

# Contact Transmutator

Jakub Svoboda

Martina Hlinková

Martin Bryndza

Martin Molnár

---

## Table of Contents

### [1. Úvod a špecifikácia projektu](#)

[Úvod](#)

[Členovia projektu, náplň práce a umiestnenie projektu](#)

[Motivácia](#)

[Špecifikácia](#)

### [2. Štruktúra projektu](#)

[Vysvetlenie niektorých pojmov](#)

[Štruktúra](#)

### [3. Informácie o použitých formátoch vstupných/výstupných dát](#)

[CSV](#)

[ODS](#)

[VCF](#)

### [4. Špecifikácia jednotlivých implementácií](#)

[InputFilter](#)

[InternalDocColumnSchema](#)

[InternalDoc2CompiledDoc](#)

[InternalDocAutodetectFormat](#)

[OutputFilter](#)

[VCFHelper](#)

[ReadCSV](#)

[ODSInput](#)

[ReadVCF](#)

[ReadCompiledDoc](#)

[InternalDocColumnSchemaImpl](#)

[InternalDocAutodetectFormatImpl](#)

[InternalDocCompiler](#)

[VCFConverter](#)

[VCFHelperImpl](#)

[WriteCSV](#)

[ODSWrite](#)

[WriteVCF](#)

[VCFTypesEnum](#)

## [5. Návod na použitie](#)

### [Návod na použitie](#)

## List of Examples

1.1. [Príklad operácie aplikácie](#)

3.1. [Príklad CSV](#)

3.2. [Príklad vCard 3.0](#)

# Chapter 1. Úvod a špecifikácia projektu

## Table of Contents

[Úvod](#)

[Členovia projektu, náplň práce a umiestnenie projektu](#)

[Motivácia](#)

[Špecifikácia](#)

## Úvod

Projekt Contact Transmutator vznikol ako súčasť výuky predmetu PB138 - Moderní značkovací jazyky a jejich aplikace vyučovanej na fakulte informatiky Masarykovej univerzity počas obdobia jarného semestra 2011 pod vedením doc. RNDr. Tomáša Pitnera Ph.D.

Cieľom projektu bolo naučiť sa pracovať s profesionálnymi nástrojmi pre komunitný vývoj, schopnosť pracovať a spracovávať dokumenty vo formátoch XML a osvojenie si práce so základnými štandardami a technológiami moderných značkovacích jazykov na báze XML a ich aplikácií.

## Členovia projektu, náplň práce a umiestnenie projektu

Členovia vývoja tohto projektu sú:

- Jakub Svoboda

### Note

základná kostra projektu, jeho základných rozhraní a implementácií funkcií; príprava filtrov pre CSV

- Martin Bryndza

### Note

programovanie grafického GUI; príprava filtrov pre CompiledDoc

- Martina Hlinková

## Note

príprava filtrov pre ODS (Open Document Format - spreadsheets);  
dokumentácia v DocBook

● Martin Molnár

## Note

príprava filtrov pre VCF; testovacie nástroje

Projekt je umiestnený na hostovskej stránke Google Code:

<http://code.google.com/p/contact-transmutator> , kde sa dajú zhliadnuť wiki stránky a návod, prípadne stiahnuť projekt.

## Motivácia

Adresáre s kontaktami sú neoddeliteľnou súčasťou moderného spôsobu života. V dnešnej dobe avšak existuje nepreberné množstvo formátov na uchovávanie kontaktov, adres, čísel, poznámok atď. Problém nastáva ak isté zariadenia preferujú len jeden z formátov. Prenos týchto citlivých dát sa preto stáva nepríjemne zdĺhavou záležitosťou. Ďalší problém nastáva s mentalitou konkrétnych užívateľov a ich potrieb. Každý jeden užívateľ môže svoje kontakty zaznamenávať inými spôsobmi nielen softvérovo ale i logicky.

Náš program Contact Transmutator je navrhnutý na riešenie tohoto problému, ako transformáciu formátov tak logické triedenie informácií v štruktúre kontaktov.

## Špecifikácia

Contact Transmutator parsuje neštandardne uložené dáta (napr. meno a číslo mobilného telefónu v jednom stĺpci záznamu formátach VCF, ODS, CSV). Aplikácia je schopná automaticky detekovať rôzne typy dát a ich hodnoty a rozparsovať ich do interného XML formátu. Užívateľovi je tiež umožnené určiť istú základnú parsovaciú logiku pomocou jednoduchého a intuitívneho grafického rozhrania. Navyše si užívateľ môže určiť v akom formáte sa uložia jeho kontakty.

### Example 1.1. Príklad operácie aplikácie

Používateľ má kontakty v MS Office Excel. V dokumente neexistuje takmer žiadna globálna logika (jeden riadok záznamu obsahuje niekoľko kontaktov, ktoré majú spolu neurčený vzťah, telefónne čísla a mená sú zmixované v jednom stĺpci, niektoré významné informácie sú v podobe poznámok). Užívateľ chce importovať tieto dáta do svojho mobilného telefónu. Ako prvý krok vyexportuje dáta z MS Office Excel do formátu CSV a použije aplikáciu Contact Transmutator pre zadelenie kontaktov do istej logickej štruktúry a uloženie do formátu, ktorý podporuje jeho mobilný telefón.

## Chapter 2. Štruktúra projektu

### Table of Contents

## Vysvetlenie niektorých pojmov

- *InternalDoc* = Vnútorý XML dokument, ktorý sa používa na načítanie informácií do Contact Transmutator. Dáta zo vstupných súborov sa parsujú do tohoto jednoduchšieho XML dokumentu. Funguje ako medzičlánok ku CompiledDoc. Neobsahuje informácie o type dát, ktoré sa v ňom nachádzajú.
- *CompiledDoc* = Vnútorý XML dokument, ktorý sa používa na spracovávanie informácií a ich parsovanie do výstupných filtrov. Obsahuje informácie o typoch dát v jednotlivých stĺpcoch/bunkách zoznamu kontaktov.
- *Stĺpcové schéma* = Pomocné XML, ktoré obsahuje informácie o jednotlivých stĺpcoch zoznamu.

## Štruktúra

Projekt sa delí na niekoľko samostatných častí, ktoré spolu komunikujú pomocou vnútorných XML dokumentov zmienených vo vysvetlení niektorých pojmov.

Prvá časť, Input, zahŕňa všetky vstupné filtry, ktoré berú zvolený dokument, načítavajú a parsujú z neho informácie, ktoré následne vypisujú do InternalDoc (resp. CompiledDoc pri vCard). V tejto časti sa nehľadí na obsah jednotlivých dokumentov. Z CSV sa parsujú informácie tak, že každý textový reťazec medzi čiarkami sa vloží do jednej položky neformátovaného kontaktu a jeden riadok vždy značí jeden kontakt. Z ODS sa otvorí zip a z XML content.xml (nesie informácie len o základnom formátovaní a obsahy jednotlivých tabuliek) sa načítajú informácie taktiež do InternalDoc. Každý riadok tabuliek predstavuje jeden kontakt a text každej bunky sa uloží do samostatnej položky. Výnimkou je VCF, pretože jeho štruktúra umožňuje získavanie informácií o každej položke v kontakte. Tieto informácie (na začiatku každého riadku) sa ľahko čítajú a je možné vytvorenie CompiledDoc bez medzistupňa.

Druhá časť je úzko spojená s GUI programom. Zahrňuje prevod InternalDoc do CompiledDoc. To znamená, že podľa výberu užívateľa v GUI sa mení XML stĺpcového schématu a následne sa podľa neho upravuje/vytvára CompiledDoc. Táto časť sa môže opakovať ľubovoľný počet krát, kým užívateľ nie je spokojný.

Posledná výstupná časť, Output, sa stará o prevod CompiledDoc do zvoleného formátu na miesto určené užívateľom.

## Chapter 3. Informácie o použitých formátoch vstupných/výstupných dát

### Table of Contents

[CSV](#)  
[ODS](#)  
[VCF](#)

# CSV

CSV(Comma-seperated values, hodnoty oddelené čiarkami) je jednoduchý súborový formát pre výmenu tabulkových dát. Súbor vo formáte CSV pozostáva z riadkov, v ktorých sú jednotlivé položky oddelené znakom čiarka (.). Hodnoty položiek môžu byť uzavreté do úvodzoviek (""), čo umožňuje, aby text položky obsahoval čiarku. Ak text položky obsahuje úvodzovky, sú zdvojené.

Keďže sa v niektorých jazykoch čiarka používa ako oddelovač desatinných miest, existujú varianty, ktoré používajú iný znak pre oddelovanie položiek než čiarku, najčastejšie bodkočiarku, prípadne tabulátor (taká varianta sa potom označuje ako TSV, Tab-separated values). Variantu s bodkočiarkou používa napríklad Microsoft Excel.

Pre tento formát neexistuje špecifikácia, popis formátu sa však nachádza v RFC 4180.

<http://tools.ietf.org/html/rfc4180>

## Example 3.1. Příklad CSV

1995,Opel,Vectra,"klimatizácia, strešné okno",45000

1998,Škoda,"Felicia ""Fun""",80000

2002,Škoda,Octavia,"klimatizácia, ABS poškodená",70000

# ODS

ODS je tabuľkový formát, ktorý je časťou The Open Dokument Format pre kancelárske aplikácie. ODF je opensource špecifikácia založená na XML formáte, vyvinutá organizáciou OASIS a implementovaná v OpenOffice.

ODS je formát vyvinutý pre tabuľkový program Calc, ktorý je súčasťou StarOffice alebo OpenOffice. Tento typ formátu pozostáva z tabuliek schopných nachádzať informácie, používať rôzne deklarácie, pracovať s numerickými dátami, matematickými formulami a grafmi. Každá tabuľka a jej analýza je uzavretá v dokumentovom prelúdiu a epilógu. Prelúdium pozostáva z formátovacích dát, možností použitia matematických formulí, pravidiel pre bunkový obsah, informáciách o zmenách atď. Rozsahy databázy, mená deklarácií a operácií a linky sú uvedené v epilógu.

# VCF

vCard je súborový formát pre výmenu osobných dát, predovšetkým elektronické obchodné vizitky. vCards sú najčastejšie priložené v e-mail správach, ale môžu sa vymieňať aj inými cestami, najčastešie prostredníctvom WWW stránok. Môžu obsahovať meno a priezvisko, adresy, telefónne čísla, URL adresy, logá, fotografie ako aj audio klipy. Súbor vCard sú ukladané vo formáte vcf, v ktorých sa ukladajú aj kontakty napríklad v mobilných telefónoch značky Nokia.

Verzia 2.1 je široko využívaná a podporovaná e-mailovými klientami. Verzia formátu vCrad 3.0 je štandardizovaná IETF a jej návrh je obsiahnutý v RFC2425 a RFC2426.

<http://tools.ietf.org/html/rfc2425><http://tools.ietf.org/html/rfc2426>

## Example 3.2. Příklad vCard 3.0

BEGIN:VCARD  
VERSION:3.0  
N:Gump;Forrest  
FN:Forrest Gump  
ORG:Bubba Gump Shrimp Co.  
TITLE:Shrimp Man  
PHOTO;VALUE=URL;TYPE=GIF:http://www.example.com/dir\_photos/my\_photo.gif  
TEL;TYPE=WORK,VOICE:(111) 555-1212  
TEL;TYPE=HOME,VOICE:(404) 555-1212  
ADR;TYPE=WORK;;;100 Waters Edge;Baytown;LA;30314;United States of America  
LABEL;TYPE=WORK:100 Waters Edge\nBaytown, LA 30314\nUnited States of America  
ADR;TYPE=HOME;;;42 Plantation St.;Baytown;LA;30314;United States of America  
LABEL;TYPE=HOME:42 Plantation St.\nBaytown, LA 30314\nUnited States of America  
EMAIL;TYPE=PREF,INTERNET:forrestgump@example.com  
REV:20080424T195243Z  
END:VCARD

## Chapter 4. Špecifikácia jednotlivých implementácií

### Table of Contents

[InputFilter](#)  
[InternalDocColumnSchema](#)  
[InternalDoc2CompiledDoc](#)  
[InternalDocAutodetectFormat](#)  
[OutputFilter](#)  
[VCFHelper](#)  
[ReadCSV](#)  
[ODSInput](#)  
[ReadVCF](#)  
[ReadCompiledDoc](#)  
[InternalDocColumnSchemaImpl](#)  
[InternalDocAutodetectFormatImpl](#)  
[InternalDocCompiler](#)  
[VCFConverter](#)  
[VCFHelperImpl](#)  
[WriteCSV](#)  
[ODSWrite](#)  
[WriteVCF](#)  
[VCFTypesEnum](#)

# InputFilter

## rozhranie

Rozhranie vstupného filtra, ktorého konštruktor s parametrami na nastavenie možností pre načítanie vstupného súboru. Ak sa načíta súbor, ktorý nie je v žiadnom validnom a používanom formáte XML (myslí sa tým formát používaný v Contact Transmutator), vytvorí sa nový prázdny dokument a metóda getColumnSchema() vráti NULL. Ak sa načíta súbor, ktorý je vo validnom a používanom formate XML, pri použití metódy read() sa vygeneruje InternalDocColumnSchema a metóda getColumnSchema vracia túto vygenerovanú schému.

- public Document read()

Metóda načítavajúca informácie zo zvoleného súboru a vytvárajúca InternalDoc pre ODS a CSV alebo priamo CompiledDoc v prípade VCF .

Return - Dokument v tvare InternalDoc alebo CompiledDoc.

- public InternalDocColumnSchema getColumnSchema()

Táto metóda vracia schému dokumentu.

Return - Logická schéma dokumentu.

## InternalDocColumnSchema

### rozhranie

Metódy tohoto rozhrania popisujú a pracujú s datovými typmi v stĺpcoch i InternalDoc:

- čo sa nachádza v jednotlivých stĺpcoch
- ktoré stĺpce by sa mali rozdeliť a ako
- ktoré stĺpce by sa mali spojiť a ako

Na svoju prácu využíva vnútorný privátny XML DOM. Tiedy tohoto rozhrania dokážu vytvoriť, meniť a pýtať sa na vlastníci stĺpcovej schémy interného XML (InternalDoc).

## InternalDoc2CompiledDoc

### rozhranie

Metódy tohoto rozhrania vezmú InternalDoc s InternalDocColumnSchema a prevedu ho do CompiledDoc. InternalDoc aj CompiledDoc sú dne formy XML, pomocou ktorých sa v Contact Transmutátore pracuje s dátami. Najdôležitejšími metódami sú:

- *getCompiledValidContacts()* Vracia správne zostavený CompiledDoc (zodpoveda približnej štruktúre VCF alebo ľahko prevediteľnej do VFC).
- *getCompiledInvalidContacts()* Vracia rovnaký formát ale predchádzajúca metóda, ale obsahuje i niektoré chyby typu: Informácie sú v dvoch rozdielnych poliach

miesto toho aby boli v jednom.

## **InternalDocAutodetectFormat**

### **rozhranie**

Triedy tohoto rozhrania vytvárajú z InternalDoc InternalDocColumnSchema s kandidátnymi typmi autodetekcie.

## **OutputFilter**

### **rozhranie**

Rozhranie tried, ktoré vytvárajú jeden z výstupných formátov. Triedy nedetekujú chyby v dátach, túto funkciu zastáva GUI ešte predtým, než sa zavolá OutputFilter. Konštruktor umožňuje zvolenie mena výstupného súboru a cestu k nemu, kódovanie a niektoré špecifické parametre.

## **VCFHelper**

### **rozhranie**

Trieda je zodpovedná za rozoznávanie, či môže existovať viac polí jedného typu pre jeden kontakt.

## **ReadCSV**

### **trieda implementujúca rozhranie InputFilter**

Trieda, ktorá nastavuje štruktúru interného XML dokumentu a následné rozparsovanie dát z pôvodného CSV do interného XML dokumentu. Ďalej vytvára novú internú stĺpovú schému.

## **ODSInput**

### **trieda implementujúca rozhranie InputFilter**

Trieda na načítavanie informácií zo súboru typu ODS do interného XML a následné rozparsovanie dát z pôvodného ODS do interného XML dokumentu. Ďalej vytvára novú internú stĺpovú schému.



## ReadVCF

### trieda implementujúca rozhranie InputFilter

Trieda na načítavanie informácií zo súboru typu VCF do interného XML a následné rozparsovanie dát z pôvodného VCF do interného XML dokumentu. Ďalej vytvára novú internú stĺpovú schému.

## ReadCompiledDoc

### trieda implementujúca rozhranie InputFilter

Trieda na načítavanie informácií z XML vyvinutého a používaného na spracovávanie informácií v Contact Transmutatore. Ďalej vytvára novú internú stĺpovú schému.

## InternalDocColumnSchemaImpl

### trieda implementujúca rozhranie InternalDocColumnSchema

Vytvára W3C XML DOM pre prácu so stĺpcami:

- umožňuje autodetekciu
- využíva sa pri grafickom rozhraní pre užívateľa
- využíva ju kompilátor pre výstupné informácie

Metódy tejto triedy dokážu rozoznať všetky možné kombinácie dát, spájať ich a rozdeľovať. Použitie XML DOM zaručuje robustnosť, flexibilitu a ľahké používanie.

## InternalDocAutodetectFormatImpl

### trieda implementujúca rozhranie InternalDocAutodetectFormat

## InternalDocCompiler

### trieda implementujúca rozhranie InternalDoc2CompiledDoc

## VCFConverter

### trieda

Pomocná trieda, ktorá prevádza názvy typov dát z VCF formy do podoby, ktorej by mal užívateľ rozumieť.

## VCFHelperImpl

### trieda implementujúca rozhranie VCFHelper

VCF podporuje mnoho rôznych úloh, ale len niektoré sa bežne používajú. Kvôli používaniu tagov ľuďmi sa využívajú identifikátory, ktoré sú ľahko pochopiteľné pre ľudské chápanie. GUI dokáže pomocou metód tejto triedy previesť identifikátory na ľahko zrozumiteľné texty.

## WriteCSV

### trieda implementujúca rozhranie OutputFilter

Trieda obsahuje metódy, ktoré z CompiledDoc vytvárajú CVS súbor so štruktúrou, ktorú požaduje užívateľ a zapisuje ho na vybrané miesto.

## ODSWrite

### trieda implementujúca rozhranie OutputFilter

Trieda obsahuje metódy, ktoré z CompiledDoc vytvárajú ODS súbor so štruktúrou, ktorú požaduje užívateľ a zapisuje ho na vybrané miesto.

## WriteVCF

### trieda implementujúca rozhranie OutputFilter

Trieda obsahuje metódy, ktoré z CompiledDoc vytvárajú VFC súbor so štruktúrou, ktorú požaduje užívateľ a zapisuje ho na vybrané miesto.

## VCFTypeEnum

### výčet

Pomocný výčet pre prácu s textovými identifikátormi, ktoré sa nachádzajú napríklad v VCFHelper.

## Chapter 5. Návod na použitie

### Table of Contents

[Návod na použitie](#)

# Návod na použitie

Po spustení programu sa na obrazovke vykreslí okno Contack Transmutator. Pre transformáciu dokumentu je potrebné zvoliť cestu k tomuto dokumentu. To sa dá dvomi spôsobmi.

- Vypísaním celej cesty do riadku.
- Kliknutím na Browse a vyhľadáním cesty k tomuto súboru.

Po nájdení/vypísaní cesty je potrebné vybrať v zozname vhodné kódovanie a kliknúť na tlačidlo Next.

V ďalšom okne sa vypíše zoznam, ktorý sa vytvoril z dát vybraného súboru. V hornej lište je nastapotrebné nastaviť, aké informácie sa vypisujú v ktorom stĺpci.

Pokiaľ stĺpec nemá byť vo výslednom zozname, stačí v zozname pre konkrétny stĺpec zadať "Delete this".

Pokiaľ jeden stĺpec obsahu informácie, ktoré by mali byť v dvoch alebo viacerých stĺpcoch, v zozname treba zvoliť funkciu "Split into". Po zvolení sa otvorí nové okno, kde je možné uviesť na koľko stĺpcov sa má zvolený stĺpec rozdeliť a v settings treba určiť, ktorý stĺpec má obsahovať aký typ informácií.

Podobne ako rozdeliť sa dajú aj spojiť stĺpce. Stačí len vybrať možnosť "Add to column" v stĺpci, ktorého obsah chceme presunúť. Otvorí sa okno, kde sa určuje spájaný stĺpec, do ktorého sa majú dopisovať informácie a znak, ktorým sa oddelia informácie. Ďalej je potrebné kliknúť na "submit". V tomto výbere je možné vybrať i ďalšie stĺpce, ktoré sa majú pripojiť k spojeniu a taktiež ich poradie.

Na spodnej časti okna sú ďalšie tlačidlá na ovládanie programu.

- *Refresh Table* - výber z vrchnej časti okna sa spracuje a tabuľka sa nanovo vypíše.
- *Add Column* - pridá nový prázdny stĺpec na konci zoznamu.
- *Back* - návrat na výber súboru a kódovania
- *Next* - dokončenie zmien na zozname.
- *Cancel* - zrušenie aplikácie.

Po kliknutí na "Next" sa otvorí okno s výzvom na zadanie cesty, mena a typu výstupného dokumentu.