Dokumentace úlohy CSV: CSV2XML v Perl do IPP 2011/2012
Jméno a příjmení: Branislav Blaškovič
Login: xblask00

Úvod

Při implementaci úlohy jsem využil vlastní vytvořený modul, který se stará jenom o převod CSV do XML formátu. Použil jsem knihovny *Text::CSV, XML::Writer* a *Getopt::Long*.

Hlavní soubor csv.pl

Soubor `csv.pl` obsahuje zpracování argumentů a volání metod nad instancí objektu `CSV2XML`. Před samotným vytvořením této instance se ale nejdříve ověří všechny vstupní argumenty, jestli nebyly zadané vícekrát, nebo jestli uživatel nechce jenom vypsat nápovědu.

Modul CSV2XML

Před voláním metody `read_file()`, která se stará o hlavní logiku převodu, je zapotřebí nastavit některé parametry, jako například znak separátoru nebo vstupní/výstupní soubor. Na všechno jsou připraveny metody objektu `CSV2XML`. Po nastavení všech možností, je zapotřebí zavolat metodu `validate()`, která se stará o ověření správnosti konfliktů v konfiguraci.

Zavoláním `read_file()` se začne načítat vstupní soubor (případně se začne číst ze standardního vstupu, pokud vstupní soubor nebyl zadán). Na čtení CSV využívám knihovnu `Text::CSV`, kterou si správně nastavím za pomoci zvolené konfigurace, která bude popsána v sekci *Nastavení knihoven*. Pro načítání po řádcích využívám metodu `getline()`, která se se správným nastavením stará o korektní načtení konce řádku.

Soubor se prohledává po sloupcích. Následně proběhne analýza, která se liší podle zadaných parametrů programu. Validuje zadaný kořenový element nebo nahrazuje nepovolené znaky za pomlčky, jestli se generují názvy sloupců z hlavičky – prvního řádku. Na to využívám regulární výrazy, které sem sestrojil za pomoci XML standardu. Řetězec, který validuji, bylo potřebné nejprve dekódovat za pomoci `utf8::decode($text)`.

Výsledek si nikam neukládám, ale přímo volám metodu `start_tag(), characters()` a `end_tag()` pro výpis.

K úspěšnému ukončení výpisu slouží metoda *print_output()*. Tato metoda zavírá soubor a ukončuje zápis voláním metody *end()* nad knihovnou *XML::Writer*.

Nastavení knihoven

`Getopt::long` knihovna slouží na načtení argumentů. Pro splnění podmínek zadaní bylo potřebné do konfigurace přidat `gnu_compat` pro správnou funkcionalitu `–param “text”` a `–param=”text”`. Řetězcové hodnoty přidávám do pole a tím zajišťuji unikátnost zadaného parametru. Skript vyžaduje přesně zadané parametry. Zkrácené verze nejsou povolené. Toto chování zabezpečuje nastavení `no_auto_abbrev`.

Na čtení CSV bylo potřebné nastavit knihovnu `Text::CSV` tak, aby vyhovovala uživatelsky zvoleným parametrům, jako například separátor. Pokud uživatel zadá jako separátor `TAB` nebo `SPACE`, program situaci zpracovává odlišně a separátor si upraví tak, aby vyhověl podmínce. Pro správné zpracování zalomení řádku v uvozovkách bylo potřebné nastavit

Dokumentace úlohy CSV: CSV2XML v Perl do IPP 2011/2012
Jméno a příjmení: Branislav Blaškovič
Login: xblask00

`binary` na hodnotu 1.

Zápis do souboru XML řídí knihovna `XML::Writer`, která se taktéž stará o vizuální stránku výsledné XML struktury. Odsazení jsem nastavil parametrem `DATA_INDENT`. Taktéž skript vkládá zalomení řádků před a za text, pro lepší přehlednost. Toto chování zabezpečuje `DATA_MODE` nastaven rovněž na 1. V případě, že uživatel nezvolí parametr -r pro vložení párového obalovacího elementu, je potřebné vypnout bezpečný mód za pomoci `UNSAFE`, jinak by zápis do XML skončil s chybou. Výsledný soubor se zadává parametrem `OUTPUT`, který je potřebné nastavit na file handler výstupního souboru.