**PART1**

**1) Write a short profile of each file we have given you.**

**FileA.xml description**

**MD5 checksum D30CBA6B00308A87FA3A384799C5FAF7**

This file seems to represent the data for ConsumerComplaints. The parent tag is consumerComplaints which contains multiple complaints pertaining data to multiple complaints.

Each complaint has an id associated with it as an attribute that is unique to every element. Each complaint has events,product,issue, consumerNarrative(which is optional),company,submitted and response. Some of the id don’t follow this exact order of tags.

Event has a type and date associated with it.

Product contains detail about the product and have two child tags productType and subproduct(optional)

Issue contains issueType and may contain subissue

Optionally the complaint may contain consumerNarrative in cases of some complaint.

This is followed by company details containing companyName , companyState and companyZip.

Then we have a submitted element that has attribute via having the details on how the complaint was submitted.

Lastly we have response which indicates timely and consumerDisputed as attributes. The response contains a publicResponse(optional) and responseType as well.

**FileB.xml description**

**MD5 checksum 47677272E76E1F4332AFE859347C8695**

This file also has data related to consumer complaints. The overall structure of this document is very much similar to FileA.xml but there are few differences. Unlike FileA this File has irregular line breaks.

Each complaint has an id as well as submissionType as attributes. Unlike FileA where submitted was an element we have a submissionType attribute. Each complaint has events,product,issue,consumerNarrative(in some cases),company and response. More or less these elements are similar to those present in FileA.xml.

Event has a type and date associated with it.

Product contains detail about the product and have two child tags productType and subproduct(optional)

Issue contains issueType and may contain subissue

Optionally the complaint may contain consumerNarrative in cases of some complaint.

This is followed by company details containing companyName , companyState and companyZip.

Lastly we have response which indicates timely and consumerDisputed or not as attributes. The attributes value are different from that of FileA.xmlThe response contains a publicResponse(optional) and responseType as well.

There is a tag <submitted/> present in the file which seems to be erroneous.

**2) Create a DTD for each XML file**

DTD for both the files have been created (internal DTD) and the files are

Consumer\_Complaints\_FileA\_withDTD.xml

Consumer\_Complaints\_FileB\_withDTD.xml

These have been validated for the provide files

Consumer\_Complaints\_FileA.xml and Consumer\_Complaints\_FileB.xml

From <http://xmlvalidator.new-studio.org/>

The DTD files for the provided files have been generated keeping in mind the structure of these two files as described in part1

**3)**

The files that we get after canonicalization of files Consumer\_Complaints\_FileA.xml and Consumer\_Complaints\_FileB.xml are FileA.xml and FileB.xml respectively.

**Steps for canonicalization**

Before we move forward to canonicalization process , we need to make following changes to the documents

1. FileA
   1. We will be removing element submitted that is present inside a complaint and we will instead add an attribute to complaint similar to FileB which is submissionType. The reason for doing is because the element submitted doesn’t have any data and also it makes more sense to have it as an attribute because there will be limited number of values for it.
   2. Replace values Y as yes and N as no. The reason for this is because Yes and No are more readable and don’t introduce any ambiguity.
2. FileB
   1. We will fill the missing values of submissionType attribute for some of the complaint ids by taking these values from FileA
   2. We will remove the tag </submitted> from our file as it has no matching tag and seems to be erroneous.
   3. Replace values Y as yes and N as no. The reason for this is because yes and no are more readable and don’t introduce any ambiguity.
   4. Remove internal DTD and remove &redaction; and replace it with XXXX wherever required.
   5. Add missing values for timely attribute from FileA for two ids

**Above changes can be seen with the github commit** [**https://github.com/elnin09/datacuration/commit/834775f034040fa0e47553ee97a0543397a94c79#diff-985959785319747668373cc6dee294b11db782b03cdd90a2851fbdc0637c6b7b**](https://github.com/elnin09/datacuration/commit/834775f034040fa0e47553ee97a0543397a94c79#diff-985959785319747668373cc6dee294b11db782b03cdd90a2851fbdc0637c6b7b)

1. **Convert to a single character encoding and normalize line ends.**
   1. Both files have UTF-8 encoding.
   2. For normalizing line ends in VSCode I used <https://marketplace.visualstudio.com/items?itemName=sohamkamani.code-eol> extension which helped in normalizing line ends
2. **Remove all comments, tabs, non-significant spaces, etc.**
   1. Removed all the non-significant spaces from both the files and tabs. This has been done for both the files.
   2. For comments there were only one comment in FileB that was removed

<!-- Note: Sally modified this event on 2015-05-06 -->

1. **Propagate all attribute defaults indicated in the schema to the elements themselves**
   1. There were no attributes present in any of the files with default values so we don’t need to change anything in the files for this issue.
2. **Put attribute/value pairs on elements in alpha order**
   1. For element event in both the files, put date attribute before type attribute for both files
   2. For element response put consumerDisputed attribute before timely for both files
3. **Expand all character references**
4. **Remove any internal schema or declarations**

This has been removed from FileB which had internal declaration and internal DTD

Now above changes made can be checked using this commit in FileA.xml and FileB.xml

[**https://github.com/elnin09/datacuration/commit/9a3552b4d1917684f73d8b58f28fa89fe2a7fe9f**](https://github.com/elnin09/datacuration/commit/9a3552b4d1917684f73d8b58f28fa89fe2a7fe9f)

1. **Now test to see if character sequences are identical.**

The files are FileA.xml and FileB.xml after above changes and the MD5 checksum from the link

[**http://onlinemd5.com/**](http://onlinemd5.com/)

is A26D2C1C1962570D68539F5CD932C78F for FileA.xml

is A26D2C1C1962570D68539F5CD932C78F for FileB.xml

the checksum is same for both the files after steps 1-6 and hence we can see that the data

**4) Create and document the DTD of the final, canonicalized data file, from step 3 above.**

The final files in previous step were FileA.xml and FileB.xml which are equivalent. The file with internal DTD and the canonicalized data file is FileC.xml the file have been validated using http://xmlvalidator.new-studio.org/

Explanation/documentation of the DTD

|  |  |
| --- | --- |
| DTD | Comments |
| <!DOCTYPE consumerComplaints [  <!ELEMENT consumerComplaints (complaint\*)>  <!ELEMENT complaint ((event\*,product,issue,consumerNarrative?,company,response)  |(company,event,issue,product,event,response))>  <!ATTLIST complaint id CDATA #REQUIRED>  <!ATTLIST complaint submissionType CDATA #REQUIRED>  <!ELEMENT event (#PCDATA)>  <!ATTLIST event type (received|sentToCompany) #REQUIRED>  <!ATTLIST event date CDATA #REQUIRED>  <!ELEMENT consumerNarrative (#PCDATA)>  <!ELEMENT product (productType,subproduct?)>  <!ELEMENT productType (#PCDATA)>  <!ELEMENT subproduct (#PCDATA)>  <!ELEMENT issue (issueType,subissue?)>  <!ELEMENT subissue (#PCDATA)>  <!ELEMENT issueType (#PCDATA)>  <!ELEMENT company (companyName,companyState,companyZip)>  <!ELEMENT companyName (#PCDATA)>  <!ELEMENT companyState (#PCDATA)>  <!ELEMENT companyZip (#PCDATA)>  <!ELEMENT response (publicResponse?,responseType)>  <!ELEMENT publicResponse (#PCDATA)>  <!ATTLIST response timely (yes|no) #REQUIRED>  <!ATTLIST response consumerDisputed (yes|no) #REQUIRED>  <!ELEMENT responseType (#PCDATA)>  ]> | The root of the document ConsumerComplaints contains multiple complaints  Complaint in itself has multiple elements. There are two set of possibilities that’s why we have two sequence of elements for complaint.  We will have attributes id and submissionType which is required instead of implied because we have data for this attribute for all the complaint.  Then we have event element with attributes type and data.  We have consumerNarrative which has normal char data.  Product contains producttype or may contain subproduct hence we have ? for subproduct  productType and subproduct are elements with char data  Issue contains issueType or may contain subissue hence we have ? for subissue  issueType and subissue are elements with char data  Element company contains subelements companyName, companyState and companyZip.  companyName, companyState and companyZip are elements with char data  Element response may contain publicResponse followed by responseType that’s why we have ? for publicResponse  Response also have attributes timely and customerDisputed with values either yes or no.  Elements responseType and publicResponse contain char data |

**5) Be sure that all of your files validate as they will be assessed for compliance to their DTDs. You must test whether your document will validate against your DTD here:** [**http://xmlvalidator.new-studio.org/**](http://xmlvalidator.new-studio.org/)

The 3 files with internal DTD have been validated FileC.xml, Consumer\_Complaints\_FileB\_withDTD.xml and Consumer\_Complaints\_FileA.xml.

**6) Be sure that all of your files validate as they will be assessed for compliance to their DTDs. You must test whether your document will validate against your DTD here:** [**http://xmlvalidator.new-studio.org/**](http://xmlvalidator.new-studio.org/)