

1. Popis programu

java -jar XMLsplitter.jar

```
-minSize=? maxSize=? divSize=? -config(=? ) headerXPath=? -contentXPath=? -i=? -o(=? ) -L=? -delimiter1=? -delimiter2=?
-minSize - half-optional. min xor max xor divSize must be set. Min size od output file-chunk (default kilobytes(k), b,m,g can be used)
-maxSize - half-optional. min xor max xor divSize must be set. Max size od output file-chunk (eg 100k is same as 100, 1m equals 1024k, 1024b
is equals to 1 or to 1k )
-divSize - half-optional. min xor max xor divSize must be set. Number of chunks file will be splitted
  Atleast one content element is always included.
  only one of ^^ should be declared (in fact, last one is used)
-config - config file with preset params. default is default.conf, commandline params overwrite in-file ones. If not set, then hardcoded
are used
-headerXPath - xpath expression defining header elements. Default is nothing
-contentXPath - xpath expression defining body of chunk. Default is /*
-footerXPath - xpath expression defining footer of chunk. Default is nothing
-o - optional. output directory for chunks. Default is working directory. When not set, output is std-out
-i - optional. inputfile. Defouult is std-in.
-L - optional. logging level. 0-(default) all messages go to error Stream. 1 - messages go to stdout, 2- no messages
-delimiter1 - optional. output files have name: inputFile_without_sufixDELIMITER1chunkNumberDEIMITER2.suffix default _part_
-delimiter2 - optional. output files have name: inputFile_without_sufixDELIMITER1chunkNumberDEIMITER2.suffix default is none
```

Tato aplikace dělí xml na menší části. Velikost částí je definovatelná třemi způsoby: minSize, maxSize a divSize. V případě minSize bude každý kousek alespoň tak velký, v případě maxSze nikdy nebude větší – než zadaná hodnota. Defaultní jsou kilobajty, lze použít příponu b-bajty,k-kilbajty,m-megabjty, g-giga... Divsiz=n dělí soubor na n přibližně stejných dílů. Výhoda divSize je že poslední dílek není doplněk jako v případě min a max sizes.

Header, content a footer xpath – ačkoliv tyto elementy nejsou povinné, jejich vynechání způsobí zajímavé výsledky. Definují části, které budou na začátku každého kousku, v těle kousků a na konci každého kousku.

Vstup/výstup – vstupem je standardní vstup nebo zadaný soubor i=file_name. Pro výstupní kousky je opět standardní výstup. Samostatné -o generuje soubory do pracovního adresáře a o=dir_name generuje soubry do zadaného adresáře

Konfigurace – místo zadávání parametrů z příkazové řádky lze nastavit konfigurační soubor sdílený XMLsplittem i XMLjoinem. Je vhodné do něj umístit xpath výrazy , dělicí velikost, logování, delimitery.. O něco méně vhodné je tam mít -i/-o. Parametry příkazové řádky přebijí parametry konfiguračku. Konfigurační soubor default.conf se použije při volbě -config. Dále se dá volat -config=file_name kdy se použije file_name subor.

Parametry delimiter není třeba měnit. Slouží jen pro kosmetické úprvy.

java -jar XMLjoiner.jar

```
-config - config file with preset params. default is default.conf, commandline params overwrite in-file ones. If not set, then hardcoded are used
-headerXPath - xpath expression defining header elements. Default is nothing
-contentXPath - xpath expression defining body of chunk. Default is /*
-footerXPath - xpath expression defining footer of chunk. Default is nothing
-o - optional. output file for joined xml. When not set, output is std-out, when set without param, then file name is generated from inputfiles. If directory, outfile will be guessed to this directory
-L - optional. logging level. 0-(default) all messages go to error Stream. 1 - messages go to stdout, 2- no messages
-delimiter1 - optional. will remove delimiter1(0-9)*delimiter2 from filename. default is _part_
-delimiter2 - optional. will remove delimiter1(0-9)*delimiter2 from filename. default is none
-checkHeaders - optional. will check headers and footers if they are consists over all files
-sortA - optional. will sort input files alphabeticly
-sortN - optional. will try to sort input files numericly
-i - optional. regular expression for finding of input files. Reg.ex is following java conventions
http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html
any other argument is cosidered as input file, notexisting are skipped, user warned
```

Tato aplikace umí spojit soubory vzniklé XMLsplitem (či čímkoliv jiným, ale aby splňovali zadání header/content/footer). Config,xpathy, L(oging),delimiters jsu shodné s XMLsplitter.

Výstup programu je na standardní výstup. S parametrem -o potom je uložen do souboru v pracovním adresáři odvozeného z názvu dílků. Při o=file_name se uloží do file_name.

Vstupem programu je množina souborů Lze zadat buď jednotlivé soubory, nebo soubory expandované příkazovou řádkou, nebo -i=some/directory/file_name_pattern kdys epřidají soubory ze some/directory vyhovující zadanému regulárnímu výrazu.

Např.: /prace/demo_partd*. kdy v disk obsahuje soubory

/demo_part15.xml

/prace/demo_part15.xml.txt

/prace/demo_part001.xml

/prace/demo_part_10.xml.txt

/prace/demo_part15-22.xml

/prace/zaloha/demo_part001.xml

vybere soubory:

/prace/demo_part001.xml

/prace/demo_part15-22.xml

Regex se řídí syntaxí javy(perlu) –

<http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html> , která je aplikací rozšířene regulární gramatiky o počítání.

Jednotlivé metody přidávání souborů lze kombinovat (dokonce i několik -o parametrů za sebou)

Pozor – pořadí vstupních suborů je zcela náhodné – a protože pořadí elementů na výstupu musí být většinou shodné se vstupním, je třeba soubory seřadit - buďto pomocí sortA – abecedně – nebo pomoci sortN – numericky. Numerické se užije hlavně v případech kdy se jsou vstupní soubory očíslovány bez doplňkových nul (čili místo smysluplného 001.002...999 je tam 1,2,3 .. pak abecední řazení selhává).

Parametr checkHeaders kontroluje, zdali mají vstupní soubory stejné hlavičky a patičky a rozdílný obsah. Měl by pomoci odhalit duplicitní soubory, případně soubory z jiného projektu.

java -jar XMLcompare.jar

```
-ignoreContent -failFast -silent file1 file2
at least two arguments - Two valid filenames are expected
-ignoreContent - optional. Will disable chcking of text values of elements (for case docment have been transalted)
-failFast - optional. terminate after first error
-silent - optional will
```

Tato aplikace slouží k validitě že soubor vzniklý rozdělením a složením původního souboru je shodný (ve smyslu XML) s originálem. Tzn ignoruje komentáře a mezi-elementový prostor.

Pozor – výstup programu je přibližně 7*velikost_vstupního_xml. S použitým silence je výstupem jediná věta – “xml jsou shodná” nebo “počet chyb ..”. FailFast je další urychlovač – program se ukončí při první chybě.

Ignorecontent je přepínač který způsobí že je kontrolována pouze struktura xml a ne jeho obsah tzn:

```
<root><element>aaa</element></root>
```

```
a
<root>
  <element>bbb</element>
</root>
```

Jsou shodné s –ignoreContent, ale rozdílné bez něj.

Samozřejmě

```
<root>
  <element>aaa</element>
  <element>bbb</element>
</root>
```

je odlišný od obou.

Vstupem jsou dva vstupní soubory a výstup jde na standardní výstup.

2. parametr conf a spouštěcí skript

Parametr conf může být zavádějící. Poněkud maximalistický zápis volání jaru může být:

```
java -Xmx1g -jar XMLsplitter.jar maxSize=1m headerXpath=/root/head/headers -contentXpath=/root/content/contents
-footerXpath=/root/footer/footers -i=/my/input/file.xml -o=/my/loved/directory -L=1 -delimiter1=_chunk- -delimiter2=-byMe
```

Což vypadá v pravdě komplikovaně. Volání `java -Xmx1g -jar` je spuštění javy `XMLsplitter.jar` je program který použijeme zbytek jsou jeho parametry. Nicméně jediné co je opravdu nutné zadat je vstupní (a pravděpodobně výstupní soubor). Přesně na toto slouží mechanismus `-conf`.

Pokud je spuštěn bez hodnoty, hledá soubor `default.conf`. Využití přímo tohoto souboru je poněkud nešťastné. Je vhodnější ho spustit `-conf=/nejaka/cesta/konfigurak.conf` kdy je tedy přesně řečeno jakou sadu parametrů použít. Syntaxe konfiguračního souboru je stejná jako příkazové řádky tedy:

```
headerXpath=/STEP-ProductInformationTranslation/Qualifiers | /STEP-ProductInformationTranslation/UnitList
contentXpath=/STEP-ProductInformationTranslation/Products/Product
L=1
delimiter1=hodnotaProProjekt
maxSize=1m

atd....
```

Důležité je že konfigurační soubor může být sdílen XMLsplitterem i XMLjoinerem. Také je třeba pamatovat že příkazová řádka přebije hodnoty v konfiguračním souboru. Volání se tedy zjednoduší na toto:

```
java -Xmx1g -jar XMLsplitter.jar -conf=/nejaka/cesta/konfigurak.conf -i=/my/input/file.xml -o=/my/loved/directory
```

Další zjednodušení volání už záleží na organizaci práce a různých stupních počítačové gramotnosti jednotlivých uživatelů. Uvážíme-li že nakonfigurování XPath výrazů může být netriviální pak připravení konfiguračního souboru může padnout na zkušenou osobu. Potom se dá ještě celý výraz upravit do nativního skriptu:

ukazka pro windows - soubor `split.bat`

```
java -Xmx1g -jar /cesta/ke/globalni/instalaci/XMLsplitter.jar -conf=/nejaka/cesta/konfigurak.conf -i=%1 -o=%2
```

Kde pak koncový uživatel zavolá `split.bat /muj/vstup.xml /muj/vystup/`

ukazka pro linux - soubor `split.sh`

```
#!/bin/sh
```

```
java -Xmx1g -jar /cesta/ke/globalni/instalaci/XMLsplitter.jar -conf=/nejaka/cesta/konfigurak.conf -i=$1 -o=$2
```

Kde pak koncový uživatel zavolá `./split.sh /muj/vstup.xml /muj/vystup/`

Podobný postup lze i pro joiner a compare.

3. porozumění kontrolním výpisům

Každý program puštěný bez parametrů vypíše všechny své možné argumenty. Po korektním spuštění vyúsuje právě prováděnou činnost (například nastavuji parametr A na hodnotu B), poté vypíše své nastavení (tedy sjednocení defaultních hodnot, konfiguračního souboru (pokud je použit) a parametrů řádky). Pokud program vykazuje nekorektní chování je dobré si toto pročíst (ideálně přesměrovat do logovacího souboru).

java -jar XMLsplitter.jar

```
logging set to: 1
reading: file:/home/jvanek/Desktop/demo/default.conf
header xpath setted: /STEP-ProductInformationTranslation/Qualifiers | /STEP-ProductInformationTranslation/UnitList
content xpath setted: /STEP-ProductInformationTranslation/Products/Product
divSze setted: 5
```

Zde končí nastavování hodnot. Následuje jejich sjednocené zopakování spolu s verifikacemi

```
output file without arg. used working directory
output dir setted: /home/jvanek/Desktop/demo/.
input file setted: /home/jvanek/Desktop/demo/demo-in.xml
logging: std-out
delimiter1 set _part_
delimiter2 set
header xpath: /STEP-ProductInformationTranslation/Qualifiers | /STEP-ProductInformationTranslation/UnitList
content xpath: /STEP-ProductInformationTranslation/Products/Product
footer xpath not set
input file: /home/jvanek/Desktop/demo/demo-in.xml
exists
output directory: /home/jvanek/Desktop/demo/.
exists
minsize is not set
maxsize is not set
file will be splitted to: 5 chunks
```

Zde začíná samotné dělení souboru

```
=====PROCESSING=====
```

```
parsing /home/jvanek/Desktop/demo/demo-in.xml Vstupní soubor
header list have 2 nodes!
```

počet elementů vybraných jednotlivými xpath výrazy – pokud se zde vyskytují 0, je xpathpravděpodobně špatně!

```
header size is 4605 bytes!
content list have 2104 nodes!
content size is 14568788 bytes! (~13m)
```

Velikost jednotlivých částí vybraných jednotlivými xpath výrazy – pokud se zde vyskytují 0, je xpathpravděpodobně špatně!

```
saving chunk 0 size 2915252bytes in 375 chunks. Total: 2910647 19%
...
saving chunk 4 size 2919938bytes in 428 chunks Total: 14568788 100%
```

Velikosta počet elementů vybraných xpath výrazem pro dělení obsah – pokud se zde vyskytují 0, je xpathpravděpodobně špatně! Pokud na konci nebude 100% (event. Nebudou se rovnat kontrolní velikosti) nastala buďto chyba zápisu nebo je opět něco (xpath?) špatně.

java -jar XMLjoiner.jar

```
logging set to: 1
reading: file:/home/jvanek/Desktop/demo/default.conf
header xpath setted: /STEP-ProductInformationTranslation/Qualifiers | /STEP-ProductInformationTranslation/UnitList
content xpath setted: /STEP-ProductInformationTranslation/Products/Product
output dir setted: /home/jvanek/Desktop/demo/concated.xml
sorting set to alhabetical
processing demo-in_partd*.* in /home/jvanek/Desktop/demo
```

15 candidates Celkový počet souborů v cílové složce. Následuje vyhodnocení které soubory vyhovují regulárnímu výrazu

```
input file added: /home/jvanek/Desktop/demo/demo-in_part_00002.xml
input file added: /home/jvanek/Desktop/demo/demo-in_part_00003.xml
input file added: /home/jvanek/Desktop/demo/demo-in_part_00004.xml
input file added: /home/jvanek/Desktop/demo/demo-in_part_00001.xml
input file added: /home/jvanek/Desktop/demo/demo-in_part_00005.xml
```

Zde končí nastavování hodnot. Následuje jejich sjednocené zopakování spolu s verifikacemi

```
logging: std-out
delimiter1 set _part_
delimiter2 set
header xpath: /STEP-ProductInformationTranslation/Qualifiers | /STEP-ProductInformationTranslation/UnitList
content xpath: /STEP-ProductInformationTranslation/Products/Product
footer xpath not set
5 input files
output file: /home/jvanek/Desktop/demo/concated.xml
```



```
input filesnow sorting alphabetically
/home/jvanek/Desktop/demo/demo-in_part_00001.xml
/home/jvanek/Desktop/demo/demo-in_part_00002.xml
/home/jvanek/Desktop/demo/demo-in_part_00003.xml
/home/jvanek/Desktop/demo/demo-in_part_00004.xml
/home/jvanek/Desktop/demo/demo-in_part_00005.xml
headers will be checked
```

Zde začíná samotné spojování souborů

```
=====PROCESSING=====
writing to /home/jvanek/Desktop/demo/concated.xml
header list have 2 nodes!
header size is 4605 bytes!
content list have 375 nodes!
content size is 2910647 bytes!)
```

Výpis počtu a velikosti elementů vrácených Xpathem. Pokud jsou tu nuly, je pravděpodobně špatně. Následující část se váže ke kontrole hlaviček:

```
checking heads:
loaded encoding: UTF-8
expected encoding: UTF-8
ok
loaded xml-version: 1.0
expected xml-version: 1.0
ok
processing instructions:
expeced: xml-stylesheet type="text/xsl" href="preview.xsl"
loaded xml-stylesheet type="text/xsl" href="preview.xsl"
have match: true
checking headers:
loaded headers size: 4605
expected headers size: 4605
are equals - ok
loaded headers count: 2
expected headers count: 2
are equals - ok
  match
  match
```

Hlavičky byvždyměli být shodné. Pokud některé nebudou pak se do seznamu souborů pravděpodobně zamíchal cizí soubor.

```
loaded content count: 375
unexpected content count: 375
are equals - quite suspicious
loaded content size: 2910647
unexpected content size: 2910647
are equals - VERY suspicious
```

Obsah by naopak měl být rozdílný (první soubor se porovnává sám se sebou, proto je zde varování). Pokud není, pak se do seznamu souborů pravděpodobně přimíchal nějaký duplikovaně.

```
saving chunk 0 size 2910647bytes in 375 chunks from total 5 files
saved chunk 0 total output size 2910647bytes in 375 chunks from total 5 files
```

Ukládání souboru, všiměte si, jak sedí velikosti vzhledem k ukládání jednotlivých dílků (2915252bytes in 375 chunks. Total: 2910647) Následuje další soubor

```
header list have 2 nodes!
....
loaded content count: 428
unexpected content count: 375
ok- differs
loaded content size: 2915333
unexpected content size: 2910647
ok- differs
```

Obsahy seliž korektně liší

```
saving chunk 4 size 2915333bytes in 428 chunks from total 5 files
saved chunk 4 total output size 14568788bytes in 2104 chunks from total 5 files
```

Opět sedí s výstupem splitteru: Total: 14568788 100%

java -jar XMLcompare.jar

```
checking concated.xml
checking demo-in.xml
f1: /home/jvanek/Desktop/demo/concated.xml
f2: /home/jvanek/Desktop/demo/demo-in.xml
```

Pokud zde nastane chyba, pravděpodobně soubor/y neexistuje.

```
Content is comapred
failFast: false
silent: false
opening parser for: /home/jvanek/Desktop/demo/concated.xml
opening parser for: /home/jvanek/Desktop/demo/demo-in.xml
Checking version: 1.0 against 1.0
OK : versions equals
Checking encoding: UTF-8 against UTF-8
OK : encodings equals
Checking processing instructions
OK : number of processing instructions equals 1
OK : instructions xml-stylesheet=type="text/xsl" href="preview.xsl" are similar
Entering root
```

```

*Checking element: STEP-ProductInformationTranslation against STEP-ProductInformationTranslation
OK : names equals (STEP-ProductInformationTranslation)
Checking attributes
OK : number of attributes equals 8
OK : attribute ContextID - EU-eng-Elfa are similar
OK : attribute ExportContext - EU-eng-Elfa are similar
OK : attribute ExportTime - 2011-02-09 21:50:27 are similar
OK : attribute PendingFileName - Pj050_Chapter1_ENToRU_20110209_EU-rus-Elfa.xml are similar
OK : attribute WorkspaceID - Approved are similar
OK : attribute translationTarget - rus are similar
OK : attribute wfmProcessID - 736105 are similar
OK : attribute wfmProcessTemplateName - STRAXML are similar
Checking childrens
OK : number of childrens equals 3
*Checking element: Qualifiers against Qualifiers
OK : names equals (Qualifiers)
Checking attributes
OK : number of attributes equals 0
Checking childrens
OK : number of childrens equals 9
*Checking element: Qualifier against Qualifier
OK : names equals (Qualifier)
Checking attributes
OK : number of attributes equals 1
OK : attribute ID - Qualifier root are similar
Checking childrens
OK : number of childrens equals 1
*Checking element: DimensionPointLink against DimensionPointLink
OK : names equals (DimensionPointLink)
Checking attributes
OK : number of attributes equals 1
OK : attribute DimensionPointID - Qualifier root are similar
Checking childrens
OK : number of childrens equals 0
Checking content
OK : content equals - lines f1: 3, f2: 5 :
leaving element: DimensionPointLink and DimensionPointLink
leaving element: Qualifier and Qualifier
*Checking element: Qualifier against Qualifier
OK : names equals (Qualifier)
Checking attributes
.....
OK : number of childrens equals 0
Checking content
OK : content equals - lines f1: 99434, f2: 101538 : 01;SGx;13-67
leaving element: Value and Value
leaving element: Values and Values
leaving element: Product and Product
leaving element: Products and Products
leaving element: STEP-ProductInformationTranslation and STEP-ProductInformationTranslation
XMLs are same nebo total errors=NUMBER

```

odsazení simuluje vnoření v XML stromu.

Pokud na některém řádku není OK, ale ERROR nebo WARNING, je tam uveden důvod a řádek v souboru (f1 nebo f2 – viz hlavička výpisu)kde k chybě došlo. Tento koparátor je relativně hloupý, a i malý rozdíl v xml může vyprodukovat obrovské množství errorů. Většinou stačí opravit první z nich

```
Checking content
```

OK : content equals - lines f1: 3, f2: 5 : Je vztažen pouze pokud je ignoreContent vypnuto (porovnání pouze struktury xml – vhodné pro přeložené texty)

4. tutorial a xpath

Tato kapitola zkusí provést uživatele krok za krokem základním použitím

1) zkontrolovat zda je nainstalována java - viz troubleshooting. Pokud ne nainstalovat.

2) Buďto použít globální kopii nebo si vytvořit lokální tohot XML programu a zkontorlvat zda má správné soubory:

├─ default.conf - nepovinný, nicméně může být užitečný, případně zkontorvast jeho obsah co vlastně obsahuje

├─ lib

| └─ axiom-api-1.2.9.jar

| └─ axiom-c14n-1.2.9.jar

| └─ axiom-dom-1.2.9.jar

| └─ axiom-impl-1.2.9.jar

| └─ commons-collections-3.2.1.jar

| └─ commons-collections-3.2.1-javadoc.jar

| └─ commons-collections-3.2.1-sources.jar

| └─ commons-collections-testframework-3.2.1.jar

| └─ commons-logging-1.1.1.jar

| └─ geronimo-activation_1.1_spec-1.0.2.jar

| └─ geronimo-javamail_1.4_spec-1.6.jar

| └─ geronimo-stax-api_1.0_spec-1.0.1.jar

| └─ jaxen-1.1.1.jar

| └─ wstx-asl-3.2.9.jar

└─ XMLcompare.jar

└─ XMLjoiner.jar

└─ XMLsplitter.jar

3) zajistit /adresar/se/soubor.xml který chceme rozdělit

Pro potřeby našeho tutoriálu necht' vypadá takto:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="preview.xsl"?><STEP-ProductInformationTranslation ContextID="EU-eng-Elfa" ...
wfmProcessID="736105" wfmProcessTemplateName="STRAXML">
  <Qualifiers>
    <Qualifier ID="Qualifier root">
      <DimensionPointLink DimensionPointID="Qualifier root"/>
    </Qualifier>
  ...
  <Qualifier ID="AND4926079">
    <DimensionPointLink DimensionPointID="AllCountries"/>
    <DimensionPointLink DimensionPointID="All ERP"/>
  </Qualifier>
</Qualifiers>
<UnitList>
  <Unit ID="unece.unit.MTR">
    <Name QualifierID="Qualifier root">m</Name>
  </Unit>
  ...
</Unit>
</UnitList>
<Products>
  <Product ID="15514" ParentID="15499" UserTypeID="Familie">
    <!-- Preview File : 15514.htm-->
    <Name QualifierID="eng">
      <TranslatableText>Silicon Small Signal Diodes, Axial Lead</TranslatableText>
    </Name>
    <ClassificationReference ClassificationID="transitest"/>
    <ClassificationReference ClassificationID="l0_global_01"/>
    <Values>
      <Value AttributeID="5359" ... QualifierID="std.lang.all"><character name="ref ... separator="&quot;&quot;"/>...</Value>
      <Value AttributeID="5547" AttributeName="Titel Familie" QualifierID="eng">
        <TranslatableText>Silicon Small Signal Diodes, Axial Lead</TranslatableText>
      </Value>
      <Value AttributeID="Bezeichnung_text" AttributeName="Название продукта" InheritedFrom="15499" QualifierID="eng">Small
Signal Diode</Value>
    ...
    <MultiValue AttributeID="XMLTransInclude" AttributeName="Language Code">
      <Value InheritedFrom="DC-47349" QualifierID="Qualifier root">10 GLOBAL=eng,ger,fre,ita,pol,hun,swe,dan,fin,cze,rus</Value>
    </MultiValue>
    <Value AttributeID="creation_date_as400" AttributeName="Creation Date AS400" QualifierID="Qualifier root">2002-05-21</Value>
    <Value AttributeID="cpmpncat2011_lov" AttributeName="Chapter-PM-PageNumber, CAT2011" InheritedFrom="DC-52993"
QualifierID="Elfa01">01;SGx;13-67</Value>
  </Values>
</Product>
</Products>
</STEP-ProductInformationTranslation>
```

4) Zamyslet se a vytvořit xpath hlavičky, obsahu a patičky.

Například v našem xml cheme jako hlavičku všechny elementy `Qualifier` a `Unit`. Patičku žádnout a jako obsah všechny elementy `Product`. Výsledek viz bod 7.

5) zamyslet se jak soubor rozdělit. Aby každý soubor měl maximálně 1mega? Vstup má 20mb, co hotakto rozdělit na 20 souborů?

6) vytvořit si /muj/konfigurak.conf (viz kapitola a 1 a 2) a zapsat do něj základní informace:

```
headerXPath=/STEP-ProductInformationTranslation/Qualifiers | /STEP-ProductInformationTranslation/UnitList
contentXPath=/STEP-ProductInformationTranslation/Products/Product
L=1
```

7) Spustit XMLplitter

```
java -Xmx1g -jar /tam/kde/je/ulozen/program/XMLsplitter.jar -config=/muj/konfigurak.conf  
divSize=20 -i=/adresar/se/soubor.xml -o=/tam/kam/ulozit/vysledek/
```

Vstup konzultovat s kapitolou 1, vystup s kapitolami 3 a 5. S trochou štěstí tento příkaz vygeneruje soubory

```
/tam/kam/ulozit/vysledek/soubor_part0001.xml  
...  
/tam/kam/ulozit/vysledek/soubor_part0020.xml
```

7) nechat zpracovat ^^ soubory tak, proč byli rozděleny

8) spojit soubory:

```
java -Xmx1g -jar /tam/kde/je/ulozen/program/XMLjoiner.jar -config=/muj/konfigurak.conf „-  
i=/tam/kam/ulozit/vysledek/soubor_part.+xml“ -checkHeaders -sortA -o=/tam/kam/ulozit/vysledek.xml
```

Vstup konzultovat s kapitolou 1, vystup s kapitolami 3 a 5. S trochou štěstí tento příkaz vygeneruje soubor

```
/tam/kam/ulozit/vysledek.xml
```

9) Možná Kontrola výsledku

Pro testovací účely je možné před překladem vytvořit opětovně složené soubory (bod 8, jen spuštěn před zpracováním a s výstupem např do **tam/kam/ulozit/vysledek_original.xml** (parametr -o...)

Pustit na tyto aplikace XMLcompare:

```
java -Xmx1g -jar /tam/kde/je/ulozen/program/XMLcomapre.jar /adresar/se/soubor.xml tam/kam/ulozit/vysledek_original.xml -silent  
java -Xmx1g -jar /tam/kde/je/ulozen/program/XMLcomapre.jar /adresar/se/soubor.xml tam/kam/ulozit/vysledek.xml -silent  
-ignoreContent  
java -Xmx1g -jar /tam/kde/je/ulozen/program/XMLcomapre.jar tam/kam/ulozit/vysledek.xml tam/kam/ulozit/vysledek_original.xml  
-ignoreContent -silent
```

Pokud libovolný z těchto příkazů vypíše chybu srovnání (kapitola 1,3,5) pustit bez parametru -silent a najít zádrhel. Pokud ne je ve **/tam/kam/ulozit/vysledek.xml** výsledný přeložený spojený soubor.

5. troubleshooting

Memory – *Not enough memory* nebo *Cant create JavaHeap Space* je způsobeno pravděpodobně chybějícím -Xmx přepínačem pro javu. Oprava – java -Xmx1g -jar ... Xmx je maximální velikost ram jakou si java alokuje. *Can not initiate virtualmnachie* naopak znamená že stroj má méně ram než je specifikováno v Xmx. Zkuste napříkald zmenšit na -Xmx512m

regex – Pro snadnější vyhledání souborů pro složení je možnost aplikovat regex. Kromě syntaxe samotné - <http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html> - je třeba myslet i na escape sekvence příkazové řádky. Tzn správně uvozovkovat a escapovat. Na windows byl detekován problém se znakem ‚\‘ Tento se používá k escapování escape znaků a je třeba ho správně odescapovat... Většinou stačí použít „i=/moje/cesta/soubor_delimiter1.+delimiter2.xml“ nebo „-i=/moje/cesta/soubor.*[0-9]+.*xml“ či mírně sofistikovanější kombinace.

java - na cílovém systému musí být nainstalována java – at už IBM, SUN, Oracle nebo OpenJDK, či jakákoliv alternativní. Jednoduchý důkaz je spuštění java -version. Pokud to napíše něco smysluplného (čili cokoliv kromě program neexistuje, neplatný příkaz, read from address FFFFFFFF...) je java nainstalována korektně.

sax, valid, wellformed... vstupní XML je rozbité nebo vůbec není XML. Je třeba prostudovat vstupní soubory.

nullpointer exception – pravděpodobně neexistující nebo rozbité vstupní/vstupní soubory/adresáře

ClassNotFoundException, MainClass not found - Je třeba zkontrolovat že v adresáři s aplikací anebo v pracovním adresáři je adresář lib se 14 soubory: axiom-api-1.2.9.jar , axiom-c14n-1.2.9.jar , axiom-dom-1.2.9.jar , axiom-impl-1.2.9.jar , commons-collections-3.2.1.jar , commons-collections-3.2.1-javadoc.jar , commons-collections-3.2.1-sources.jar , commons-collections-testframework-3.2.1.jar , commons-logging-1.1.1.jar , geronimo-activation_1.1_spec-1.0.2.jar , geronimo-javamail_1.4_spec-1.6.jar , geronimo-stax-api_1.0_spec-1.0.1.jar , jaxen-1.1.1.jar , wstx-asl-3.2.9.jar , A že existuje tké je třeba zkontrolovat že existuej jar kterýje spouštěn. Eg: java -jar XML.comapre.jar

xpath -

<http://www.w3schools.com/xpath/default.asp>

http://www.w3schools.com/xpath/xpath_syntax.asp

<http://www.w3.org/TR/xpath/>

... aneb good luck:)