User guide for CDA tools

Guide for Users

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe the process of generating CDA test data using the command line tools in 4s-cda-builders. Likewise, the document describes the usage of input files and the contents of these files.

The generated test data conforms to the danish profiles specified by MedCom of PHMR, QRD and APD CDA document types.

1.2 Reading guide

This document is directed towards developers and architects that will be using CDA tools to build, generate and/or upload CDA documents that conforms to the Danish profile of the types PHMR, QRD, APD and QFDD.

It is assumed that the reader is familiar with Maven, Java and CDA.

1.3 Document History

Version	Date	Responsible	Description
0.1	08.11.2016	Systematic	Initial draft. Still a lot of todos
0.9	14.11.2016	Systematic	Reference support in 01_patient.csv Upload section ready Proper document
1.0	21.11.2016	Systematic	Final version
1.1	28.11.2016	Systematic	Fix typos in commandline examples
1.2	06.12.2016	Systematic	New columns for patient info Fix typos
1.3	10.01.2017	Systematic	Minor fixes. Generation of PHMR xml files corrected.
1.4	24.02.2017	Systematic	Added usage of legal authenticator Changed parameter usage for generators
1.5	24.03.2017	Systematic	Corrected for CDA tools 1.0.0 release Added optional parameter for cda uploader
1.6	30.03.2017	Systematic	Corrected for CDA tools 1.1.0 release Description of CDA Downloader
1.7	10.04.2017	Systematic	Corrected for CDA tools 1.1.1 release Description of encoding property for downloader
1.8	10.05.2017	Systematic	Corrected for CDA tools 1.1.2 release Description of cron notation for CDA test data generation
1.9	08.06.2017	Systematic	Corrected for CDA tools 1.2.0 release

Version	Date	Responsible	Description
2.0	18.09.2018	Systematic	Test data generation for appointment documents (APD) added.
2.1	15.10.2018	Systematic	Branch for Pilot only[1]. Replaced appointment status values in section 2.2.3.1 to those in APD v.1.1. Added fields apd_oprettelsestidspunkt and apd_aftale_periode_ikke_eksakt to section 2.2.3.
2.2	22.10.2018	Systematic	Dangling reference fixed for healthcareFacilityCode in table with CDA Downloader's parameters.

1.4 Definitions and References

Definition	Description
APD	Appointment Document
CDA	Clinical Document Architecture
KIH	Klinisk integreret hjemmemonitorering
PHMR	Personal Healthcare Monitoring Report
QRD	Questionnaire Response Document
QFDD	Questionnaire Form Definition Document

[1] Throughout the document, elements marked as available for the Pilot only will cause APD XML to be generated as described for the particular element in http://svn.medcom.dk/svn/releases/Standarder/Komplekse%20Forloeb/Aftaler/Implementeringsguide/IG_Aftale_CDA-APD_v07.docx.

2 Input Files

In the following sections the content for the input files is described.

Please note: using Excel to edit these may result in improper csv format, wrong character encoding and SOR codes getting transformed into scientific notation (3,68E+14).

2.1 CDA Header Files

01_organization.csv

This file contains general data about an organization and the healthcare professional associated to the organization. In the CDA header this data is used for the author of the document.

An example is shown in Table 15. Each field is described in the following table.

Field	Description
organisation_afdeling_sor	SOR code of the department where the healthcare professional is located
organisation_afdeling_stednavn	Name of the department
organisation_afdeling_vejnavn	Address of the department
organisation_afdeling_postnummer	Postal code of the department
organisation_afdeling_by	City of the department
organisation_ansvarlig_fornavn	First name of the healthcare professional
organisation_ansvarlig_efternavn	Last name of the healthcare professional
organisation_sor	SOR of the parent organization of the department
organisation_navn	Name of the parent organization of the department

Table 1

01_legalauthenticator.csv

This file contains data about an organization and the healthcare professional associated to the organization. In the CDA header this data is used for the legal authenticator of the document.

An example is shown in Table 16. Each field is described in the following table.

Field	Description
legal_organisation_afdeling_sor	SOR code of the department where the healthcare professional is located
legal_organisation_afdeling_stednavn	Name of the department

Field	Description
legal_organisation_afdeling_vejnavn	Address of the department
legal_organisation_afdeling_postnummer	Postal code of the department
legal_organisation_afdeling_by	City of the department
legal_organisation_ansvarlig_fornavn	First name of the healthcare professional
legal_organisation_ansvarlig_efternavn	Last name of the healthcare professional
legal_organisation_sor	SOR of the parent organization of the department
legal_organisation_navn	Name of the parent organization of the department

Table 2

01_custodian.csv

Custodian data about the organization and the healthcare professional associated to the organization. An example is shown in Table 17.

Field	Description
custodian_sor	SOR code of the custodian organization
custodian_stednavn	Name of the custodian organization
custodian_afdeling_navn	Name of the specific department in the organization
custodian_postnummer	Postal code of the custodian organization
custodian_by	City name of the custodian organization

Table 3

01_patient.csv

Contains the test patients for whom the documents are generated for. An example is shown in Table 18.

Field	Description
patient_cpr	The social security number of the patient
patient_fornavn	First name of the patient
patient_efternavn	Last name of the patient
patient_vejnavn	Address of the patient
patient_postnummer	Postal code of the patient
patient_by	City of the patient
patient_tlf	Phone number for patient

Field	Description
patient_mail	E-mail address for patient
referencer	'Ja' or 'nej' value. If 'ja' then all PHMR documents generated for this patient will have a reference to a generated QRD document for this patient.

Table 4

2.2 Data Specific Files

2.2.1 PHMR specific files

PHMR specific data files are have a file name of "02_phmr_målinger*.csv" where the star is a wildcard for extra characters.

2.2.1.1 Template input file

A template for PHMR measurements generation is specified by each line representing a measurement and a range of possible measurement values. Each line has a count column that defines how many of this type of measurement should be generated per document. An example is shown in Table 19.

Field	Description
Phmr_tidspunkt	This field is overriden when using data auto generation. See usage of config.properties The date and time at which the data is created. Use ISO8601 datetime formatting in this field. See example section.
Phmr_type	NPU code system name (http://svn.medcom.dk/svn/releases/Standarder/POCT-Hjemmemonitorering/Oversigtoverhjemmem%c3%a5linger.xlsx)
Phmr_enhed	Unit of the measurement
Phmr_værdi	Range of valid measurements for this type of measurement. Set as minimum_value#maximum_value where the '#' is used as a separator. Example: 50#66 Here the value is randomly picked between 50 and 66.
Phmr_medicalDeviceCode	Device code of the device used to perform the measurement (https://svn.medcom.dk/svn/releases/Standarder/POCT-Hjemmemonitorering/MedCom%20Instrument%20Codes.xlsx)
Phmr_medicalDeviceDisplayName	Display name of the device used to perform the measurement (https://svn.medcom.dk/svn/releases/Standarder/POCT-Hjemmemonitorering/MedCom%20Instrument%20Codes.xlsx)

Field	Description
Phmr_manufactorModelName	Model name of the device used to perform the measurement (https://svn.medcom.dk/svn/releases/Standarder/POCT-Hjemmemonitorering/MedCom%20Instrument%20Codes.xlsx)
Phmr_maalinger_patient	How many measurements of this type is generated per patient

Table 5

2.2.2 QRD template file

A template for QRD answers to a QFDD. Each patient will get QRD response based on this template. Note that if multiple QRD's are created for one patient from the same template, the timestamp will be the same for the generated QRD's. An example is shown in Table 20.

Field	Description
Qrd_dok_titel	Title of the QRD document
Qrd_skabelon	Reference id to the QFDD document for which this QRD is a response to
Qrd_spørgsmål_id	Reference id to the QFDD question for which this answer is a response to.
Qrd_type	One of following types: Numeric Text Multiple Analog Discrete These types matches the types in the QRD specification
Qrd_svar	The user provided answer range for which the generator will create an answer. This depends on the qrd_type field. Examples: Numeric: 1#50 (minimum#maximum) Text: text1#text2 (list of texts) Multiple: very much#ok (list of choices) Analog: 0#100 (minimum#maximum) Discrete: ok#not ok (list of values)
Qrd_titel	Title of the question
Qrd_vejledende_tekst	Additional text for the user
Qrd_spørgsmål_text	The actual question

Field	Description
Qrd_antal	Ignored for manually crafted files

Table 6

2.2.3 APD specific files

Beyond the declaration of patient, author, and other data in the header, an appointment document consists of the following elements:

1. Base-data: Appointment id, status, creation date
2. Time-data: Specifying when the appointment will take place.
3. Reason-data: The reason for the appointment, the performer and the location of the appointment.
4. Request-data: Requester of the appointment.

The data for these elements can be set in the following four input csv files:

- 02_apd_base.csv (base-data plus optional time-data and optional reason-data)
- 02_apd_time.csv (appointment start and end time)
- 02_apd_reason.csv
- 02_apd_requester.csv

An appointment will either be constructed by one row from the 02_apd_base.csv file and a random row from each of the other three input files, or by one row from the 02_apd_base.csv file and a random row from the 02_apd_requester.csv file. This depends on whether the optional time and reason data has been set for a given row from the 02_apd_base.csv file.

The optional fields provide a mechanism for both being very specific and also being able to churn out a great number of appointments. Specifying the optional fields in 02_apd_base can be used for describing the life cycle of appointments (created, changed, cancelled) explicitly Leaving out the optional fields can be used when just needing to produce appointments in masse. When leaving optional fields out, the values are taken from the other input files.

2.2.3.1 02_apd_base.csv

Each row from the 02_apd_base.csv will result in at least one appointment (unless the time-data specifies something different, see 2.2.3.2). A row in the 02_apd_base.csv file consists of required base data (appointment id, etc.) but can also contain time data (optional) and reason data (optional).

If time data has not been set in the apd_base file, a random row will be fetched from the 02_apd_time.csv file and used instead. Likewise, if no reason data has been set in the apd_base file, a random row will be fetched from the 02_apd_reason.csv file and used instead.

The following table show the input values for rows in the 02_apd_base.csv file and an example can be seen in Table 21.

Field	Description
apd_id	Id of the appointment
apd_status	Status of the appointment, can be ACTIVE, ABORTED, SUSPENDED or COMPLETED.

Field	Description
apd_oprettelsestidspunkt	Creation date of the appointment. This value is also used as the date all participants were attached to the appointment.
Time-data (optional)	See 2.2.3.2
Reason-data (optional)	See 2.2.3.3

Table 7

2.2.3.2 02_apd_time.csv

The 02_apd_time.csv rows are used to set the start and end time for an appointment. However, there are two ways of setting these values:

- Setting apd_aftale_start and apd_aftale_slut. This will result in the creation of *one* appointment file where the start time of the appointment is set to the apd_aftale_start value and the end time of the appointment is set to the apd_aftale_slut value.
- Setting apd_cron and apd_cron_aftale_laengde. Depending on the cron-expression this will create *n* appointment files where each appointment will have the start time specified by the cron-expression and the end time calculated by adding the apd_cron_aftale_laengde value to the start time.

The following table show the input values for rows in the 02_apd_time.csv file and an example can be seen in Table 22.

Field	Description
apd_aftale_periode_ikke_eksakt	Whether the appointment time period is not to be interpreted as exact. Set to TRUE if not exact. Set to FALSE or null if the time period is exact. Note that this column and behavior is available for the Pilot Branch only. When set to not exact, the XML uses the inclusive attribute defined for the pilot.
apd_aftale_slut_ukendt	Whether the appointment end time is unknown. Set to TRUE if unknown, FALSE or null otherwise. Note that this column and behavior is available for the Pilot Branch only. When set to unknown, the XML will include nullFlavor for the end time of the appointment.
apd_aftale_start	Start time of the appointment
apd_aftale_slut	End time of the appointment
apd_cron	A CRON expression used to specify the start time for possibly multiple appointments. Depending on how the cron expression evaluates, each consecutive value is used as start time for a new appointment.

Field	Description
	The CRON notation is an expression for which a series of dates are generated. It is recommended to read the following web page: http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger. Example: The following CRON expression: 0 0 19 ? 1 FRI 2018 evaluates to: “At 19:00:00, on every Friday, in January, in 2018”.
apd_cron_aftale_laengde	Duration of the appointment given in **minutes**. Used to calculate the end time for appointments generated by the cron expression.

Table 8

2.2.3.3 02_apd_reason.csv

The 02_apd_reason.csv file contains a text describing the reason for the appointment, data about the location of the appointment and data about the performer of the appointment. These have been gathered in this file as the reason (subject line) of an appointment is likely to be related to where it is performed (location) and by whom (performer).

The following table show the input values for rows in the 02_apd_reason.csv file and an example can be seen in Table 23.

Field	Description
apd_aftale_emne	Subject line of the appointment
apd_lokation_organisation_id	Id of the location organization. Note that for the Pilot Branch only, this can be empty or null.
apd_lokation_organisation_navn	Name of the location organization
apd_lokation_organisation_tlf	Telephone numbers for the location organization. Multiple can be entered separated by a ‘#’. Note that for the Pilot Branch only, this can be empty or null. When empty or null, the generated XML contains nullFlavor.
apd_lokation_organisation_email	Email addresses for the location organization. Multiple can be entered separated by a ‘#’. Note that for the Pilot Branch only, this can be empty or null. When empty or null, the generated XML contains nullFlavor.

Field	Description
apd_lokation_organisation_adresse_vej	Street name, street number etc. Of the location organization. Multiple values can be entered separated by a '#'. Note that for the Pilot Branch only, this can be empty or null. When empty or null, the generated XML contains nullFlavor.
apd_lokation_organisation_adresse_postnr	Zip code for the address of the location organization. Note that for the Pilot Branch only, this can be empty or null. When empty or null, the generated XML contains nullFlavor.
apd_lokation_organisation_adresse_by	City of the location organizations address. Note that for the Pilot Branch only, this can be empty or null. When empty or null, the generated XML contains nullFlavor.
apd_lokation_organisation_adresse_land	Country of the location organizations address. Note that for the Pilot Branch only, this can be empty or null. When empty or null, the generated XML contains nullFlavor.
apd_udfoerer_id	If of the performer of the appointment.
apd_udfoerer_adresse_vej	Street name, street number etc. Of the performer organization. Multiple values can be entered separated by a '#'.
apd_udfoerer_adresse_postnr	Zip code for the address of the performer organization.
apd_udfoerer_adresse_by	City of the performer organizations address.
apd_udfoerer_adresse_land	Country of the performer organizations address.
apd_udfoerer_tlf	Telephone numbers for the performer organization. Multiple can be entered separated by a '#'.
apd_udfoerer_email	Email addresses for the performer organization. Multiple can be entered separated by a '#'.
apd_udfoerer_pers_identitet_praefix	Prefix for the performing person's identity.
apd_udfoerer_pers_identitet_fornavne	First names for the performing person's identity.
apd_udfoerer_pers_identitet_efternavn	Last name for the performing person's identity.
apd_udfoerer_organisation_navn	Name of the performing organization.

Table 9

2.2.3.4 02_apd_requester.csv

The 02_apd_requester.csv file contains data about the author of the appointment.

The following table show the input values for rows in the 02_apd_requester.csv file, and an example can be seen in Table 29

Field	Description
apd_forfatter_id	Id of the author of the document.
apd_forfatter_adresse_vej	Street name, street number etc. of the author organization. Multiple values can be entered separated by a '#'.
apd_forfatter_adresse_postnr	Zip code for the address of the author organization.
apd_forfatter_adresse_by	City of the author organizations address.
apd_forfatter_adresse_land	Country of the author organizations address.
apd_forfatter_tlf	Telephone numbers for the author organization. Multiple can be entered separated by a '#'.
apd_forfatter_email	Email addresses for the author organization. Multiple can be entered separated by a '#'.
apd_forfatter_pers_identitet_praefix	Prefix for the author person's identity.
apd_forfatter_pers_identitet_fornavne	First names for the author person's identity.
apd_forfatter_pers_identitet_efternavn	Last name for the author person's identity.
apd_forfatter_organisation_navn	Name of the author organization.

Table 10

2.2.3.5 Reusing reason-data

If the appointment id of one row in the 02_apd_base.csv file is identical to the appointment id of the previous row in that file, the reason data will be reused from the previous row. This makes it easy to make different versions of the same appointment, where as an example only the status of the appointment has changed.

2.3 Program Properties

There are several property values that need to be set before the test data generation can start. Note that the default path values match if the data generator executable is located at the parent folder of the "datacreation" folder.

Here is an example of a properties file

```
headerPath=./datacreation/headerfiles/
generatedCdaHeaderDatePath=./datacreation/output/generatedCDADataF
ilesPath/
qrdFilesPath=./datacreation/qrdfiles/
```

```
generatedQrdDataFilesPath=./datacreation/output/generatedQrdDataFi
lesPath/
generatedQrdXmlFilesPath=./datacreation/output/generatedQrdXMLFile
sPath/
phmrFilesPath=./datacreation/phmrfiles/
generatedPhmrDataFilesPath=./datacreation/output/generatedPhmrData
FilesPath/
generatedPhmrXmlFilesPath=./datacreation/output/generatedPhmrXmlFi
lesPath/
apdFilesPath=./datacreation/apdfiles/
generatedApdDataFilesPath=./datacreation/output/generatedApdDataFi
lesPath/
generatedApdXmlFilesPath=./datacreation/output/generatedApdXmlFile
sPath/

#QRD cron notation for when documents will be created:
# Example: At 19.00 every Friday in January 2015
qrd_cron=0 0 19 ? 1 FRI ? 2015

#PHMR cron notation for when documents will be created:
# Example: At 18.00 every Monday, Wednesday and Friday in January
2015
phmr_cron=0 0 18 ? 1 MON,WED,FRI 2015
```

The properties are as follows:

Property	Value
headerPath	Path to the folder where the all 01_*.csv files are located (01_custodian.csv, 01_legalauthenticator.csv, 01_organization_input.csv, 01_patient.csv)
generatedCdaHeaderDatePath	Output folder for the first step in the data generation process. Files are a merge of 01_organization.csv, 01_custodian.csv and 01_patient.csv.
qrdFilesPath	Path to the folder where the 02_qrd_xxxx.csv template files are located. These are input files for the second step in the process.
phmrFilesPath	Path to the folder where the 02_phmr_xxxx.csv template files are located. These are input files for the second step in the process.
apdFilesPath	Path to the folder where the 02_apd_xxxx.csv template files are located. These are input files for the second step in the process.
generatedQrdDataFilesPath	Path to the output folder of the second step for generated QRD csv files. The files are a merge of files from “generatedCdaHeaderDatePath” and “qrdFilesPath”. It’s recommended to manually edit these files if the auto generated answers needs adjusting
generatedPhmrDataFilesPath	Path to the output folder of the second step for generated PHMR csv files. The files are a merge of files from “generatedCdaHeaderDatePath” and “phmrFilesPath”.

Property	Value
	It's recommended to manually edit these files if the auto generated measurements needs adjusting
generatedApdDataFilesPath	Path to the output folder of the second step for generated APD csv files. The files are a merge of files from "generatedCdaHeaderDatePath" and "apdFilesPath". It's recommended to manually edit these files if the auto generated measurements needs adjusting
generatedQrdXmlFilesPath	Output for the final QRD xml files. Takes the csv files from the "generatedQrdDataFilesPath" as input.
generatedPhmrXmlFilesPath	Output for the final PHMR xml files. Takes the csv files from the "generatedPhmrDataFilesPath" as input.
generatedApdXmlFilesPath	Output for the final APD xml files. Takes the csv files from the "generatedApdDataFilesPath" as input.
qrd_cron	Cron notation that generates the dates for which the QRD documents will be created. Note that the dates cannot be created earlier than the year 1980
phmr_cron	Cron notation that generates the dates for which the PHMR documents will be created. Note that the dates cannot be created earlier than the year 1980

The properties qrd_cron and phmr_cron uses the CRON notation which is an expression for which a series of dates are generated. To properly fill out the two parameters it is recommended to read the following web page: http://www.quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger.

3 Data Generation Process

The process of generating data contains several steps described in this section. Overall, there are four steps.

1. Create CDA header
2. Create data content
3. Build PHMR, QRD or APD xml files
4. Upload the files

Make sure that the following projects have been build or the latest versions are available at the maven repository http://artifactory.4s-online.dk/artifactory/net4care. The versions used at release of this document are:

Library	Version	Java
4s-cda-builders	2.0.0	6 or newer
4s-cda-data-generators	1.2.0	7 or newer
4s-net4care-xds-connector	0.2.0	7 or newer

Table 11

To build the projects from source be sure to have the following software installed.

Software	Version	Description
Maven	>= 3.0.3	Commandline build tool for java projects. 3.0.3 was used during development
Java	7 (6 for builders)	Due to the XDS connector being dependent of Java 7 so are the data generators as the uploader uses the XDS connector. Only Java 6 is needed for cda builders

Table 12

Building the 4s-cda-datagenerator is done by:

```
<source-root>mvn clean install
```

The following sections makes references to a config.properties file which is located at:

```
<source-root>\target\datacreation
```

3.1 Create CDA Header

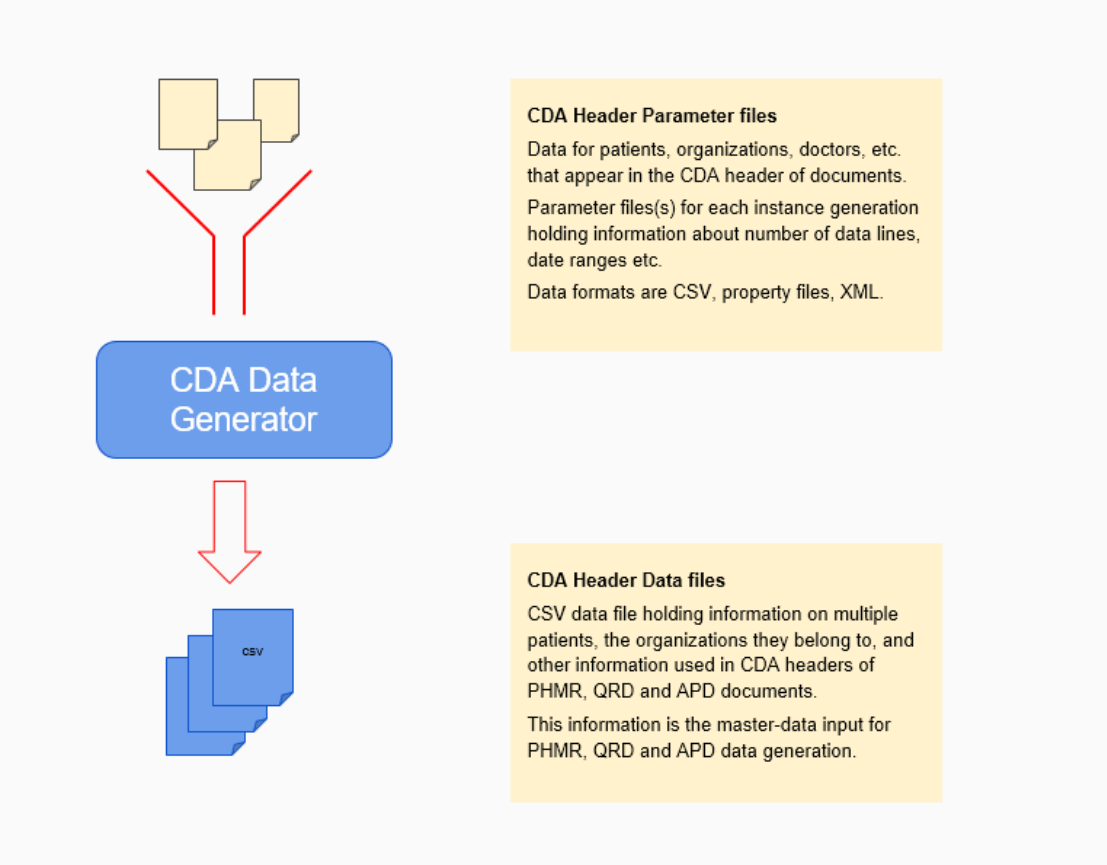

Figure 1: Input header info merged to one csv

The CDA header files consists of three input files. Data for patient, organizations and custodians. In this process, these files are merged in such a way that each patient are randomly matched with multiple organizations from the organizations file and one custodian. The output of this process is a csv file per patient containing all merged data.

3.1.1 Commands

The overall concept of this process is to use a command line to generate the generatedCdaHeaderDateFile.csv file. This file contains a row for each patient. It is possible to modify this after generating it if you want to change address or other things in the CDA header files.

Go to the target folder:

```
<source-root>cd target
```

And run the following command:

```
java -cp datagenerators-1.2.0-jar-with-dependencies.jar
dk.s4.hl7.cda.datacreation.CdaDataGenerator
```

3.2 Create Data Content

Next step is to merge the generated CDA header files with PHMR measurements, QRD answers or APD appointments. This is done be processing the PHMR, QRD and APD template which generates or sets the specified values in input files. The output is a csv file per patient per type of document where each row is a measurement, an answer to a question or an appointment.

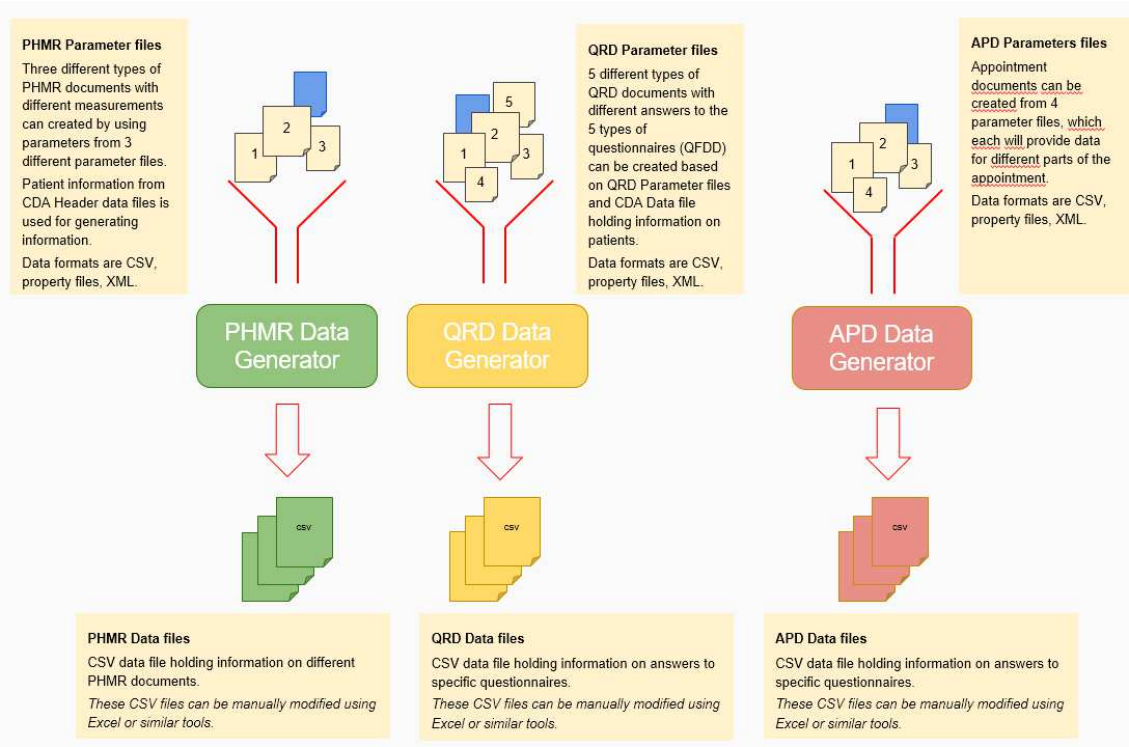

Figure 2: Merge header and PHMR or QRD templates

3.2.1 Commands

The overall concept of this process is to use a command line to generate the CDA header data + QRD, PHMR or APD specific data in a csv file. It is possible to modify this after generating it, so you can change the CDA header data and PHMR, QRD or APD specific data in the csv file.

```
<source-root>cd target
```

The following command creates the QRD generated csv files. The number of generated documents is located in the config.properties as "qrd_antal_documents_per_patient".

```
java -cp datagenerators-1.2.0-jar-with-dependencies.jar
dk.s4.hl7.cda.datacreation.QrdDataGenerator
```

Likewise the PHMR generator has a property "phmr_antal_dage_mellem_maaling" in the config.properties file which is used to auto generate PHMR csv files by the following command:

```
java -cp datagenerators-1.2.0-jar-with-dependencies.jar
```

```
dk.s4.hl7.cda.datacreation.PhmrDataGenerator
```

The following command creates the APD generated csv fils. The number of generated documents depends on the number of lines in one of the input files plus on the given time values for these lines. This is further explained in 2.2.3.:

```
java -cp datagenerators-1.2.0-jar-with-dependencies.jar
dk.s4.hl7.cda.datacreation.ApdDataGenerator
```

3.3 Build PHMR, QRD and APD Xml Files

At this point the csv files contains all the necessary information and the test data generation tool is able to build the actual PHMR, QRD and APD files. Please note the following about this step:

- If references has been selected for the patient the PHMR generator will look up the patients QRD csv files and create a reference to the QRD file. If no QRD csv file for the patient is present no references are made. Hence, if references is needed generate the QRD csv files first.
- There will be generated one QRD file per patient where all answers are included
- The QRD document will have a reference to the QFDD document for which the QRD contains answers for.

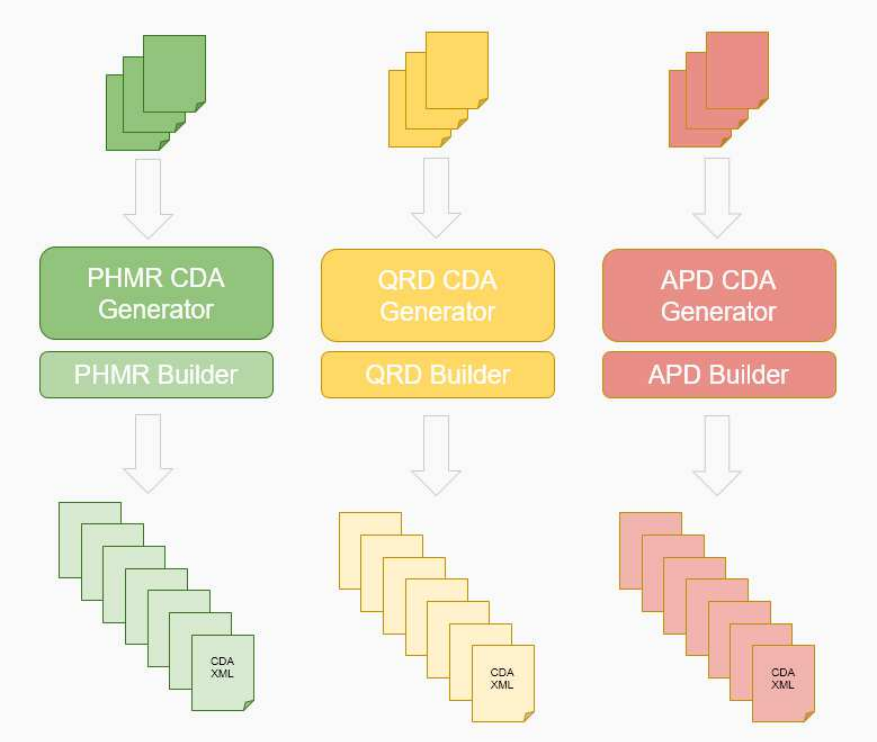

Figure 3: Generate CDA xml from csv

3.3.1 Commands

```
<source-root>cd target
```

The following command generates QRD xml files, one for each QRD csv file generated in the previous step.

```
java -cp datagenerators-1.2.0-jar-with-dependencies.jar
dk.s4.hl7.cda.datacreation.QrdXmlCdaGenerator
```

The following command generates PHMR xml files, one for each PHMR csv file generated in the previous step.

```
java -cp datagenerators-1.2.0-jar-with-dependencies.jar
dk.s4.hl7.cda.datacreation.PhmrXmlCdaGenerator
```

The following command generates APD xml files, one for each APD csv file generated in the previous step.

```
java -cp datagenerators-1.2.0-jar-with-dependencies.jar
dk.s4.hl7.cda.datacreation.ApdXmlCdaGenerator
```

3.4 Upload Files

After generating the PHMR, QRD and APD xml files the tools can be used to upload the CDA documents to an XDS Repository and Registry. This is done by calling the provide and register method at the repository with CDA xml document from a parameter specified folder.

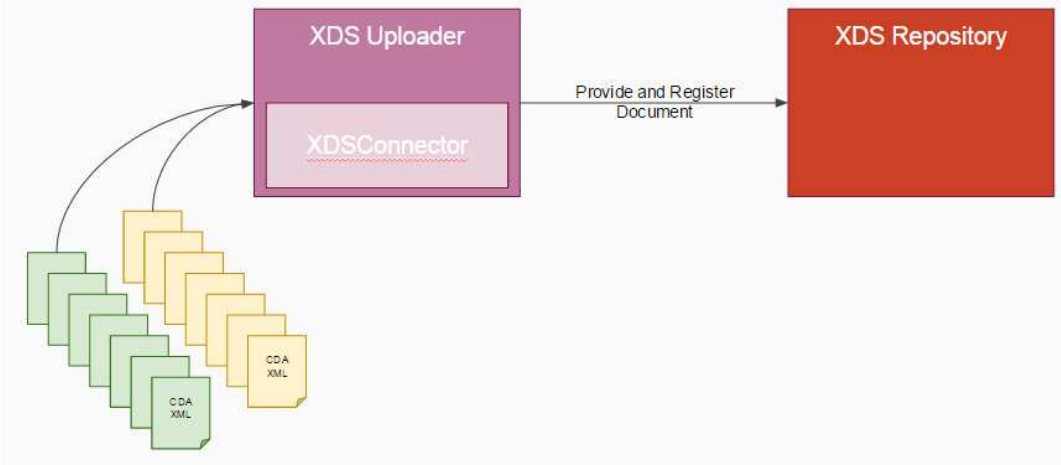

Figure 4: Upload xml files to XDS Repository

3.4.1 Commands

The overall concept of this process is to use a command line application to point out a folder holding CDA documents in xml format, which are uploaded to an XDS Repository and the metadata registered in the XDS Registry. At this step the CDA documents must be generated and ready to upload at a given folder.

Note that the values of attributes and elements in the CDA xml files must be properly escaped before the document can be uploaded successfully.

The 4s-cda-data-generators library is built using the following maven command:

```
<source-root>/mvn clean install
```

After a successful build go to the folder ‘”target/input” and update the upload.properties with proper configuration. Following is an example:

```
# XDS Repository info
xds.repositoryUrl=http://localhost
xds.registryUrl=http://localhost
xds.repositoryId=1.2.3.4
xds.homeCommunityId=1.2.3.4
# Number of documents to batch in a single submissionset per
provide and register call
xds.submissionSet.batchsize=1

# Metadata info for provide and register calls
# For proper values see:
# http://svn.medcom.dk/svn/drafts/Standarder/IHE/OID/DK-
IHE_Metadata-Common_Code_systems-Value_sets.xlsx under the
'DK_IHE_HealthcareFacilityTypeCo' sheet
healthcarefacilitytype.code=22232009
healthcarefacilitytype.codesystem=2.16.840.1.113883.6.96
healthcarefacilitytype.displayname=hospital
```

```
# For proper values see:
# http://svn.medcom.dk/svn/drafts/Standarder/IHE/OID/DK-
IHE_Metadata-Common_Code_systems-Value_sets.xlsx under the
'DK_IHE_practiceSettingCode_VS' sheet
practicesettingscode.code=418112009
practicesettingscode.codesystem=2.16.840.1.113883.6.96
practicesettingscode.displayname=lungesygdomme
```

Next step is to run the actual upload command. The following example will upload PHMR and QRD example documents for the test patient Nancy from the input folder.

```
<source-root>cd target
```

```
java -cp datagenerators-1.2.0-jar-with-dependencies.jar
dk.s4.hl7.cda.upload.CDAUploader -properties
./input/upload.properties -inputfolder ./input
```

As seen in the above example there are two required parameters for the upload command line tool.

Parameter	Description
-properties	Path to the upload.properties file
-inputfolder	Folder where all the CDA xml documents resides which are to be uploaded. The application supports loading from subfolders at the inputfolder location by default.
-retry	Optional parameter the makes the uploader use a retry strategy where the uploader will retry on error

Table 13

3.5 Download Files

The CDA Downloader is a command-line tool in the CDA Tools package that is able to download documents from an XDS Repository. The downloaded documents are stored as xml files in a configurable output folder. The CDA Downloader sends a FindDocuments Stored Query (ITI-18) to the XDS Registry retrieving the metadata that corresponds to the search criterias. All documents found in the metadata are retrived from the XDS Repository by sending a Retrieve Document Set request (ITI-43). The search parameters in the FindDocuments query is configurable by using parameters on the command-line.

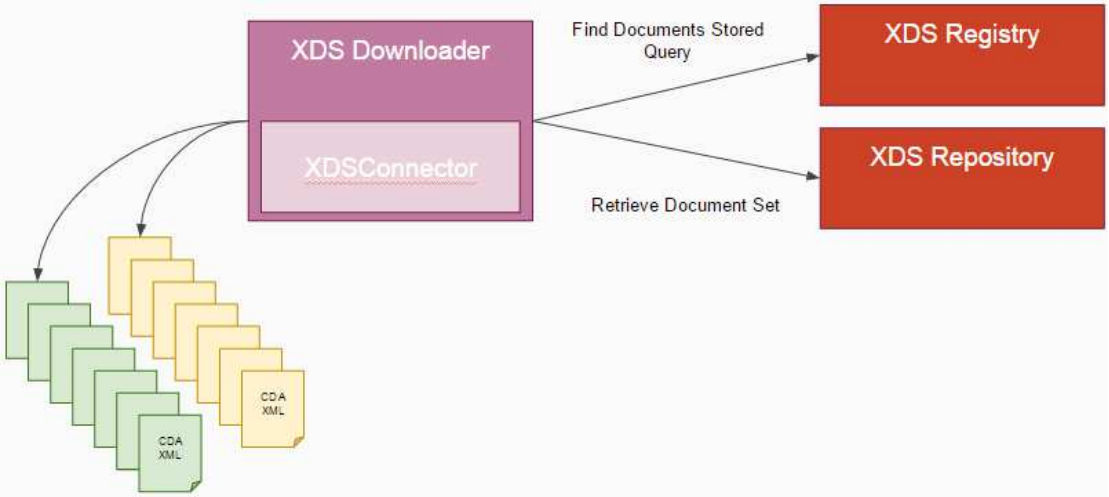

Figure 5 Download documents from a XDS Repository

3.5.1 Commands

The 4s-cda-data-generators library is built using the following maven command:

```
<source-root>/mvn clean install
```

After a successful build go to the folder “”<source-root>/target/download/” and update the download.properties with proper configuration. Following is an example:

```
# XDS info
xds.repositoryUrl=http://localhost
xds.registryUrl=http://localhost
xds.repositoryId=1.2.3.4
xds.homeCommunityId=1.2.3.4
# The output folder where the downloaded documents will be placed.
Note that the folder must exist and the application must have
permissions to write to the folder.
output.folder=./download/output/
# Force encoding when writing documents to file
output.encoding=UTF-8
```

Next step is to run the actual download command. The following example will download all documents created in the period 2016-01-01 to 2016-02-01 for the test patient Nancy to the output folder (./download/output/).

```
<source-root>cd target
```

```
java -cp datagenerators-1.2.0-jar-with-dependencies.jar
dk.s4.hl7.cda.download.CDADownloader -patient 2512489996 -
creationTime 2016-01-01,2016-02-01 -approved
```

The CDA Downloader supports the complete range of parameters defined by the FindDocuments Stored Query. All parameters to the downloader are explained in the following.

Parameter	Optionality	Argument format	Description
-approved	0..1	None	Searches for documents with approved status. At least one of –approved or –deprecated parameters must be specified
-author	0..1	XCN. Eg. ^Andersen^Anders^ Frederik&Ingolf	Search by the author of the document. The format must be XCN (http://www.healthintersections.com.au/?p=1175) as stated by the DK metadata profile: https://svn.medcom.dk/svn/drafts/Standarder/IHE/DK_profil_metadata/
-classCode	0..*	<CODE>,<CODE_SYSTEM> Eg. 12,34	Class code of the document[2]
-confidentialityCode	0..*	<CODE>,<CODE_SYSTEM> Eg. 12,34	Confidentiality code of the document[2]
-creationTime	0..1	<FROM_TIME>,<TO_TIME> See explanation of format after this table	Date time interval of the document creation time
-deprecated	0..1	None	Searches for documents with deprecated status. At least one of –approved or –deprecated parameters must be specified
-eventCode	0..*	<CODE>,<CODE_SYSTEM> Eg. 12,34	Event code of the document[2]
-formatCode	0..*	<CODE>,<CODE_SYSTEM> Eg. 12,34	Format code of the document[2]
-healthcareFacilityCode	0..*	<CODE>,<CODE_SYSTEM> Eg. 12,34	Healthcare facility type code of the document[2]
-help	0..1	None	Print help usage page
-ondemand	0..1	None	Search for on-demand documents
-patient	1..1	10 digits	The social security number of the patient
-practiceSettingCode	0..*	<CODE>,<CODE_SYSTEM> Eg. 12,34	Practice setting code of the document[2]
-properties	0..1	Path URI. /properties/	Path to the file download.properties. Uses default path (./download/ /) if not specified
-serviceStart	0..1	See explanation of format after this table	Date time interval of the document service start time
-serviceStop	0..1	See explanation of format after this table	Date time interval of the document service stop time

[2]It is recommended to use the common codes from MedCom: https://svn.medcom.dk/svn/drafts/Standarder/IHE/OID/DK-IHE_Metadata-Common_Code_systems-Value_sets.xlsx

Parameter	Optionality	Argument format	Description
-stable	0..1	None	Search for stable documents
-typeCode	0..*	<CODE>,<CODE_SYSTEM> Eg. 12,34	Type code of the document[2]

The time intervals creationTime, serviceStart and serviceStop are defined as two comma separated ISO8601 time stamps: <FROM_TIME>,<TO_TIME>. From_time and To-time are both optional, but not at the same time. Table 15 shows a couple of examples:

Argument	Is valid	Description
2016-01-01,2016-02-01T12:00:00	Yes	Complete interval
2016-01-01,	Yes	All dates after 'from' time
,2016-01-01	Yes	All dates before 'to' time
,	No	-

Table 14 Date and time interval examples

The parameters that have an optionality of '0..*' may be defined several times on the command-line. Eg. '-eventCode 1,2 –eventCode 3,4'. The two codes are added to the FindDocuments Query request.

4 Examples

organisation_afdeling_sor	organisation_afdeling_stednavn	organisation_afdeling_vejnavn	organisation_afdeling_postnummer	organisation_afdeling_by	organisation_ansvarlig_fornavn	organisation_ansvarlig_efternavn	organisation_sor	organisation_navn
368061000016003	Aalborg Lungemedicinsk Afsnit	Mølleparkvej 4 6	9000	Aalborg1	Anders	Andersen	275901000016006	Aalborg Universitetshospital
368061000016003	Aalborg Lungemedicinsk Afsnit	Mølleparkvej 4 6	9000	Aalborg2	Anders	Andersen	275901000016006	Aalborg Universitetshospital
368061000016003	Aalborg Lungemedicinsk Afsnit	Mølleparkvej 4 6	9000	Aalborg3	Anders	Andersen	275901000016006	Aalborg Universitetshospital

Table 15

legal_organisation_afdeling_sor	legal_organisation_afdeling_stednavn	legal_organisation_afdeling_vejnavn	legal_organisation_afdelig_postnummer	legal_organisation_afdeling_by	legal_organisation_ansvarlig_fornavn	legal_organisation_ansvarlig_efternavn	legal_organisation_sor	legal_organisation_navn
368061000016003	Aalborg Lungemedicinsk Afsnit	Mølleparkvej 4 6	9000	Aalborg1	Anders	Andersen	275901000016006	Aalborg Universitetshospital

Table 16

custodian_sor	custodian_stednavn	custodian_afdeling_navn	custodian_vejnavn	custodian_postnummer	custodian_by
368061000016003	Aalborg Universitetshospital	Aalborg Lungemedicinsk Afsnit	Mølleparkvej 46	9000	Aalborg4
368061000016003	Aalborg Universitetshospital	Aalborg Lungemedicinsk Afsnit	Mølleparkvej 46	9000	Aalborg5
368061000016003	Aalborg Universitetshospital	Aalborg Lungemedicinsk Afsnit	Mølleparkvej 46	9000	Aalborg6

Table 17

patient_cpr	patient_fornavn	patient_efternavn	patient_vejnavn	patient_postnummer	Patient_by	Patient_tlf	Patient_mail	referencer
2512484916	nancy	berggren	Forsvindingsvej 1	8210	Aarhus V	1234	asdb@asdb.com	Ja
2512484916	nancy	berggren	Forsvindingsvej 1	8210	Aarhus V	1234	asdb@asdb.com	Nej

Table 18

phmr_tidspunkt	phmr_type	phmr_enhed	phmr_værdi	medicalDeviceCode	medicalDeviceDisplayName	manufacturerModelName	phmr_maalinger_patient
2016-10-14T08:47:55+0100	NPU03804	kg	50#66	MCI00001	Weight	Manufacturer: AD Company / Model: 6121ABT1	2
2016-10-14T08:47:55+0100	DNK05472	mmHg	80#120	MCI00012	Bloodpressure	Manufacturer: A&D Medical / Model: UA-767PlusBT-Ci Bluetooth	2
2016-10-14T08:47:55+0100	DNK05473	mmHg	80#120	MCI00012	Bloodpressure	Manufacturer: A&D Medical / Model: UA-767PlusBT-Ci Bluetooth	2
2016-10-14T08:47:55+0100	NPU03958	g/L	0,1#1,5	MCI00007	Urinstix	Manufacturer: Roche Diagnostics / Model: Urisys 1100	4
2016-10-14T08:47:55+0100	NPU03011	%	0,5	MCI00005	Lung Monitor	Manufacturer: Vitalograph / Model: Lung Monitor Bluetooth	0

Table 19

qrd_dok_titel	qrd_skabelon	qrd_spørgsmål_id	qrd_type	qrd_svar	qrd_titel	qrd_vejledende_tekst	qrd_spørgsmål_text	qrd_antal
Spørgeskema om KOL	1	q1	numeric	1#50	Spørgsmål 1	Spørgsmål 1	Har du taget antibiotika siden sidste Spørgsmål (ja/nej)	1
Spørgeskema om KOL	2	q2	text	jeg har det meget bedre#jeg har det dårligere#jeg har det mega godt#jeg er fuldstændig rask	Spørgsmål 2	Spørgsmål 2	Har du taget antibiotika siden sidste måling (ja/nej)	1
Spørgeskema om KOL	3	q3	multiple	ok#måske#ved ikke#nej#ja	Spørgsmål 3	Spørgsmål 3	Har du taget antibiotika siden sidste måling (ja/nej)	1
Spørgeskema om KOL	4	q4	analog	0#100#1	Spørgsmål 4	Spørgsmål 4	Har du taget antibiotika siden sidste måling (ja/nej)	1
Spørgeskema om KOL	5	q5	discrete	Ok	Spørgsmål 5	Spørgsmål 5	Har du taget antibiotika siden sidste måling (ja/nej)	1

Table 20

apd_id	apd_status	apd_oprettelsestidspunkt	Time	apd_aftale_emne
9a6d1bac-17d3-4195-89a4-1121bc809b4d	ACTIVE	2016-10-14T08:47:55+0100	See Table 22 (optional)	See Table 23 (optional)
null	CANCELLED	2016-11-14T08:47:55+0100	See Table 22 (optional)	Table 23 (optional)

Table 21 - Base data example

apd_aftale_start	apd_aftale_slut	apd_cron	apd_cron_aftale_laengde
2016-10-14T08:47:55+0100	2016-10-15T08:47:55+0100	null	null
null	null	0 0 19 ? 1 FRI 2018	1

Table 22 Time data example

apd_aftale_emne	Lokation organisation	Performer
2016-10-14T08:47:55+0100	See Table 24	See Table 26

Table 23 Reason data example

apd_lokation_organisation_id	Apd_lokation_organisation_navn	Apd_lokation_organisation_tlf	Apd_lokation_organisation_email	Location organization Address
320161000016005	OUH Radiologisk Ambulatorium (Nyborg)	66113333-4	null	See Table 25

Table 24 Location organization example

apd_lokation_organisation_adresse_vej	apd_lokation_organisation_adresse_postnr	apd_lokation_organisation_adresse_by	apd_lokation_organisation_adresse
Vestergade 17	5800	Nyborg	Denmark

Table 25 Location organization address example

apd_udfoerer_id	apd_udfoerer_adresse	apd_lokation_organisation_tlf	apd_lokation_organisation_email	Performer Identity	apd_udfoerer_organisation_navn
320161000016005	See Table 27	66113333-4	null	See Table 28	OUH Radiologisk Ambulatorium (Nyborg)

Table 26 Performer example

apd_udfoerer_adresse_vej	apd_udfoerer_adresse_postnr	apd_udfoerer_adresse_by	apd_udfoerer_adresse_land
Valdemarsgade 53	5700	Svendborg	Denmark

Table 27 Performer address example

apd_udfoerer_pers_identitet_praefix	apd_udfoerer_pers_identitet_fornavne	apd_udfoerer_pers_identitet_efternavn
Læge	Anders	Anderssen

Table 28 - Performer identity example

apd_forfatter_id	apd_forfatter_adresse	apd_forfatter_organisation_tlf	apd_forfatter_organisation_email	Author identity	apd_forfatter_organisation_navn
48681000016007	See Table 30	62214518#62214519	Toldbodvej@læge.com	See Table 31	Lægerne Toldbodvej

Table 29 Requester example

apd_forfatter_adresse_vej	apd_forfatter_adresse_postnr	apd_forfatter_adresse_by	apd_forfatter _adresse_land
Toldbodvej 10	5700	Svendborg	Denmark

Table 30 Requester address example

apd_forfatter_pers_identitet_praefix	apd_forfatter_pers_identitet_fornavne	apd_forfatter_pers_identitet_efternavn
Læge	Anders	Anderssen

Table 31 Requester identity example