně uzavře okno formuláře, ke změně přihlašovacích údajů nedojde a program si v paměti zachová poslední otestované přihlašovací údaje. Uživatel dostane možnost předčasné zavření formuláře odvolat, viz okno se zprávou na Obr. 4.



Obr. 4 - Add-In – Varování při předčasném uzavření okna formuláře

Také ostatní funkce programu jsou v podobném duchu ošetřeny proti chybám nebo nevhodným uživatelským zásahům, ale v dalších popisech již budou zmiňovány pouze krátce.

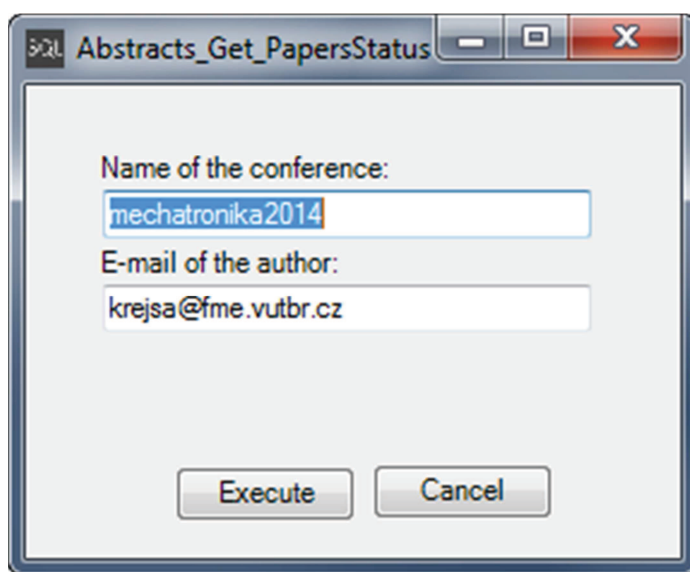
File								Home	Insert	Page Layout	Formulas	Data	Review	View	SQL
SQL Login		Custom Query		AddIn Control		SQL AddIn		SQL Main							
List of Owners		List of Buildings		Building Types		Generic Query		Invalid Query		No Results					
A1								fx id_student							
	A	B	C	D	E	F	G								
1	id_student	Jméno	Příjmení	Dosažené vzdělání											
2	1	David	Kamrla	VŠ											
3	2	Rostislav	Procházka	VŠ											
4	3	Jan	Opletal	SŠ											
5	4	Hung	Nguyen	SŠ											
6	5	Jozef	Strniště	ZŠ											
7															

Obr. 5 - Add-In – Ukázka promítnutí dat z SQL serveru do MS Excelu

Add-In si pamatuje poslední známé přihlašovací údaje, kromě hesla, pomocí inicializačního souboru (*.ini). Při novém spuštění MS Excelu budou poslední známé přihlašovací údaje kromě hesla předvyplněny ve formuláři.

5.3 Volání SQL příkazů pomocí Add-Inu

Jak již bylo řečeno, v sekci „SQL Main“ lišty „SQL“ jsou tlačítka s uživatelsky nastavitelnými parametry, která slouží pro vykonávání SQL příkazů a následného výpisu výsledků do aktivního listu (worksheet) aktivního *.xls souboru (workbook), přičemž naformátuje hlavičku – změní barvu a tloušťku písma, barvu pozadí, zmrazí hlavičku a zapne auto-filtr. Na Obr. 5 je ukázka vyplnění prázdného sešitu daty z SQL databáze programově pomocí Add-Inu.



Obr. 6 - Add-In – Dynamický formulář pro upřesnění dotazu

Vybrané dotazy jsou v Add-Inu uloženy se zástupnými symboly místo proměnných, které uživatel zadá až při volání dotazu ve zvláštním formuláři. Např. pod tlačítkem „Paper Status“ (z Obr. 1) se skrývá dotaz:

```
EXEC Abstracts_Get_PapersStatus
```

```
@IN_nameofconference = '#',
```

```
@IN_useremail = '#';
```

V tomto řetězci (string) je znak mřížka # zástupný symbol pro proměnnou, kterou vloží uživatel do dynamického formuláře, jak je ukázáno na Obr. 6. Program za pomoci třídy StringBuilder nahradí zástupný znak uživatelskou proměnnou, odešle jej SQL serveru a přijaté výsledky vloží do aktivního listu sešitu. Pro uvedený příklad by Add-In odeslal tento dotaz:

```
EXEC Abstracts_Get_PapersStatus

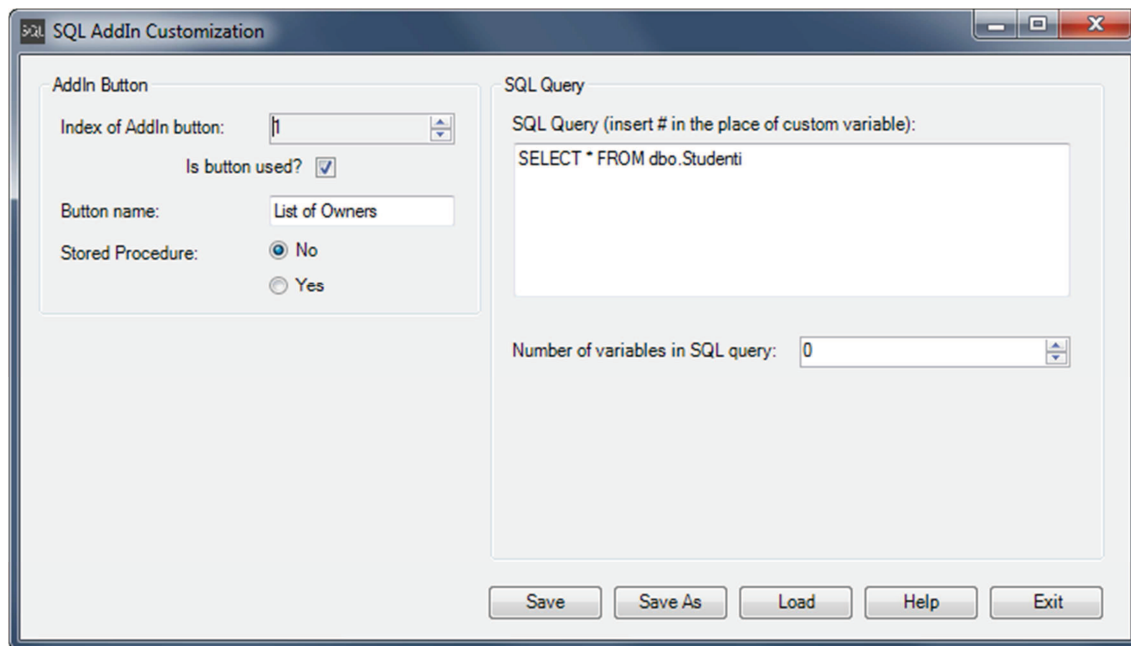
@IN_nameofconference = 'mechatronika2014',

@IN_useremail = 'krejsa@fme.vutbr.cz';
```

Dotaz výše volá uloženou proceduru v Transact-SQL databázi, stejný postup by fungoval i pro strukturovaný dotaz SELECT # FROM # WHERE #.

5.4 Možnosti přenastavení Add-Inu

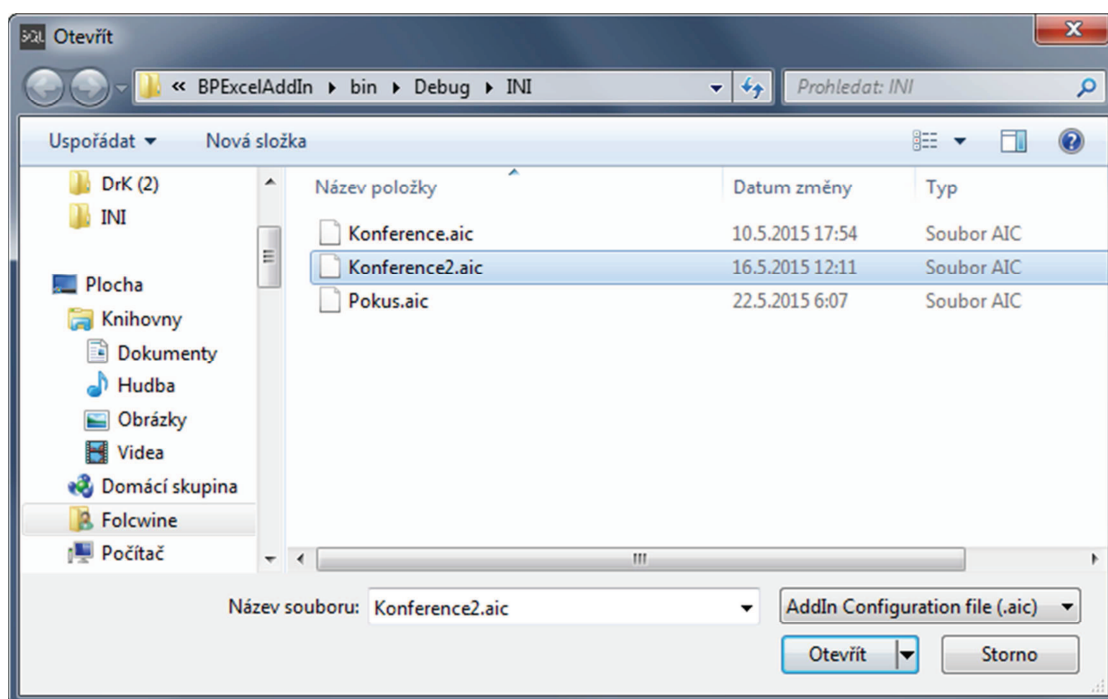
Při srovnání Obr. 1 a Obr. 5 je patrné, že tlačítka jsou různá, lze je totiž přenastavit pomocí formulářového okna (viz Obr. 7) spustitelného tlačítkem „Add-In Control“ v sekci „SQL Add-In“.



Obr. 7 - Add-In – Formulářové okno pro uživatelské přizpůsobení

5.4.1 Nahrání nastavení

Alternativní nastavení je možné vyvolat tlačítkem „Load“, které otevře běžné okno prostředí Windows pro otevírání souborů, viz Obr. 8. Lze tak nahrát jiné nastavení uživatelských tlačítek a v několika sekundách kompletně Add-In přenastavit. Programově je toto řešeno pomocí INI souborů, u nichž byla pro lepší orientaci změněna koncovka na *.aic (Add-In Control).



Obr. 8 - Add-In – Okno pro nahrávání souborů

5.4.2 Změna nastavení

Uživatelská tlačítka lze také jedno po druhém modifikovat, k čemuž ve formuláři „SQL Add-In Customization“ slouží ovládací prvky ve skupinách „Add-In Button“, „SQL Query“ a „Exceptions“.

Add-In Button

Index of Add-In button – jedná se o pořadové číslo tlačítka, při změně se z paměti nahrají hodnoty pro zvolené nové tlačítko.

Is button used? – pokud tato kolonka je odškrtnuta, tak dané tlačítko na liště nebude zobrazeno a ovládací prvky související s tímto tlačítkem se v konfiguračním formuláři zablokují.

Button Name – je název tlačítka, který se zobrazí na liště pod ikonkou.

Query Type – pokud se zatrhne volba „Stored Procedure = Yes“, tak se zpřístupní sekce „Exceptions“, sloužící pro ošetřování výjimek uložených procedur.

SQL Query

SQL Query – zde uživatel může vepsat vlastní dotaz, který se spustí příslušným tlačítkem. Na místo každé proměnné, kterou má uživatel ručně vkládat při každém spuštění příkazu, se vepíše znak mřížka „#“ (hash).

Number of variables in SQL query - tato hodnota se nastaví na počet proměnných, které uživatel má ručně definovat při každém běhu programu.

Name of query form – zde se uvede nadpis formuláře, kterým bude uživatel dotazován při spuštění daného SQL dotazu.

Selected index of custom variable – číslo pořadí proměnné v SQL dotazu, následující položky se vážou k tomuto číslu a při změně tohoto čísla se nahrají z paměti.

Label of variable in Add-In form – zde patří název proměnné, která se uživateli zobrazí u kolonky pro danou proměnnou

Prefill for variable in Add-In form – zde je možno uvést předvyplněnou hodnotu pro danou proměnnou, která může sloužit jako příklad nebo vodítko, co se do tohoto políčka má vyplnit.

Exceptions

Number of Exceptions – uvádí počet návratových hodnot z T-SQL databáze, které se ošetří chybovou hláškou, jež se zobrazí uživateli.

Exception index – číslo pořadí návratové hodnoty, zbývající kolonky v této sekci jsou řízeny tímto číslem. Při změně tohoto čísla se hodnoty dalších kolonek načtou z paměti.

Exception return value – je návratová hodnota (exception).

Exception message – je políčko pro obsah zprávy, který se zobrazí uživateli, pokud server pošle návratovou hodnotu uvedenou o kolonku výše.

Confirmation of success – pokud je zaškrtnuto, tak to značí, že daná návratová hodnota je potvrzením, že T-SQL dotaz byl přijat a úspěšně zpracován. Zpravidla bude toto políčko nezaškrtnuto, což znamená, že došlo k výjimce (exception) a dotaz nebyl zpracován.

SQL AddIn Customization

AddIn Button

Index of AddIn button: 1

Is button used?

Button name: AuthorsAbstracts DataS

Stored Procedure: No Yes

Exceptions

Number of Exceptions: 3

Exception index: 1

Exception return value: 50310

Exception message: OK: Abstracts_Get_Getl

Confirmation of success?

SQL Query

SQL Query (insert # in the place of custom variable):

```
EXEC Abstracts_Get_GetListOfAuthorsAbstractsDataSet
@IN_emailofauthor = '#, @IN_nameofconference = '#;
```

Number of variables in SQL query: 2

Name of the query form: Abstracts_Get_GetListOfAuthorsAbs

Selected index of custom variable: 1

Label for variable in AddIn from: E-mail of the author:

Prefill for variable in AddIn from: kovar@fme.vutbr.cz

Save Save As Load Help Exit

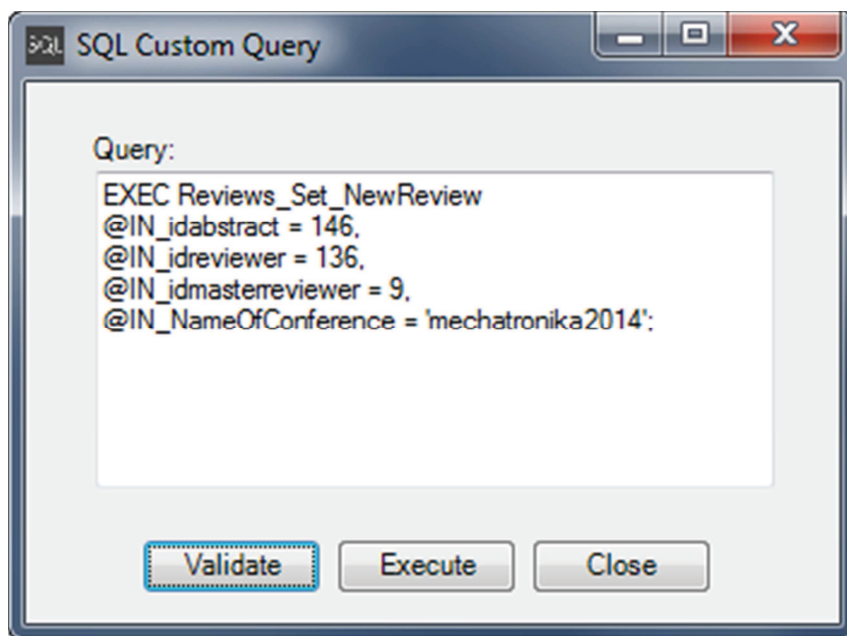
Obr. 9 - Add-In – Formulář pro uživatelské přizpůsobení s ukázkou pro Konferenční databázi ÚAI FME VUT v Brně

Jak je zřejmé z porovnání obrázků Obr. 7 a Obr. 9, tento formulář se snaží být co nejvíce intuitivní, zobrazují se pouze ovládací prvky, které jsou pro dané

volby relevantní. Např. pokud dotaz neobsahuje žádnou uživatelskou proměnnou (*Number of variable in SQL query = 0*), nezobrazí se všechny níže položené ovládací prvky ze skupiny „SQL query“. A to z toho důvodu, že způsob jejich vyplnění by byl stejně ignorován. Podobně tak celá sekce Exception je viditelná, jen pokud je zatrženo „Stored Procedures“ = „Yes“.

Přenasazení uživatelských tlačítek se ukládá do mezipaměti a není implementováno, dokud se dané nastavení nepotvrdí tlačítkem „Save“. Tím se také aktualizuje konfigurační soubor, takže změny se zachovají i v novém běhu MS Excelu. Alternativně lze tlačítkem „Save As“ dané nastavení z mezipaměti uložit do nového konfiguračního souboru, zároveň se tento konfigurační soubor nastaví jako aktivní a původní zůstane beze změn.

Zavřením okna s formulářem se mezipaměť vyčistí a všechny neuložené změny zapomenou. Uživatel o tom bude zpraven a bude mu dána možnost zavření formuláře zvrátit v dialogovém okně. Uložené změny se ihned promítnou na liště Excelu.



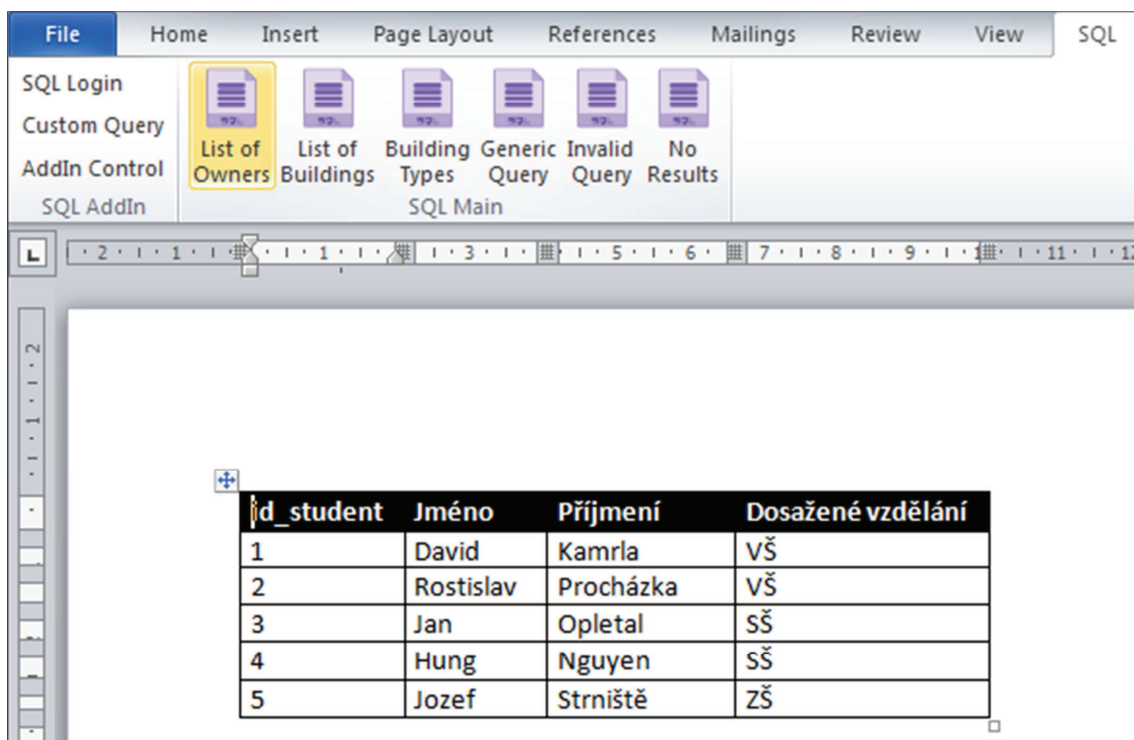
Obr. 10 - Add-In – Formulář pro obecný dotaz

Celkem lze nastavit až 20 tlačítek pro rychlý přístup do SQL, každé z nich může mít až 10 proměnných a 10 výjimek. Tyto počty lze poměrně

jednoduchým zásahem do zdrojového kódu navýšit, podrobný postup je popsán v komentářích ve zdrojovém kódu u deklarace příslušných konstant NumMaxButtons, NumMaxExeptions a NumMaxVariables, viz zdrojový kód na přiloženém CD.

5.5 Obecné SQL dotazy

Tlačítkem „Custom Query“ ze sekce „SQL Add-In“ a lišty „SQL“ se vyvolá formulář, viz Obr. 10, pomocí něhož lze odeslat takřka libovolný dotaz. Tato funkce je zde spíše jako doplněk, protože podobnou funkci provádí základní Excel příkazem v „Data/From other Source/From SQL Server“ [1]. Výhodou naprogramovaného tlačítka tohoto Add-Inu je automatické zapamatování posledního přihlášení na SQL server a posledního dotazu. Hlavním důvodem pro tvorbu této funkce bylo testování a ladění jiných částí programového kódu.



Obr. 11 - Add-In – Ukázka v MS Word

5.6 Přizpůsobení Add-Inu pro MS Word

Add-In byl prvně vytvořen a otestován pro MS Excel, poté byla vytvořena verze Add-Inu pro MS Word. Obě verze se liší jen nepatrně, hlavní změnou je v sekci pro vypisování výsledků – do MS word dokumentu se data z SQL vkládají ve formě nové tabulky, která se umístí na pozici karetu (textového kurzoru) nebo výběru v dokumentu a jejíž velikost a obsah odpovídá datové tabulce (DataTable) získané od SQL serveru (podrobnosti viz kapitoly 4.3 a 4.4). Ukázka tabulky vytvořené za pomoci Add-Inu je na Obr. 11.

6 Závěr

Výsledkem této práce je MS Excel a Word Add-In, který byl vytvořen pomocí jazyka C# 4.0 ve vývojovém prostředí Microsoft Visual Studio 2013. Tento Add-In může uživatel využít ke snadnému přístupu do SQL databáze pomocí často používaných dotazů. Všechny ovládací prvky Add-Inu jsou v nové liště „SQL“, popřípadě ve vyskakovacím okně přivolaném z dané lišty.

Add-In byl testován vůči obvyklým chybám jako ztráta spojení, chybné přihlašovací údaje a tyto výjimky byly ošetřeny. Pro testování byla používána vlastní databáze spravovaná programem Microsoft SQL Server Management Studio 2013.

Tento Add-In byl přizpůsoben pro volání uložených procedur T-SQL databáze Konferenčního systému Ústavu automatizace a informatiky. Většinu často používaných dotazů lze vyvolat stiskem jednoho tlačítka. Vybrané dotazy obsahují uživatelské proměnné, které se zadávají do dynamického formuláře.

Pro zvýšení univerzálnosti byla do Add-Inu implementována možnost úpravy nastavení uživatelských tlačítek přímo z Add-Inu bez nutnosti zásahu do zdrojového kódu. Je např. možno upravit název tlačítka, SQL dotaz, uživatelské proměnné a zprávy pro uživatele, když dojde k výjimce. Add-In také umožňuje jednoduše přepínat mezi různými sadami nastavení, uživatel díky tomu může střídavě pracovat s různými databázemi, pro které používá různé dotazy. Uživatelské nastavení, např. přihlašovací údaje (s výjimkou hesla), je přechováváno v inicializačních souborech.

Tento Add-In může být nadále rozšiřován, např. místo textových polí v uživatelských formulářích by bylo možné implementovat rolovací výběry, které by obsahovaly položky extrahované z SQL databáze. Možnost dalších uživatelských nastavení nebo podpora více jazykových mutací by mohla být vhodným navázáním na tuto práci.

Použitá literatura

[1] *Microsoft Developer Network* [online] Documentation for C#, .NET Framework and Visual Studio 2013, [cit. 2015-05-20]. Dostupné z WWW:

<<https://msdn.microsoft.com/en-us/library/ms123401.aspx>>

[2] LACKO, Ľuboslav. *PHP 5 a MySQL 5: hotová řešení*. Vyd. 1. Brno: Computer Press, 2007, 320 s. K okamžitému použití (Computer Press). ISBN 978-80-251-1695-1.

[3] TROELSEN, Andrew W. *C# a .NET 2.0 profesionálně*. Vyd. 1. Brno: Zoner Press, 2006, 1197 s. Encyklopedie Zoner Press. ISBN 80-86815-42-0.

[4] SHARP, John. *Microsoft Visual C# 2008: krok za krokem*. Vyd. 1. Brno: Computer Press, 2008, 592 s. Krok za krokem (Computer Press). ISBN 978-80-251-2027-9.

[5] BĚHÁLEK, Marek. *Programovací jazyk C#* [online]. VŠB-TU Ostrava, 2007. 144 s. Učební text. TU Ostrava, VŠB-FEI, katedra informatiky [cit. 2015-05-20]. Dostupné z WWW:

<<http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text.pdf>>.

[6] HERCEG, Tomáš. *Visual Basic .NET : .NET Framework od začátku* [online]. 2009, [cit. 2015-05-20]. Dostupné z WWW:

<http://www.vbnet.cz/serial--13-net_framework_od_zacatku.aspx>.

Seznam obrázků

Obr. 1 - Add-In – Ukázka lišty.....	20
Obr. 2 - Add-In – Formulář pro přihlašovací údaje	21
Obr. 3 - Add-In – Chybová hláška pro špatné přihlašovací údaje.....	22
Obr. 4 - Add-In – Varování při předčasném uzavření okna formuláře	23
Obr. 5 - Add-In – Ukázka promítnutí dat z SQL serveru do MS Excelu.....	23
Obr. 6 - Add-In – Dynamický formulář pro upřesnění dotazu	24
Obr. 7 - Add-In – Formulářové okno pro uživatelské přizpůsobení.....	25
Obr. 8 - Add-In – Okno pro nahrávání souborů	26
Obr. 9 - Add-In – Formulář pro uživatelské přizpůsobení s ukázkou pro Konferenční databázi ÚAI FME VUT v Brně	28
Obr. 10 - Add-In – Formulář pro obecný dotaz	29
Obr. 11 - Add-In – Ukázka v MS Word	30

Seznam použitých zkratek

CIL	Common Intermediate Language
CLR	Common Language Runtime
CLS	Common Language Specifications
INI	Inicializační (soubor)
MS	Microsoft™
MSDN	Microsoft Developer Network
MSIL	Microsoft Intermediate Language
PHP	Hypertext PreProcessor (po staru Personal Home Page)
SQL	Structured Query Language
T-SQL	Transact-SQL

Příloha – obsah CD

1. Elektronická verze bakalářské práce
 \76283_BP.pdf
2. Microsoft Visual studio projekt pro Excel Add-In
 \BPExcelAdd-In\
3. Microsoft Visual studio projekt pro Word Add-In
 \BPWordAdd-In\