

ASTA documentation – IT-enheden, Rigsarkivet

Indhold

ASTA documentation – IT-enheden, Rigsarkivet	1
1. Name and version.....	2
2. Overall solution description – business logic	2
2.1. Create SIP - Hybris	3
2.2. Edit SIP - Hybris.....	7
2.3. Validate SIP - Nemesis	8
2.4. Create AIP - Athena	9
2.5. Create DIP – Styx	13
2.6. Diagrams of business logic	16
3. Architecture description and diagrams	19
3.1. Hybris.....	20
3.2. Nemesis	20
3.3. Athena	28
3.4. Styx	28
4. Location of code, scripts and possibly binary files	29
5. Operation Setup	29
5.1. Creation of packages and releases	29
5.2. General setup	31
6. Development Setup	33
7. Class diagram.....	35
8. Browser and operating system version supported	38
9. Languages	39
10. Profile	41

1. Name and version

ASTA Version 2.0.0

ASTA stands for "Aflevering af Statistikfiler Til Arkiv" (Delivering of statistical files to the archive).

2. Overall solution description – business logic

What is ASTA?

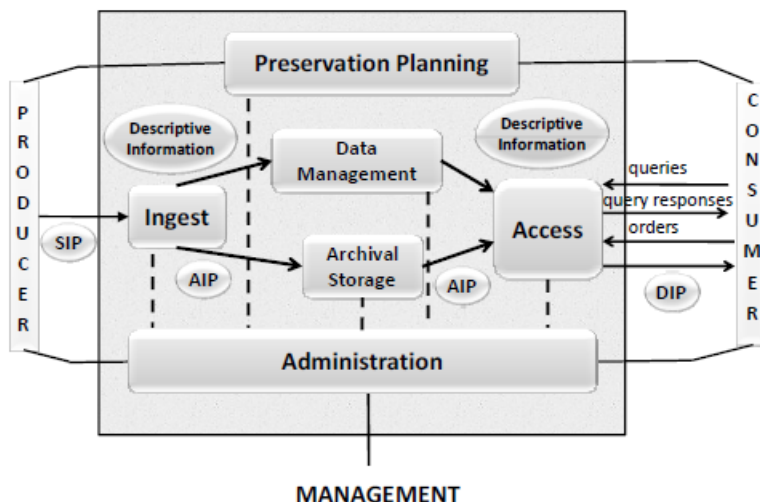
ASTA is a tool for long term preservation of statistical data originated from statistical file formats. The program holds functionality for both creating a SIP (delivery format), validating the SIP, converting the SIP to an AIP (preservation format) and converting the AIP back to a DIP (access format). The terms SIP, AIP and DIP comes from the OAIS model (see figure 1) and stands for:

SIP = Submission Information Package

AIP = Archival Information Package

DIP = Dissemination Information Package

Figure 1. Open Archival Information System model (OAIS)



ASTA comes in two release formats:

- **AstaInstaller.exe** (Windows) or **AstaInstaller.dmg** (Mac) is used by the data creator/submitter and holds only functionality to create, edit and validate a SIP.
- **AstaIntallerExtended.exe** (Windows only) is used by the archive and has both functionality to create, edit and validate a SIP and also to create an AIP and a DIP.

Why is ASTA developed?

The purpose of ASTA is to relieve the data creator and minimize workload when producing a SIP with data from a statistical file format to the archive. As the wait for having your package accepted by the archive can be as much as three months, the validating tool in ASTA make the users almost 100 % sure they have delivered all required material and correct data in the first try. The program also provides the archive an automated conversion of the submitted SIP to an AIP and to create a DIP.

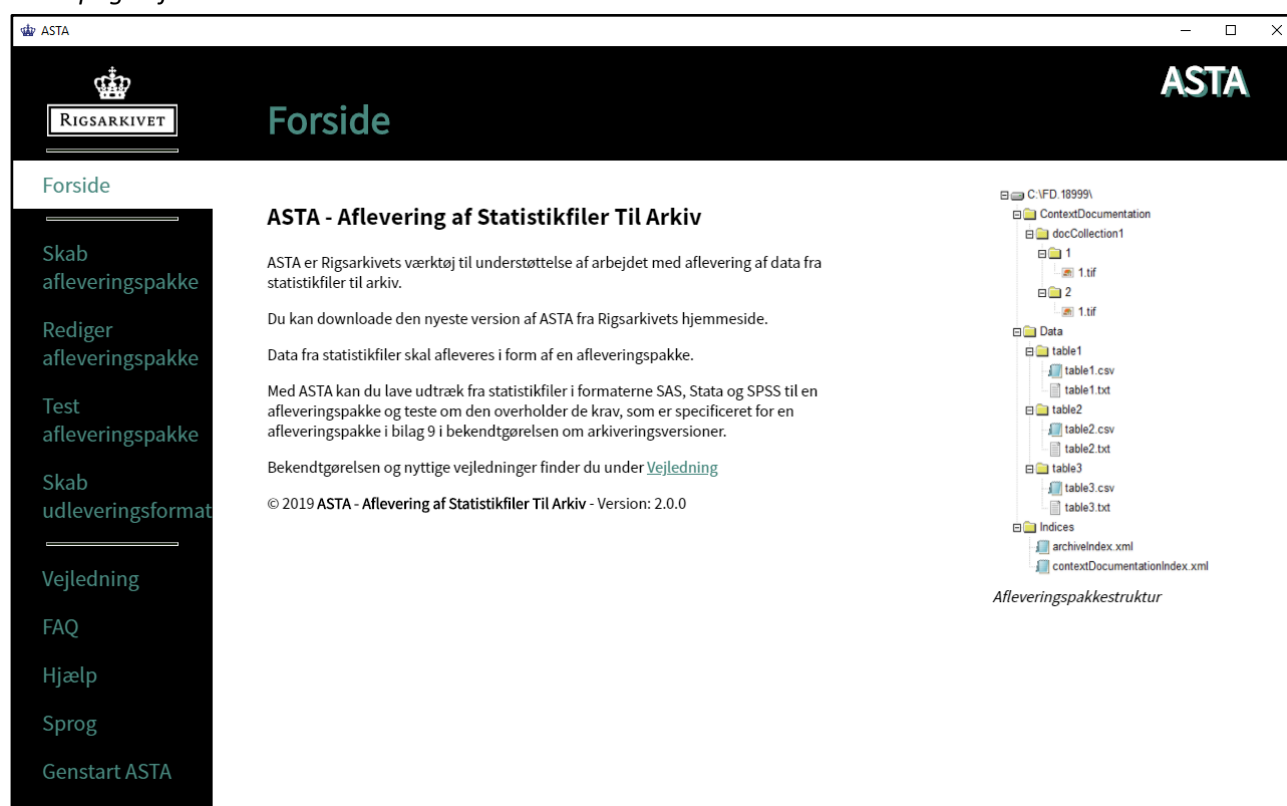
Functionality in ASTA

ASTA has the following functionality and thus support the user to:

- Create SIP = Hybris (**Skab afleveringspakke**)
- Edit SIP = Hybris (**Rediger afleveringspakke**)
- Validate SIP = Nemesis (**Test afleveringspakke**)
- Create AIP = Athena (**Konverter til AV**)
- Create DIP = Styx (**Skab udleveringsformat**)

The names HYBRIS, NEMESIS, ATHENA and STYX are listed above, because the code and this documentation still holds these names for the functionality.

Front page of ASTA



2.1. Create SIP - Hybris

The function *Create SIP* (**Skab afleveringspakke**) in ASTA is a tool for converting statistical data files like SAS, Stata and SPSS files and documentation into a SIP (delivery format) that can be delivered to a Danish archive. The structure and content of a SIP is illustrated in figure 2, 3 and 4.

This happens when a user use the function *Create SIP*:

1. **User** add id for the SIP
2. **User** browse destination for the SIP
3. **ASTA** creates the folders *ContextDocumentation*, *Data* and *Indices* of the SIP (see figure 2)
4. **User** browse a statistical file (.sav, .sas7bdat or .dta)

5. **ASTA** creates the folder *table1* in folder *Data* (see figure 2)
6. **ASTA** places an export script according to the chosen statistical file (.sps, .sas or .do) in the same location as the chosen statistical file
7. **User** runs the export script in a statistical program installed on the PC
8. **Export script** creates 2-4 .txt files with metadata and 1 .csv file (table1.csv) with data from the statistical file and place the output files in the folder *table1* (see figure 2)
9. **User** add metadata information about the statistical file
10. **ASTA** add all metadata from step 6 and 7 into one single metadatafile (table1.txt)
11. **User** converts data from more statistical files (step 3-7 again – table2, table3) (not mandatory)
12. **User** add references across data (not mandatory)
13. **User** copy control files (not mandatory) and ASTA saves the control files in folder *ASTA_kontrolfiler_FD.XXXXXX* next to the SIP
14. **User** browse the index files (archiveIndex.xml and contextDocumentationIndex.xml)
15. **ASTA** places the index files in folder *Indices* (see figure 2)
16. **User** browse the context documentation files
17. **ASTA** places the context documentation in correctly named folders under folder *ContextDocumentation* according to the folder order in the contextDocumentationIndex.xml file (see figure 2)
18. **User** print list of added context documents (not mandatory)
19. **ASTA** saves the list of added context documents in folder *ASTA_kontrolfiler_FD.XXXXXX* next to the SIP

ASTA GUI for step 3

ASTA

RIGSARKIVET

Skab afleveringspakke

FD.12345

Udtræk data fra statistikfilen

Du kan her udtrække data og metadata fra dine statistikfiler til afleveringspakken

Udtrækket består af 3 trin for hver statistikfil:
 TRIN 1: Vælg statistikfil
 TRIN 2: Kør export script i dit statistikprogram
 TRIN 3: Indtast informationer om statistikfilen

Information! Du kan kun lave udtræk fra statistikfiler i formaterne SPSS, SAS og Stata.

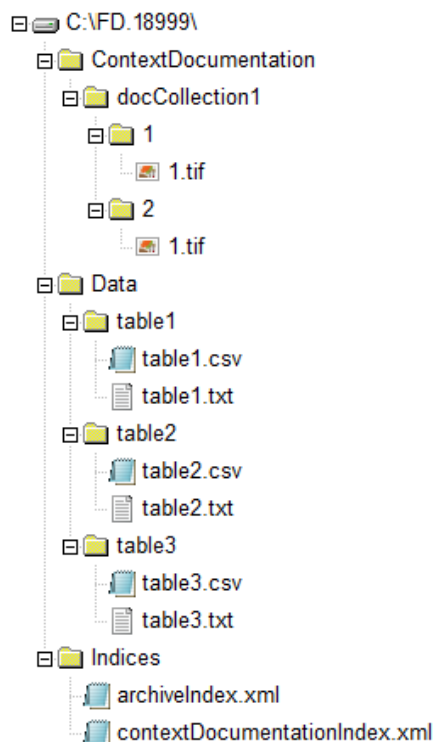
TRIN 1: Vælg statistikfilen som udtræk skal laves fra: ⓘ

C:\Users\b030304\Desktop\12345\Generationsundersøgelsen.sav **Browse**

Næste

INFO: Mappestrukturen er skabt og placeret her: [C:\Users\b030304\Desktop\12345]

Figure 2. Structure and content of a SIP (afleveringspakke)



The folder **ContextDocumentation** contains documents in TIFF-format. The context documentation describes the data and content of the SIP. Description of data collection methods, research methods, questionnaire, publications and so on.

The folder **Data** contains a data file (**table1.csv**) and a metadata file (**table1.txt**) compliant with the rules in schedule 9 in the Executive order on Information Packages. Data and metadata in these files has been extracted from the statistical file format. The metadata is on variable level, e.g. variable labels, variable datatypes, value labels, codes for missing values.

The folder **Indices** contains two indexfiles with metadata about the SIP on a more general level. The file **archiveIndex.xml** holds information about the data creator, dates of data collection, access restrictions and so on. The file **contextDocumentationIndex.xml** holds metadata on the context documentation, like title and the document,

Figure 3. Example of a data file (table1.csv) in the SIP

table1.csv - Notesblok

Filer Rediger Formater Vis Hjælp

```
barn_id;koen;vaegt_idag;slut_dato;hobby;laegebesoeg;maaling;aarsag
11;2;78.2;2017/01/08;4;30-MAR-2002 15:31:22;12:02:57;blødning i urinen
204;1;52.3;2018/05/11;10;03-SEP-2008 12:00:01;4:31:36;smerte når jeg går på toilettet
10095;1;83.1;2013/02/02;9;01-MAR-1999 05:10:45;1:42:20;svaghed
10098;1;56.0;2017/02/02;4;15-NOV-2009 11:15:00;15:46:35;svært at trække vejret
10115;1;74.5;2013/08/28;5;02-JAN-1996 10:44:20;11:15:55;træt hele tiden
10116;1;80.8;2014/12/20;4;31-OCT-2001 23:51:30;18:23:05;"ingen symptomer; men vil gerne tjekkes"
10118;1;102.4;2013/02/18;5;10-JUN-1965 01:26:10;10:01:06;træthed
10119;2;60.2;2015/07/16;2;06-NOV-2008 20:07:30;18:42:26;besvimer
10120;1; ;2018/01/11;1;18-APR-1992 18:00:01;22:34:57;konstant mangel på vitaminer
10121;1;80.0;2017/02/10;3; ;20:06:18;vil gerne tjekkes
10125;1;95.2;2018/07/08;10;03-SEP-2001 02:10:01;6:44:57;smerte
1013;2;64.5;2017/10/09;3;01-FEB-1995 08:20:45;8:55:41;blødning i urinen
10139;1;87.7;2013/08/28;3;30-AUG-2012 18:10:22;8:45:18;"får ""kvalme"" når jeg spiser"
10140;2;50.1;2014/12/20;3;03-NOV-2018 11:22:01;15:48:36;kan ikke trække vejret
10142;2;56.8;2017/02/07;4;01-JAN-1991 07:15:56;19:42:31;træthed
12906;9;90.8;2014/11/08;9;30-MAY-2001 15:31:22;15:57:57;oppustet mave
16434;1;96.7;2015/02/28;9;03-DEC-2010 10:00:01;6:26:36;konstant blødning - menstruation
4520;1;59.1;2017/11/13;5;30-OCT-2009 17:31:20;17:57:55;smerte i maven
4574;1; ;2013/09/23;4;03-APR-1998 17:10:01;6:01:35;fejler ikke noget
9382;2;49.1;2018/05/11;3;01-JUL-1996 08:10:45;12:37:20;lavt immunforsvar
```

Figure 4. Example of a metadata file (table1.txt) in the SIP

```
table1.txt - Notesblok
Filer Rediger Formater Vis Hjælp
SYSTEMNAVN
SPSS

DATAFILNAVN
Børn

DATAFILBESKRIVELSE
Oplysninger om børnene i generationsstudiet

NØGLEVARIABLE
barn_id

REFERENCE
Forældre 'barnid' 'barn_id'
Bedsteforældre 'barnebarnid' 'barn_id'

VARIABLE
barn_id a765
koen f3 koen.
vaegt_idag f3.1
slut_dato sdate10
hobby f3 hobby.
laegebesoeg datetime20
maaling time8
aarsag a100

VARIABLEBESKRIVELSE
barn_id 'Barn ID. Unikt ikke-betydningsbærende identifikationsnummer.'
koen 'Er du mand eller kvinde?'
vaegt_idag 'Hvad er din vægt idag? (angivet i kilogram/kg)'
slut_dato 'Gymnasial uddannelse slut dato'
hobby 'Hvad er din hobby?'
laegebesoeg 'Lægebesøg - dato og tidspunkt'
maaling 'Tidspunkt for måling af blodtryk'
aarsag 'Årsag til videre undersøgelse'

KODELISTE
hobby
'1' 'Sport'
'2' 'Musik'
'3' 'Håndværk'
'4' 'Kunst'
'5' 'Bøger'
'9' 'uoplyst'
'10' 'irrelevant'
koen
'1' 'Mand'
'2' 'Kvinde'
'9' 'uoplyst'

BRUGERKODE
hobby '9' '10'
koen '9'
```

2.2. Edit SIP - Hybris

ASTA has no “Undo” or “Back” functionality so the creation of a SIP must be made in one linear take. However, index files and context documentation can be replaced or added to the SIP later using the function Edit SIP (**Rediger afleveringspakke**).

This happens when a user use the function *Edit SIP*:

1. **User** browse SIP (eg. FD.12345)
2. **User** replace index files (archiveIndex.xml and contextDocumentationIndex.xml) (not mandatory)
3. **User** replace or add context documentation files

ASTA GUI for step 3

ASTA

Rediger afleveringspakke

FD.12345

✓ Indeksfilerne er placeret

Placer kontekstdokumenter i afleveringspakken

Du skal tilføje nedenstående kontekstdokumenter til afleveringspakken.
Du kan kun placere en fil per dokumentmappe.
Hvis et kontekstdokument ikke tilføjes, oprettes en tom dokumentmappe navngivet med mappennummeret.
Hvis du ikke har kontekstdokumenterne klar, kan du udskrive dokumentlisten og tilføje dem manuelt senere.

Information! Kontekstdokumenter skal konverteres til bevaringsformat før de afleveres til arkiv, fx .tif eller .mp3

Dokumentliste

Mappennummer	Dokumenttitel	Vælg dokument
1	Afleveringsbestemmelse	C:\Users\b030304\Desktop\FD.12345\ContextDocumentation\docCollection1\1\1.tif Browse
2	System formål notat	C:\Users\b030304\Desktop\FD.12345\ContextDocumentation\docCollection1\2\1.tif Browse
3	Indsamlingsmetode	C:\Users\b030304\Desktop\FD.12345\ContextDocumentation\docCollection1\3\1.tif Browse
4	Forskningsresultater	C:\Users\b030304\Desktop\FD.12345\ContextDocumentation\docCollection1\4\1.tif Browse
5	Projektbeskrivelse	Vælg sti med knappen Browse
6	Spørgeskema	Vælg sti med knappen Browse
7	Protokol	Vælg sti med knappen Browse

[Udskriv dokumentliste](#)

[Næste](#)

INFO: Filene archiveIndex.xml, contextDocumentationIndex.xml er kopieret til mappen "Indices" i afleveringspakken

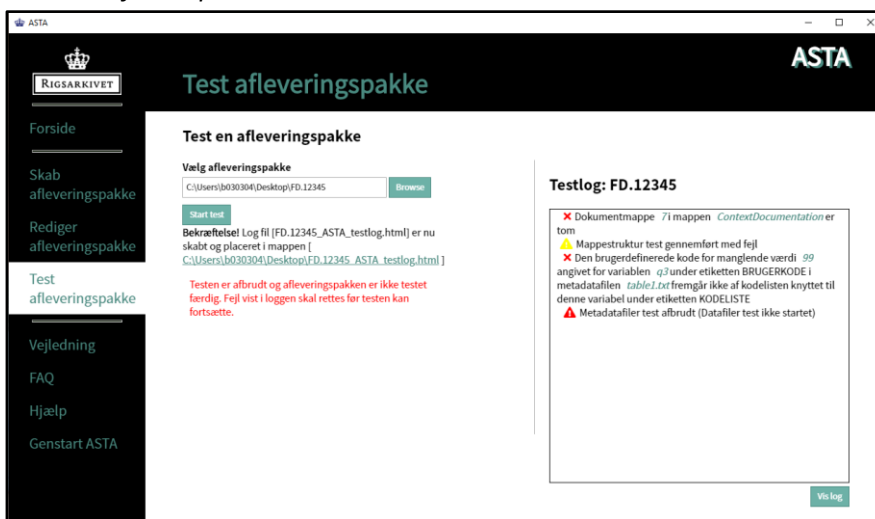
2.3. Validate SIP - Nemesis

When a SIP has been created ASTA can be used to validate whether the structure and content of the SIP is compliant with the rules specified in the Executive Order on Information Packages (See Appendix 1). This Executive Order holds all the technical business logic implemented in ASTA. The function for validating the SIP is called *Validate SIP (Test afleveringspakke)*.

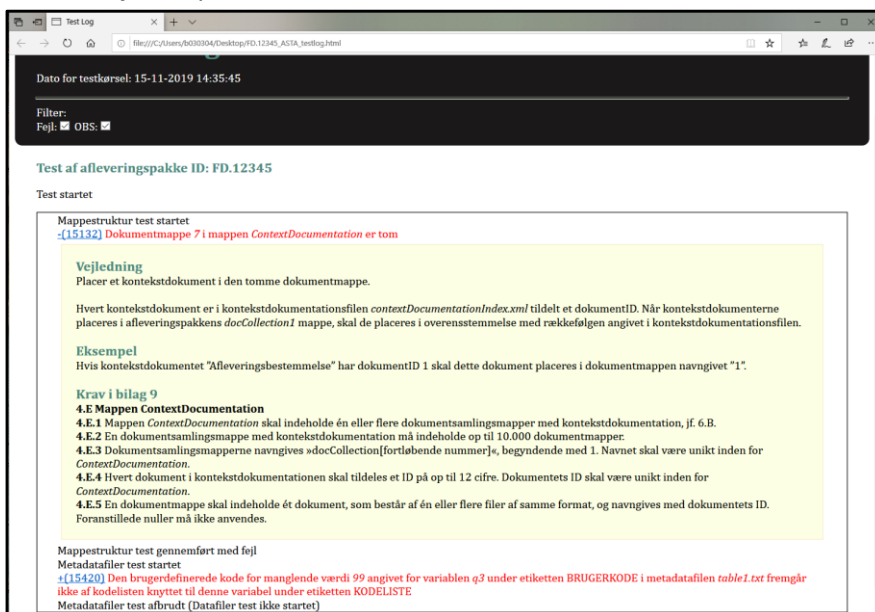
This happens when a user use the function *Validate SIP*:

1. **User** browse SIP (eg. FD.12345)
2. **User** click the button "Start test" to start validation
3. **ASTA** generates af testlog and save a html version in a folder
4. **User** click the errorID in the testlog and read instructions on how to correct the error

ASTA GUI for step 3



ASTA GUI for step 4



2.4. Create AIP - Athena

When the SIP has been delivered to the archive ASTA can be used to convert the SIP to an AIP used for long term preservation. The AIP must be compliant with the rules in the Executive Order on Information Packages schedule 1-8 and will be stored in the preservation storage of the archive. The function for converting the SIP to an AIP is a button called *Create AIP (Konverter til AV)*. This button will only be visible under the function “Test afleveringspakke” when the SIP is compliant with the rules in Executive Order on Information Packages and ready for conversion to an AIP. The structure and content of an AIP is illustrated in figure 5, 6 and 7. The Danish National Archives has developed the tool ADA to validate the AIP. This happens when a user use the function *Create AIP*:

1. **User** test SIP under “Test afleveringspakke” (Validate SIP) and click the button “Konverter til AV”
2. **ASTA** .Net application (Athena) pops up
3. **ASTA** browse the .json file made by Validate SIP under validation of the SIP
4. **ASTA** add the AIP ID according to the chosen AIP (eg. AVID.SA.12345.1)
5. **User** choose the output destination for the converted AIP
6. **User** click the button “Konverter til AV”
7. **ASTA** convert the SIP to AIP format (except the *fileIndex.xml* file) (see figure 5)
8. **ASTA** generates a conversion log file and place it in the folder
ASTA_konverteringslog_AVID.SA.XXXXX.1 in the same folder as the AIP
9. **User** click the button “Næste”
10. **ASTA** gives an overview of data before (SIP) and after (AIP) conversion where user control that the conversion is lossless.
11. **User** click the button “Afslut konvertering”
12. **ASTA** generates the *fileIndex.xml* file and place it in the Indices folder (see figure 5)
13. **ASTA** add information to the conversion log file and generate a conversion report and place both files in the folder ASTA_konverteringslog_AVID.SA.XXXXX.1 in the same folder as the AIP

ASTA GUI for step 10

ASTA

AVID.SA.12345.1 - Kontrol af konvertering

Hovedtabel: spss12345

Række 1 ud af 20

Forskelle i række: 1

Variabelnavn	Beskrivelse	Datatype SIP	Værdi	Datatype AIP	Værdi	Formataendring	Forskelle
bruger_id	BrugerID. Uni...	a765	111	VARCHAR(5)	111	0	0
koen	Er du mand e...	f3	2	INTEGER	2	0	0
alder	Hvor gammel...	f2	52	INTEGER	52	0	0
vaegt_idag	Hvad er din v...	f3.1	2	DECIMAL	0.2	0	1
vaegt_sidste...	Hvad var din ...	f3.2	89.23	DECIMAL	89.23	0	0
hojde	Højde (angiv...	f3.1	168.0	DECIMAL	168.0	0	0
bopael	Bopælsted	a30	Roskilde	VARCHAR(3)	Roskilde	0	0
klasse	Har du gået i ...	f3	9	INTEGER	9	0	0
gymuddannel...	Gymnasial ud...	f3	9	INTEGER	9	0	0
start_dato	Gymnasial ud...	sdate10	2013/01/08	DATE	2013-01-08	19	0
slut_dato	Gymnasial ud...	sdate10	2017/01/08	DATE	2017-01-08	20	0
hobby	Hvad er din h...	f3	4	INTEGER	4	0	0
idraet	Hvilken idræt...	a765		VARCHAR(28)		3	0
laegebesoeg	Lægebesøg	datetime20	30-MAR-200...	TIMESTAMP	2002-03-30T...	19	0
kommune	Navn på kom...	a100	København	VARCHAR(10)	København	0	0
tidspunkt	Tidspunkt for...	time8	11:31:22	TIME	11:31:22	7	0
maaling	Tidspunkt for...	time8	12:02:57	TIME	12:02:57	7	0
anmeldelse	Tidspunkt for...	datetime20	30-DEC-200...	TIMESTAMP	2006-12-30T...	20	0
aarsag	Årsag til vider...	a100	blødning i ur...	VARCHAR(35)	blødning i urin...	0	0

Forskelle i tabel 1

Afslut konvertering

Figure 5. Structure and content of an AIP (arkiveringsversion)

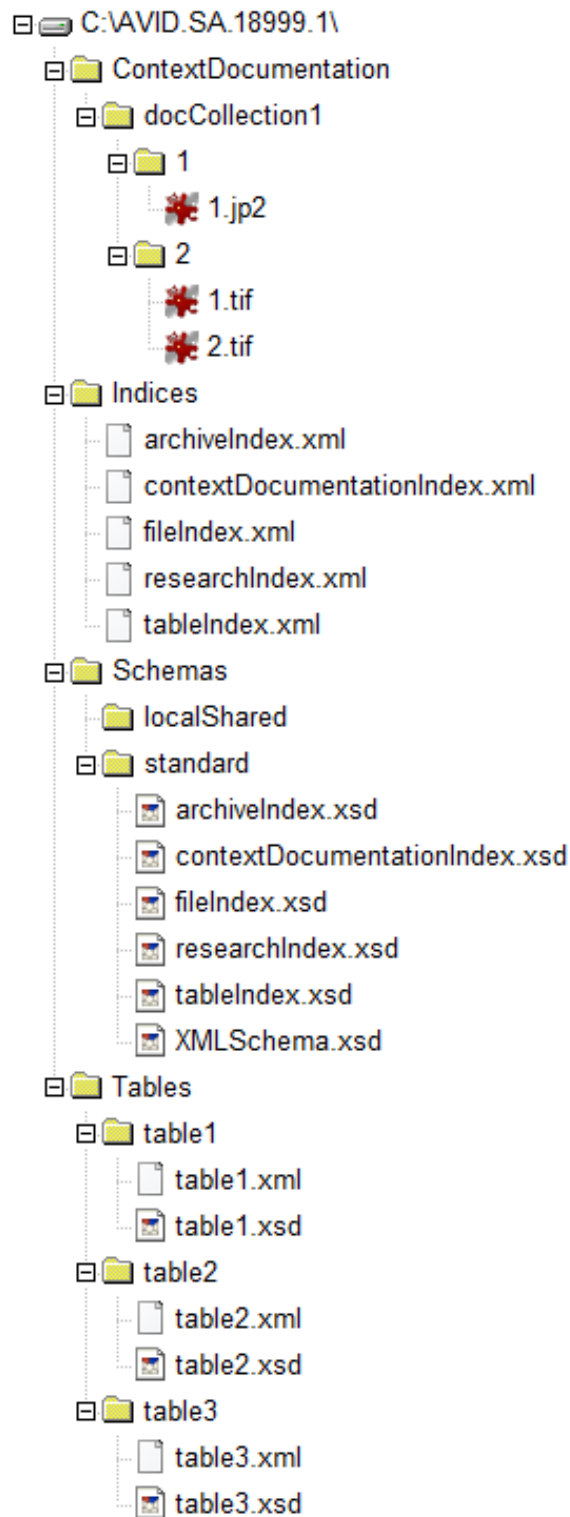


Figure 6. Example of a data file (table1.xml) in an AIP

```
<?xml version="1.0" encoding="UTF-8"?>
- <table xmlns="http://www.sa.dk/xmlns/siard/1.0/schema0/table1.xsd" x
  - <row>
    <c1>11</c1>
    <c2>2</c2>
    <c3>52</c3>
    <c4>78.2</c4>
    <c5>89.23</c5>
    <c6>168.0</c6>
    <c7>Roskilde</c7>
    <c8>9</c8>
    <c9>9</c9>
    <c10>2013-01-08</c10>
    <c11>2017-01-08</c11>
    <c12>4</c12>
    <c13 xsi:nil="true"/>
    <c14>2002-03-30T15:31:22</c14>
    <c15>København</c15>
    <c16>11:31:22</c16>
    <c17>12:02:57</c17>
    <c18>2006-12-30T18:11:22</c18>
    <c19>blødning i urinen</c19>
  </row>
- <row>
    <c1>204</c1>
    <c2>1</c2>
    <c3>69</c3>
    <c4>52.3</c4>
    <c5>50.33</c5>
    <c6>199.0</c6>
    <c7>Slagelse</c7>
    <c8>9</c8>
    <c9>2</c9>
    <c10>2013-02-02</c10>
    <c11>2018-05-11</c11>
    <c12>10</c12>
    <c13 xsi:nil="true"/>
    <c14>2008-09-03T12:00:01</c14>
    <c15>Slagelse</c15>
    <c16>04:00:01</c16>
    <c17>04:31:36</c17>
    <c18>2013-06-05T14:40:01</c18>
    <c19>smerte når jeggår på toilettet</c19>
  </row>
- <row>
    <c1>10095</c1>
    <c2>1</c2>
    <c3>19</c3>
    <c4>83.1</c4>
    <c5>80.13</c5>
    <c6>204.0</c6>
    <c7>København</c7>
    <c8>0</c8>
```

Figure 7. Example of a metadata file (tableIndex.xml) in an AIP

```
<?xml version="1.0" encoding="UTF-8"?>
- <siardDiark xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.sa.dk/xmlns/diark"
  <version>1.0</version>
  <dbName>ikke_relevant</dbName>
  <databaseProduct>Microsoft SQL server</databaseProduct>
- <tables>
  - <table>
    <name>Børn</name>
    <folder>table1</folder>
    <description>Oplysninger om børn i generationsstudiet.</description>
  - <columns>
    - <column>
      <name>barn_id</name>
      <columnID>c1</columnID>
      <type>VARCHAR(5)</type>
      <typeOriginal>a765</typeOriginal>
      <nullable>>false</nullable>
      <description>Barn ID. Unikt ikke-betydningsbærende identifikationsnummer.</description>
    </column>
    - <column>
      <name>koen</name>
      <columnID>c2</columnID>
      <type>INTEGER</type>
      <typeOriginal>f3</typeOriginal>
      <nullable>>false</nullable>
      <description>Er du mand eller kvinde?</description>
    </column>
    - <column>
      <name>alder</name>
      <columnID>c3</columnID>
      <type>INTEGER</type>
      <typeOriginal>f2</typeOriginal>
      <nullable>true</nullable>
      <description>Hvor gammel er du? (angivet i år)</description>
    </column>
    - <column>
      <name>vaegt_idag</name>
      <columnID>c4</columnID>
      <type>DECIMAL</type>
      <typeOriginal>f3.1</typeOriginal>
      <nullable>true</nullable>
      <description>Hvad er din vægt idag? (angivet i kilogram/kg)</description>
    </column>
    - <column>
      <name>vaegt_sidsteaar</name>
      <columnID>c5</columnID>
      <type>DECIMAL</type>
      <typeOriginal>f3.2</typeOriginal>
      <nullable>true</nullable>
      <description>Hvad var din vægt i starten af sidste år? (angivet i kilogram/kg)</description>
    </column>
    - <column>
      <name>hojde</name>
      <columnID>c6</columnID>
      <type>DECIMAL</type>
      <typeOriginal>f3.1</typeOriginal>
```

2.5. Create DIP – Styx

Finally ASTA also holds the function *Create DIP (Skab udleveringsformat)* to convert an AIP to a DIP. A DIP is the access format a user of the archive uses. The DIP is a folder structure with the data and metadata in a format that through the use of a SPSS import script (.sps) can be converted to in a SPSS file format (.sav). The DIP also contains the SPSS import script and the context documentation from the AIP in TIFF-format. The structure and content of a DIP is illustrated in figure 8.

The Danish National Archives have no tool to validate the DIP. The validation of the data in the DIP conversion can be made by comparing the converted SPSS-file in the DIP (Styx output) with the original SPSS statistical file used for the SIP (Hybris input) using a compare syntax in the program SPSS Statistics. Other validation methods must be used for SAS and Stata files.

This happens when a user use the function *Create DIP*:

1. **User** click on “Skab udleveringsformat” in the left menu
2. **ASTA** .Net application (Styx) pops up
3. **User** browse an AIP (eg. AVID.SA.12345.1)
4. **ASTA** add the DIP ID according to the chosen AIP (eg. DIP.12345)
5. **User** choose the output destination for the converted DIP
6. **User** click the button “Konverter til DIP”
7. **ASTA** creates the DIP (se figure 8)
8. **User** runs the SPSS import script (.sps) in his SPSS program installed on his PC
9. **SPSS import script** generates the SPSS file and save the file in the same folder as the script

ASTA GUI for step 7

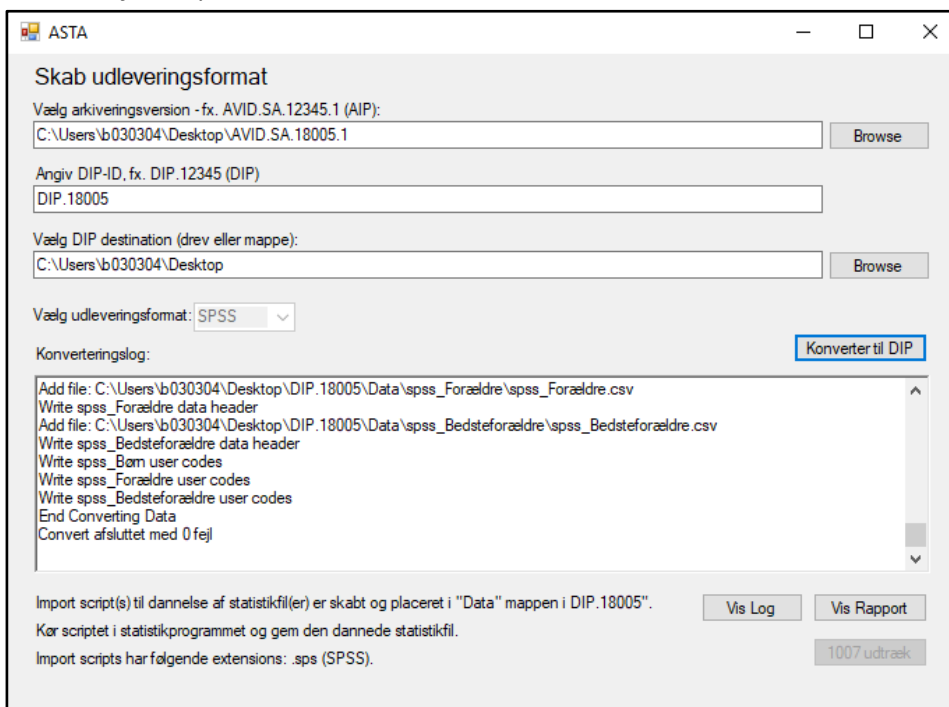
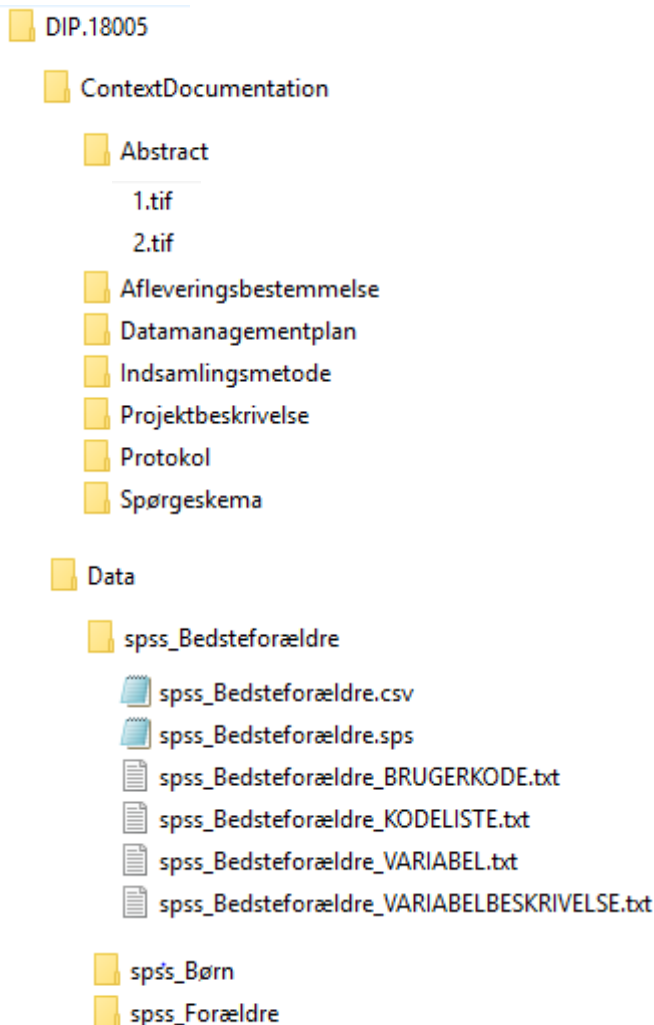


Figure 8. Structure and content of a DIP



The 1007 solution

Styx was originally developed to convert an AIP submitted according to schedule 9 and made by ASTA to a DIP.

However The Danish National Archives also had a strong requirement to convert data and metadata from an AIP not created by ASTA to a SPSS file format. These AIP's are compliant with the rules in the previous Executive Order on Information Packages no. 1007, schedule 1-8. The 1007-format do not contain a *researchIndex.xml* file with information about main tables. Also more datatypes are allowed in the 1007-format than in a SIP according to schedule 9 in the new Executive order.

This *1007 solution* functionality is also implemented in Styx and works this way:

1. **User** click on "Skab udleveringsformat" in the left menu
2. **ASTA** .Net application (Styx) pops up
3. **User** browse an AIP (eg. AVID.SA.12345.1)
4. **ASTA** add the DIP ID according to the chosen AIP (eg. DIP.12345)
5. **User** choose the output destination for the converted DIP

6. **User** click the button “Konverter til DIP”
7. **ASTA** reports *No research Index file found* (se first figure below)
8. **User** click the button “1007 udtræk”
9. **User** selects main tables and code tables for conversion. User mark column with the code and delete all other columns in code lists than the code explanation. user click the button “Konverter til DIP” (se second figure below)
10. **ASTA** creates the DIP (se figure 8)
11. **User** runs the SPSS import script (.sps) in his SPSS program installed on his PC
12. **SPSS import script** generates the SPSS file and save the file in the same folder as the script

ASTA GUI for step 7

ASTA

Skab udleveringsformat

Vælg arkiveringsversion - fx. AVID.SA.12345.1 (AIP):
C:\Users\b030304\Desktop\AVID.SA.18001.1\AVID.SA.18001.1

Angiv DIP-ID, fx. DIP.12345 (DIP):
DIP.18001

Vælg DIP destination (drev eller mappe):
C:\Users\b030304\Desktop

Vælg udleveringsformat: SPSS

Konverteringslog:

```
Start Converting structure AVID.SA.18001 -> DIP.18001
Create path: C:\Users\b030304\Desktop\DIP.18001
No Research Index file found
Ensure Tables: C:\Users\b030304\Desktop\DIP.18001\Data
End Converting structure
Start Converting Metadata AVID.SA.18001 -> DIP.18001
Build spss_AGG metadata
End Converting Metadata
Convert afsluttet med 0 fejl
```

Konverter til DIP

Vis Log Vis Rapport

1007 udtræk

ASTA GUI for step 9

ASTA

Definer 1007 udtræk - vælg tabeller og kodetabeller til udtræk

Marker en hovedtabel for at se hvilke kodelister der er knyttet til denne tabel. Tabeller uden fremmednøgler vises som kodetabeller.
Flyt en kodetabel tilbage til hovedtabel ved brug af pileknappen, hvis den ikke skal udtrækkes som kodetabel til hovedtabellen den er tilknyttet.

Hovedtabel: AGG

TABELLER

Hovedtabeller

Kodetabeller

ART_kode
AMT_kode

Slet

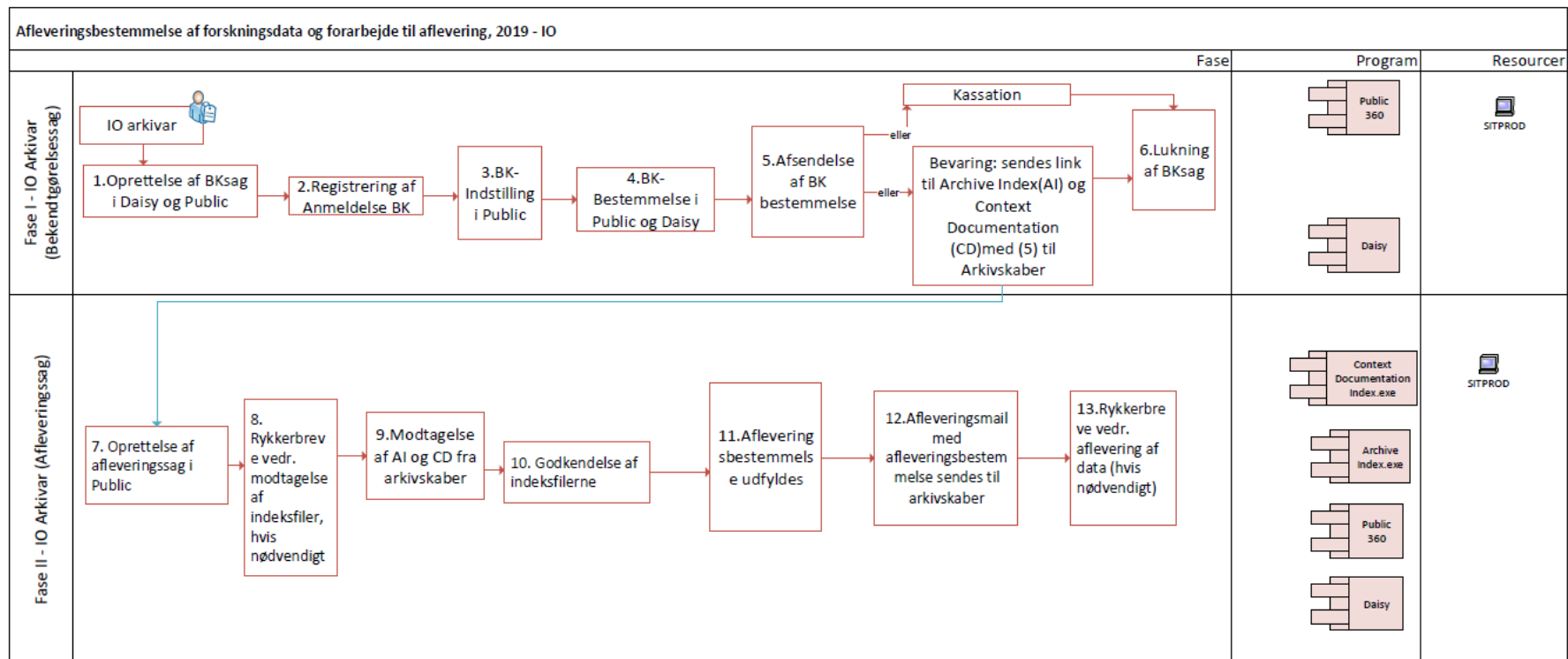
Angiv som kode

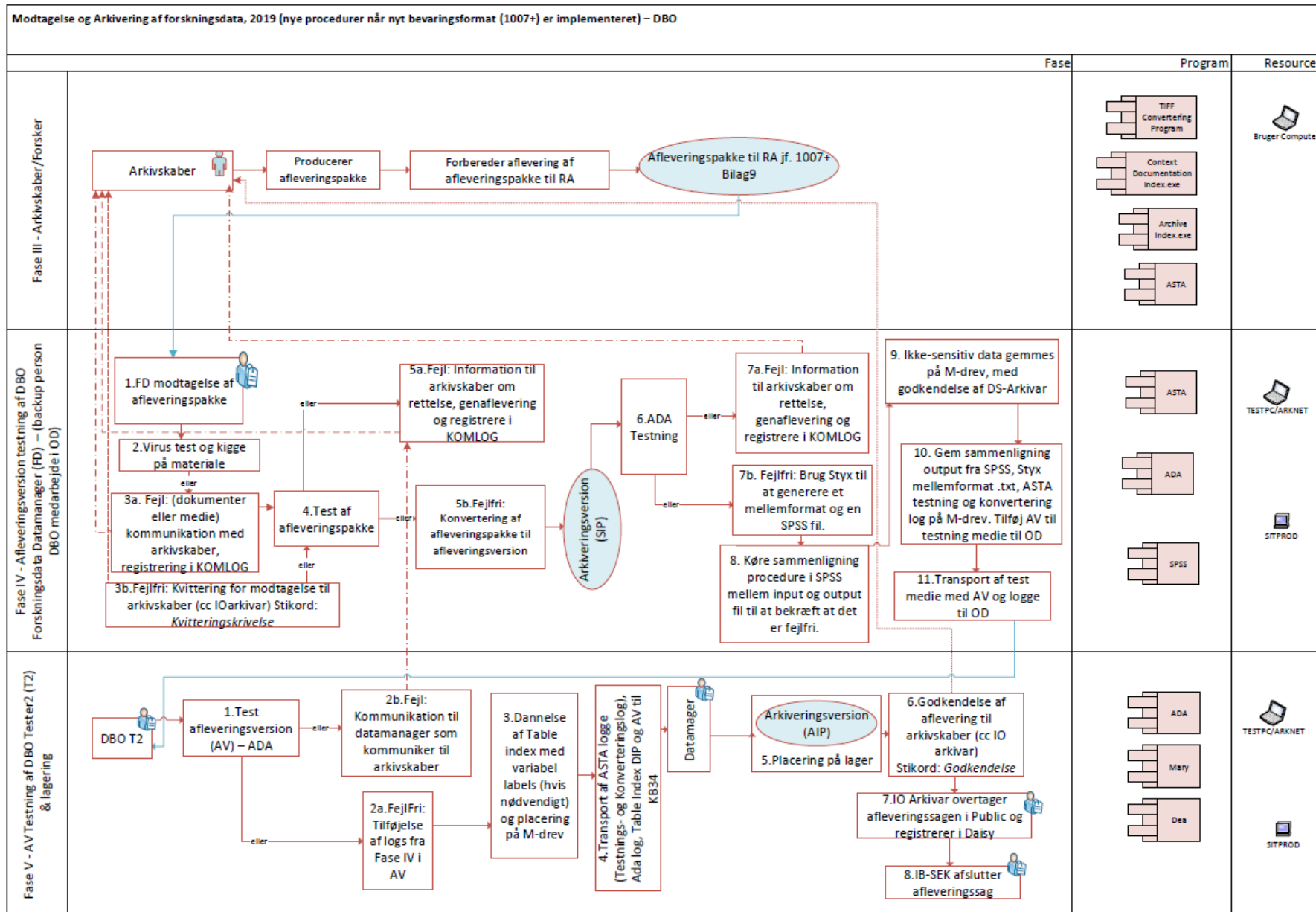
Nogle	Variabelnavn	Beskrivelse	Datatype AIP
	Aar	Jagtsæson, indeholder årstallet for sæ...	VARCHAR(2...
F	ArtID	Fremmednøgle til Art_kode	VARCHAR(2...
P	AmtID	Fremmednøgle til tabel Amt_kode	VARCHAR(2...
	Antal	Angiver antal stykker nedlagt vildt	INTEGER

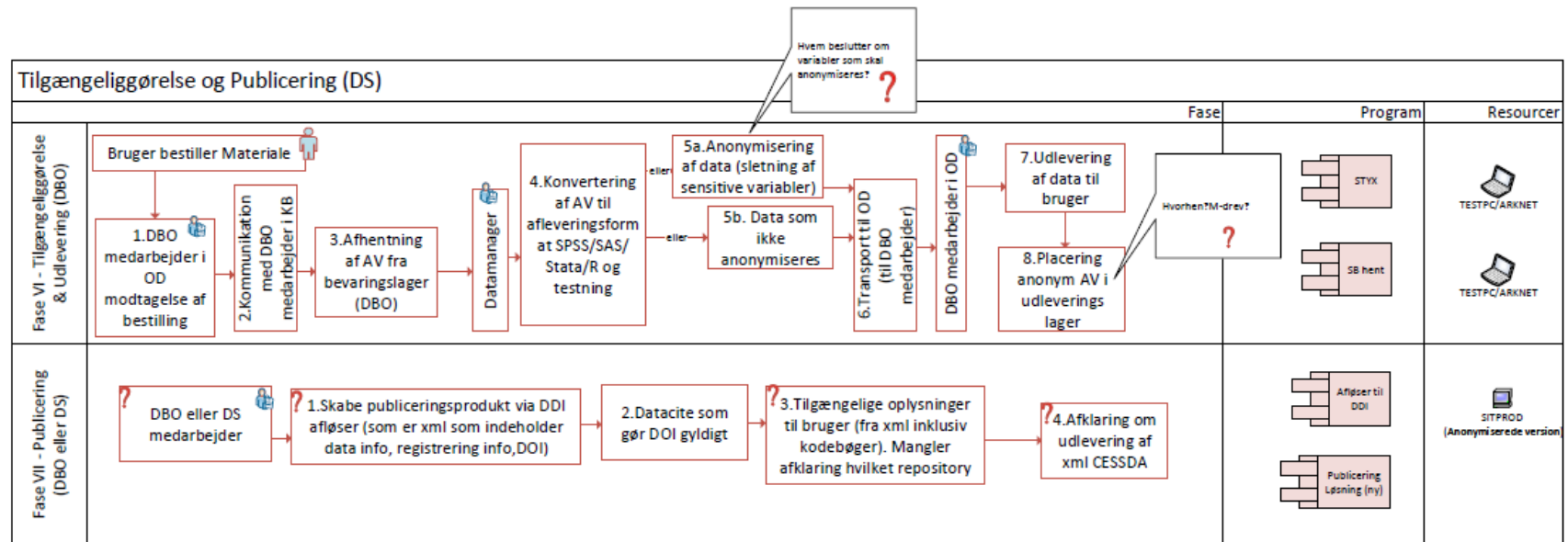
Slet

Konverter til DIP

2.6. Diagrams of business logic







3. Architecture description and diagrams

The full solution consists of two different pieces of software (program 1 and 2 for ease of identification). These two consists of the following four subprograms;

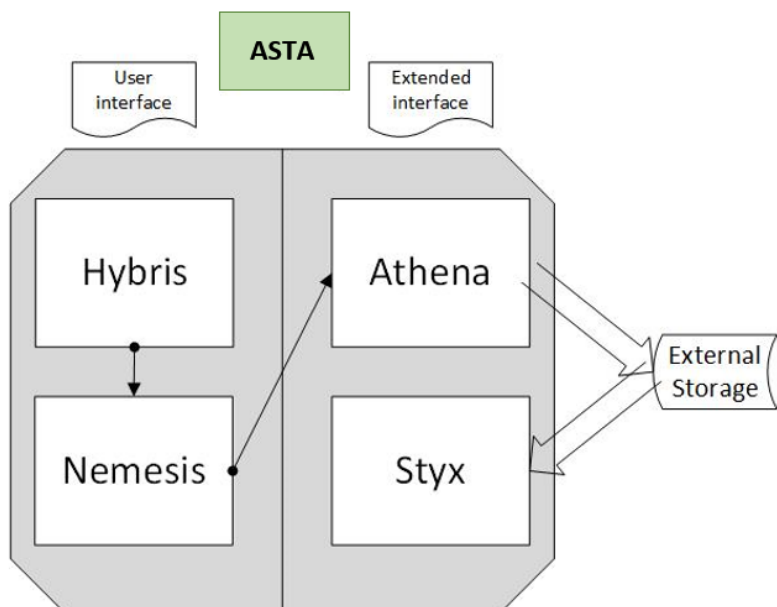
- Hybris: The creation of the SIP/delivery package (program 1)
- Nemesis: Test of the SIP/delivery package. (program 1)
- Athena: The creation of the AIP/preservation format. (program 2)
- Styx: The creation of a DIP/statistical file format (SPSS) from the data in the AIP/preservation format. (program 2)

There will be two different program releases. One for the external users and one for the archive, the latter containing extra features and program parts (only this version will have access to Athena and Styx). The one for the archive will also contain extra options in the program running Nemesis and Hybris. So, the overall setup of the two releases will be as follows;

- **User version:** Program 1, which has the basic functionality the user needs to create and test a SIP.
- **Archive version:** Program 1 + 2, which has extra functionality in program 1.

Program 1 is a cross platform solution made with Electron (<https://electronjs.org/>).

Program 2 is made with .NET technologies (WinForms). The applications have no responsibilities in regard to the external storage and storing of the transformed data.



3.1. Hybris

Export scripts

When a SIP is created Hybris delivers export scripts that enable the user to extract data and metadata from the statistical file format to the format of the data and metadata files in the SIP specified in schedule 9 in the Executive Order on Information Packages. These scripts are placed as resources in the code.

ASTA 2.0.0 holds the following 4 scripts:

- sas_with_catalog_script.sas
- sas_without_catalog_script.sas
- spss_script.sps
- stata_script.do

The export scripts are tested to work well in the following programs:

- SAS v. 9.4
- SPSS v. 24
- Stata v. 14.2

Performance

NOTE: The SPSS export script (spss_script.sps) has low performance on Mac due to the “Golden solution” (Guldøløsningen) in the script, that make sure all value labels are extracted correctly to the metadata file according to the values used in the data file.

Quality of the Hybris solution

The quality of the Hybris solution is good. It has been tested thoroughly both with external and internal users and much of the desired functionality is implemented.

Bugs and new functionality reported to Hybris can be found in YouTrack > Hybris_1007plus Project Management > Unscheduled (<https://bit.myjetbrains.com/youtrack/agiles/64-96/65-463>).

3.2. Nemesis

Flowcharts and Error stop

Nemesis validates a SIP according to the rules in schedule 9 in the Executive Order on Information Packages. Nemesis should test as much as possible before terminating the test to prevent the user to redo the SIP from the start and run Nemesis (validate SIP) more times than absolutely necessary.

The flow of the tests in Nemesis is illustrated in the **flowcharts** in figure 10-14 below. Appendix 2-5 group test suites for Nemesis according to the flowcharts.

An **error-stop** in the Nemesis code, means that the test will be terminated if this error is found. When an error stop is met the test can not continue to the tests in next flowchart, eg. from 2.1 to 3.0. and the button “Konverter til AV” that allows the creation of the AIP in the Athena application will not be shown in Nemesis.

NOTE: Some error stops are inserted in Nemesis to prevent conversion errors in Athena.

A test also terminates when 40 *errors* has been found.

A test of a specific rule also terminates when 100 of the same kind of *hint* has been found.

When a test terminates Nemesis display one of the following **status messages** on the GUI:

- **Afleveringspakken er gennemtestet uden fejl og kan nu afleveres til arkivet**

This means that everything in the SIP has been tested and no errors were found.

- **Afleveringspakken er fuldt gennemtestet. Fejl vist i loggen skal rettes før aflevering til arkiv**

This means that everything in the SIP has been tested, but errors occurred and these should be corrected before submitting the SIP to the archive.

- **Testen er afbrudt, og afleveringspakken er ikke testet færdig. Fejl vist i loggen skal rettes før testen kan fortsætte**

This means that the test has terminated, and not everything in the SIP has been tested. The errors shown must be corrected before the test can continue. Thus more errors can occur in the next test.

Performance

The performance of Nemesis has been tested on different sizes of SIPs and hardware and has been accepted. See the result of the performance test in figure 9.

The performance tests have been executed using the Athena console converter with performance output (under Athena folder). In order to run console start ASTA (nemesis) that's generate json output then run console as followings example:

```
.\\AthenaConsole.exe "C:\\VSO\\Rigsarkivet\\test\\FD.15002.json" "C:\\VSO\\Rigsarkivet\\test"
"AVID.SA.15002.1"
```

Quality of the Nemesis solution

The quality of the Nemesis solution is good. It has been tested thoroughly using a test suite of 247 SIP's with known errors. 150 original statistical datasets (SIPs) has been tested during development. All of the known desired functionality is implemented.

Bugs and new functionality reported to Nemesis can be found in YouTrack > Nemesis_1007plus Project Management > Unscheduled (<https://bit.myjetbrains.com/youtrack/agiles/64-98/65-465>)

Figure 9. Results from performance test of Nemesis and Athena

FileConverterAdmin (8)_a9a601c				AstaExtended 4a9f1d6		
PC Stationær	Medietype (USB, HD, SSD)	Nemesis tid	Athena tid	Medietype (USB, HD, SSD)	Nemesis tid	Athena tid
FD.28415	USB (Stik)	30 sek.	61 sek.	USB (Stik)		
32731 rækker	USB (HD)	30 sek.	56 sek.	USB (HD)		
767 kolonner	USB (SSD)	28 sek.	56 sek.	USB (SSD)		
57 MB	HD (intern)	28 sek.	50 sek.	HD (intern)		
FD.111111	USB (Stik)	51 sek.	38 sek.	USB (Stik)		
508500 rækker	USB (HD)	52 sek.	35 sek.	USB (HD)		
50 kolonner	USB (SSD)	50 sek.	35 sek.	USB (SSD)		
54 MB	HD (intern)	50 sek.	35 sek.	HD (intern)		
FD.333333	USB (Stik)		Plads ???	USB (Stik)		
11.300.000 rækker	USB (HD)			USB (HD)		
50 kolonner	USB (SSD)	18 min 12 sek.	14 min. 30 sek.	USB (SSD)		
1200 MB	HD (intern)	19 min	35 min.	HD (intern)		
PC bærbar	Medietype (USB, HD, SSD)	Nemesis tid	Athena tid	Medietype (USB, HD, SSD)	Nemesis tid	Athena tid
FD.28415	USB (Stik)	43 sek.	1 min 20 sek.	USB (Stik)	43 sek.	1 min 20 sek.
32731 rækker	USB (HD)	44 sek.	1 min 15 sek.	USB (HD)	44 sek.	1 min 20 sek.
767 kolonner	USB (SSD)			USB (SSD)		
57 MB	HD (intern)	46 sek.	1 min 10 sek.	HD (intern)	47 sek.	1 min 20 sek.
FD.111111	USB (Stik)	1 min 40 sek.	40 sek.	USB (Stik)		
508500 rækker	USB (HD)			USB (HD)	1 min 50 sek.	44 sek.
50 kolonner	USB (SSD)			USB (SSD)		
54 MB	HD (intern)	1 min 45 sek.	37 sek.	HD (intern)	1 min 50 sek.	40 sek.
FD.333333	USB (Stik)			USB (Stik)		
11.300.000 rækker	USB (HD)			USB (HD)	40 min 15 sek.	25 min 20 sek.
50 kolonner	USB (SSD)			USB (SSD)		
1200 MB	HD (intern)	42 min 00 sek.	20 min 00 sek.	HD (intern)	41 min 50 sek.	19 min 40 sek.

Figure 10. Nemesis flowchart 0.0: Overview

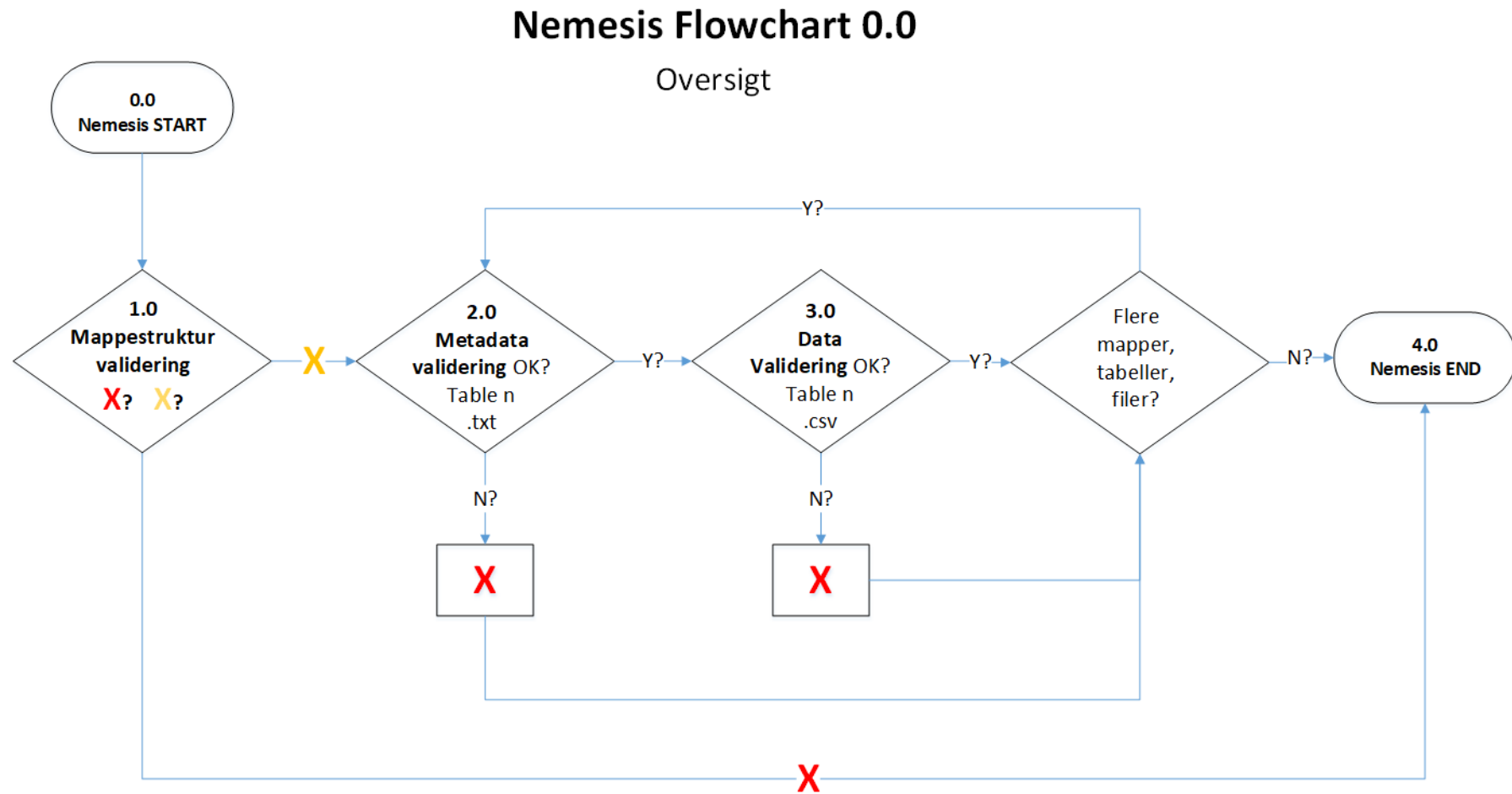
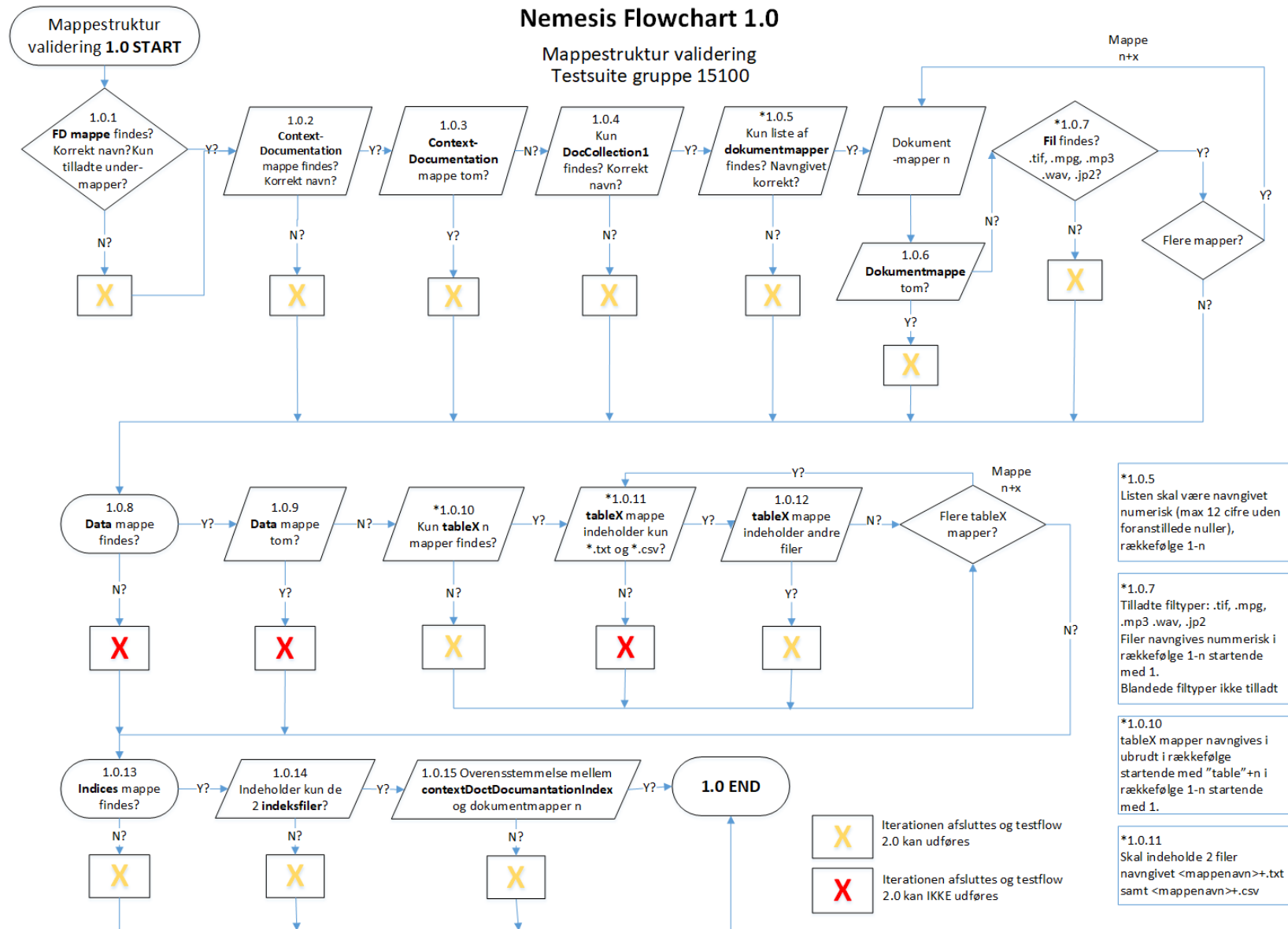


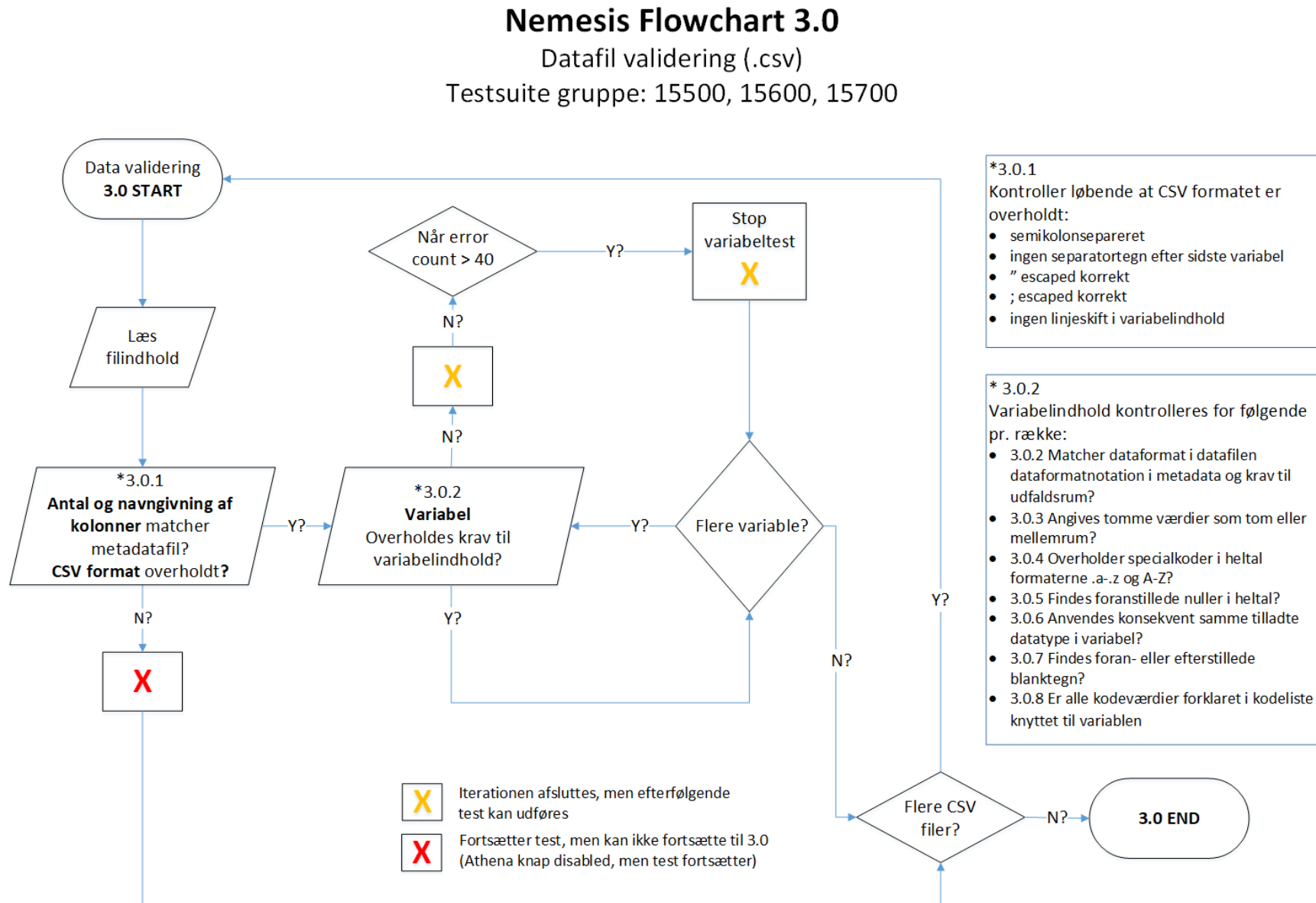
Figure 11. Nemesis flowchart 1.0: Validating folder structure of SIP (see Appendix 2 for test suites)



Metadatafil EBNF validering
Testsuite gruppe 15300



Figure 14. Nemesis flowchart 3.0: EBNF validation of the data file (see Appendix 5 for test suites)



3.3. Athena

Athena convert a SIP (delivery format) to the AIP format (preservation format). In the Danish National Archives the AIP format is called an *arkiveringsversion* and is compliant with the rules in schedule 1-8 in the Executive Order on Information Packages.

Nemesis creates a **.json file** with information collected during the validation of the SIP that is used by Athena during conversion of the SIP to an AIP.

Performance

The performance of Athena has been tested on different sizes of SIPs and hardware and has been accepted. See the result of the performance test in figure 9. Read more about the test in section 3.2.

Quality of the Athena solution

The quality of the Athena solution is good. An AIP created by Athena is accepted with no errors in the validation tool ADA.

However due to the architecture of the code an use of the .json file from a Nemesis validation Athena lacks the ability to report on how many tables and rows in tables the SIP contained before conversion to an AIP. Improvements can be thus be made to the conversion report that Athena creates.

Bugs and new functionality reported to Athena can be found in YouTrack > Athena_1007plus Project Management > Unscheduled (<https://bit.myjetbrains.com/youtrack/agiles/64-100/65-467>).

3.4. Styx

SPSS import script

Styx convert an AIP (preservation format) to the DIP format (access format). In the Danish National Archives the DIP format is statistical file formats. ASTA 2.0.0 only support conversion to the SPSS file format.

Styx delivers an SPSS import script (.sps) for creating the SPSS file (.sav) from the DIP-format (data in a .csv file and metadata in several .txt files).

Styx also convert *data formats* from the data xml-files in the AIP and *data types* from the metadata file *tableIndex.xml* to valid SPSS data formats and datatypes that are accepted when imported by the SPSS import script.

Drafts for import scripts to SAS and Stata files have been made, but due to challenges in importing codes for missing values no final versions are ready for use yet.

Quality of the Styx solution

The quality of the Styx solution is poor. It is usable but had low priority during development.

The testing done on Styx shows that often the SPSS import script do not read all data from the data file in the DIP into the SPSS format. Consequently, when using Styx is it necessary to make a visual control and compare data and metadata in the AIP with the content of the SPSS file to make sure the DIP conversion is lossless. The validation of the data in the DIP conversion can also be made automatically by comparing the converted SPSS-file in the DIP with the original statistical file used for the SIP using a compare syntax in the program SPSS Statistics.

Bugs and new functionality reported to Styx can be found in YouTrack > Styx_1007plus Project Management > Unscheduled (<https://bit.myjetbrains.com/youtrack/agiles/64-102/65-469>).

4. Location of code, scripts and possibly binary files

The solution is hosted in GitHub (<https://github.com/the-danish-national-archives/ASTA>). From here the solution can be cloned to a local machine.

The test suite for Nemesis and other test material is also placed in GitHub.

Backlog for ASTA with reported bugs and further requirements are placed in YouTrack under *Unscheduled* in the following Agile Boards (<https://bit.myjetbrains.com/youtrack>):

- Hybris_1007plus Project Management
- Nemesis_1007plus Project Management
- Athena_1007plus Project Management
- Styx_1007plus Project Management

5. Operation Setup

5.1. Creation of packages and releases

Code is placed on GitHub

The solution is hosted in GitHub as described above and can be accessed via Git commands. There are no requirements for tools to do this, only a terminal with Git installed. GitHub does have a desktop application that can be used instead of a terminal.

In the package.json file, all script commands used by the program are detailed, like 'npm start', which is used to start the program 1 locally.

To run one of the other custom scripts, use 'npm run <script>' – e.g. npm run package-win. That command will make a user version of the solution. In the package.json file it is detailed how to make different versions of the program, this is listed in the 'scripts' part, see below:

```
"scripts": {
  "start": "electron .",
  "package-mac": "electron-packager . asta-app --overwrite --extra-resource=assets/scripts/version.js",
  "package-mac-extended": "electron-packager . asta-extended-app --overwrite --extra-resource=assets/s",
  "package-win": "electron-packager . asta-app --overwrite --extra-resource=assets/scripts/version.js",
  "package-win-extended": "electron-packager . asta-extended-app --overwrite --extra-resource=assets/s",
  "package-linux": "electron-packager . asta-app --overwrite --extra-resource=assets/scripts/version.j",
  "package-linux-extended": "electron-packager . asta-extended-app --overwrite --extra-resource=assets",
  "create-installer-win": "node installers/windows/createinstaller.js",
  "create-installer-win-extended": "node installers/windows/createadmininstaller.js",
  "create-installer-mac": "electron-installer-dmg ./release-builds/asta-app-darwin-x64/asta-app.app AS",
  "create-installer-mac-extended": "electron-installer-dmg ./release-builds/asta-extended-app-darwin-x",
},
```

Creating packages

When creating a release package, you need to use the 'electron-packager' component. This will make a

package of the solution, which then can be made into an .exe / .dmg file. There is made several scripts to handle this, these all start with “package- “.

“electron-packager” is an npm package that can be found at: <https://www.npmjs.com/package/electron-packager>. Here you can find relevant information about the use of the module. Please note that the ‘—extra-resources’ flag have not been described here. This flag allows for the inclusion of extra files in the build (such as the scripts, languages and PDF files). The other used flags are handling out folders, icon for the application and so on.

There is two different packaging script for the various operating systems. Those that are called “-extended” is for the archive only. In this package the administrative tools are included in the interface (they are hidden, but inaccessible in the user version).

Versioning the releases

Firstly the ‘version’ field in package.json needs to be updated. This field will be reflected in the interface and program list (on Windows). The release number consists of three digits e.g. 1.0.4, these digits are defined as;

1 – Major change (e.g. new language pack)

0 – Minor change (new functionality in existing features in ASTA, addition of R scripts)

4 – Bug fixes

After this a package can be made using the package- scripts.

Creating an executable file

Once the package has been made, an executable file then needs to be made. This is done by running the scripts starting with “create-“. Here are also scripts for making admin and user versions of the files.

NOTE: the .dmg files must be made on a Mac machine. These cannot be created on Windows. The script will throw an error and inform about this if attempted.

On the GitHub page, you can find more information about how to use GitHub.

NOTE: When creating the .exe file, be aware that an error can occur if you have an old file in the same output folder as you’re packaging to. Make sure to delete old executable files before creating new ones. The error is that the packaging module will include the old .exe file in the build and the program will then start twice when you run it.

Creating release on GitHub

After the executable files have been created, a release needs to be created on GitHub. Releasing on GitHub follows the GitHub guide on releases, which can be found at <https://help.github.com/en/articles/creating-releases>

This does in short include;

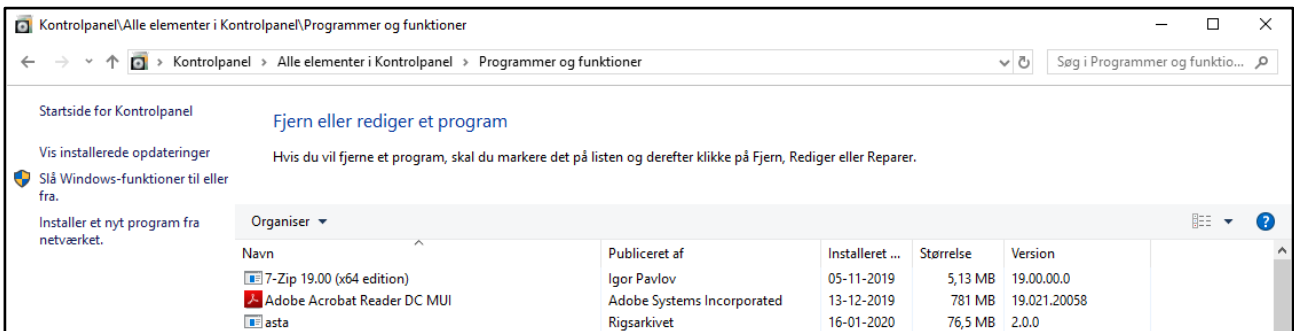
- Draft a new release
- Create version number and tag for the release

- Include the binary files for the release (non-code related files)
- Create the (pre-)release.

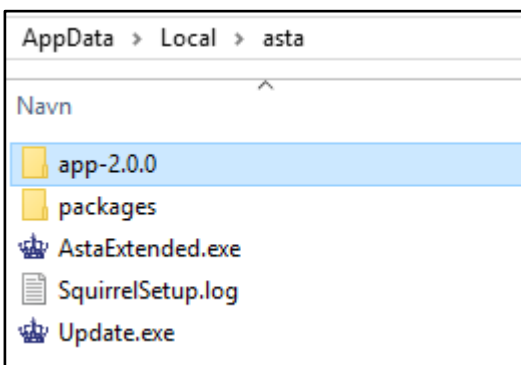
5.2. General setup

Installation of ASTA

When the executable files of ASTA (.exe and .dmg) is run ASTA is installed at the users PC. This might require administrative rights.



On Windows ASTA is placed here: C:\Users\%USERPROFILE%\AppData\Local\asta\app-2.0.0



To create a SIP using Hybris in ASTA the user also need to have a statistical program installed as well, either SAS, Stata or SPSS.

System log files

ASTA creates system log files for the purpose of detecting where something when wrong if ASTA crashes or do not work as expected.

By default, ASTA writes system logs to the following locations:

- **on macOS:** ~/Library/Logs/asta/ASTA_systemlog.log
- **on Windows:** %USERPROFILE%\AppData\Roaming\asta\ ASTA_systemlog.log

[%USERPROFILE%](#): UserProfile environment variable represents the path to the user's profile folder (C:\Users\{username})

The asta.log file format is build up of the following information: [timestamp] [log type] text [code path]

- log type: info, warning or error
- code path: Rigsarkiv.[Model].[Class].[Function]

Example as followings:

[2019-07-03 10:24:04.604] [info] selected path: C:\VSO\Rigsarkivet\SPSS\spss23765_short.sav
Rigsarkiv.Hybris.DataExtraction.AddEvents

[2019-07-03 10:24:24.909] [error] ENOENT: no such file or directory, scandir
'C:\VSO\Rigsarkivet\test\FD.12345\Data' Rigsarkiv.Hybris.DataExtraction.EnsureData

Athena C# .Net- batch and log files

Athena C# .Net has 2 packaged applications within the administrator version of electron:

- AthenaConsole.exe: used to batch converting by starting & passing parameters explicitly as following:
AthenaConsole.exe "SIP json-file path" "AIP output path" "AIP folder name"
- AthenaForm.exe: Invoked from electron as child process with required parameters. Implements by using .Net 4.5 Windows Forms

Both above applications depending on Athena.dll component see [Klassediagram](#). Both above applications System logs settings using log4net configuration in (AthenaConsole.exe.config & AthenaForm.exe.config) files:

```
<log4net>
  <appender name="RollingFile" type="log4net.Appender.RollingFileAppender">
    <file value="${USERPROFILE}\AppData\Roaming\asta\athena_systemlog.log" />
    <appendToFile value="true" />
    <maximumFileSize value="100KB" />
    <maxSizeRollBackups value="2" />
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%date %level %logger - %message%newline" />
    </layout>
  </appender>
</root>
  <level value="INFO" />
  <appender-ref ref="RollingFile" />
</root>
</log4net>
```

Log path on Windows: \${USERPROFILE}\AppData\Roaming\asta\athena_systemlog.log

Styx C# .Net - batch and log files

Styx C# .Net has 2 packaged applications within administrator version of electron:

- StyxConsole.exe: used to batch converting by starting & passing parameters explicitly as following:
StyxConsole.exe "AIP path" "DIP output path" "DIP folder name" "script type" (just supporting 1007+)
- StyxForm.exe: Invoked from electron as child process. Implements by using .Net 4.5 Windows Forms

Both above applications depending on Styx.dll component see [Klassediagram](#). Both above applications System logs settings using log4net configuration in (StyxConsole.exe.config & StyxForm.exe.config) files:

```
<log4net>
  <appender name="RollingFile" type="log4net.Appender.RollingFileAppender">
    <file value="${USERPROFILE}\AppData\Roaming\asta\styx_systemlog.log" />
    <appendToFile value="true" />
    <maximumFileSize value="100KB" />
    <maxSizeRollBackups value="2" />
    <layout type="log4net.Layout.PatternLayout">
      <conversionPattern value="%date %level %logger - %message%newline" />
    </layout>
  </appender>
</root>
  <level value="INFO" />
  <appender-ref ref="RollingFile" />
</root>
</log4net>
```

Log path on Windows: \${USERPROFILE}\AppData\Roaming\asta\styx_systemlog.log

6. Development Setup

The solution for program 1 (Hybris and Nemesis) is made in [Electron](#), so any IDE that supports this, is optimal. As the files never compiles, a simple text editor can do as well. Program 2 (Athena and Styx) is made in .NET and requires Visual Studio to compile.

Once the solution is cloned locally, you need to ensure that the required npm modules for program 1 are installed (these are listed in package.json), therefore you need to run 'npm install' once the initial clone is done. This will install the packages on the machine.

When this is complete you can run 'npm start' which runs program 1 locally.

For program 2 you just run the application from Visual Studio.

Modules used for program 1 (Hybris and Nemesis – Electron)

There have been used various modules (npm) in the solution for program 1, these are listed here;

- Electron (<https://www.npmjs.com/package/electron>). Core component for the solution.
- Electron-packager (<https://www.npmjs.com/package/electron-packager>). Handles the packaging for the various operating systems in the program.
- Electron-installer-dmg (<https://www.npmjs.com/package/electron-installer-dmg>). Creates a mac dmg file (this needs to be run on a Mac machine).
- Fontawesome/fontawesome-free (<https://www.npmjs.com/package/@fortawesome/fontawesome-free>). Used to make icons in the log view in the Nemesis part of the program.
- Chardet (<https://www.npmjs.com/package/chardet>). Checks the format of the data files from the user (if they are UTF-8 formatted).
- Junk (<https://www.npmjs.com/package/junk>). Filters files in searches.

- Nodejs-base64 (<https://www.npmjs.com/package/nodejs-base64>). Does base 64 encoding / decoding.
- XmlDom (<https://www.npmjs.com/package/xmlDom>). Xml file handling.
- electron-log (<https://www.npmjs.com/package/electron-log>). Creates a log file.
- fast-csv (<https://www.npmjs.com/package/fast-csv>). Parsing CSV-data files.
- fs-extra (<https://www.npmjs.com/package/fs-extra>). Adds file system methods that aren't included in the native fs module and adds promise support to the fs methods.

All modules are listed with a minimum version in the code, but once the code is cloned and installed, the latest modules will be downloaded as well.

The list of dependencies can be found at:

<https://github.com/the-danish-national-archives/ASTA/network/dependencies>

Modules used for program 2 (Athena and Styx – C# .Net)

Athena C# .Net has followings modules:

log4net (<http://logging.apache.org/log4net/>) .NET logging

Useful API references:

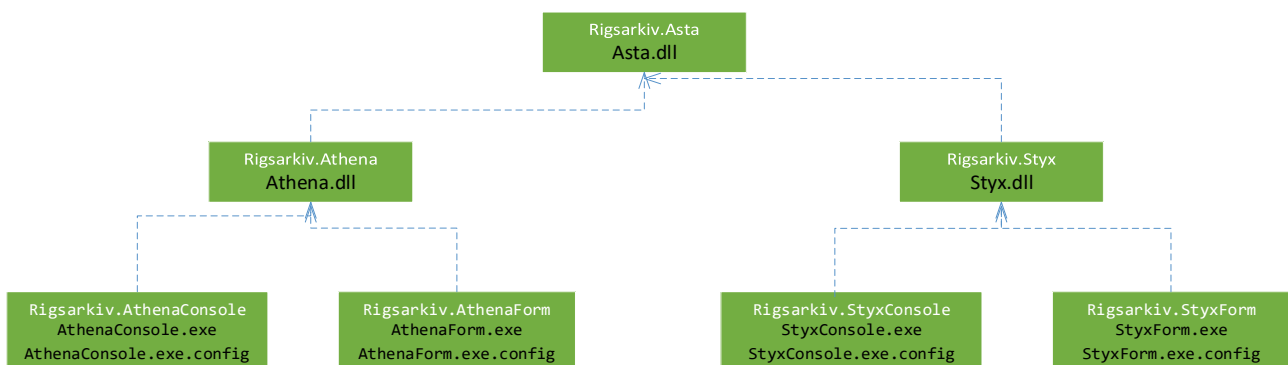
- [Node.js](#): is a JavaScript runtime across platforms. It has a set of built-in modules which you can use without any further installation. (https://www.w3schools.com/nodejs/ref_modules.asp).
 - [File system](#): is one of most used modules in ASTA
 - [Child Process](#): Invoke .Net WinForms applications
 - [OS](#): Platform Information (Windows, Mac or Linux)
- Node.js NPM: is a package manager for Node.js packages, or modules if you like.
- [Electron](#): is an open source library developed by GitHub for building cross-platform desktop applications with HTML, CSS, and JavaScript. Electron accomplishes this by combining Chromium and Node.js into a single runtime and apps can be packaged for Mac, Windows, and Linux.
 - Main and Renderer Processes (<https://electronjs.org/docs/tutorial/application-architecture#differences-between-main-process-and-renderer-process>).
 - Demo API (<https://github.com/electron/electron-api-demos/blob/master/docs.md>).
- [Electron Packager](#): is a command line tool and Node.js library that bundles Electron-based application source code with a renamed Electron executable and supporting files into folders ready for distribution.
 - --extra-resource (<https://gist.github.com/fodra/d6c4572a0c6f0c405e6c2f35a09efb78>)
 - electron-installer-dmg (<https://www.npmjs.com/package/electron-installer-dmg>)
 - electron-winstaller (<https://github.com/electron/windows-installer>)
- [Visual Studio Code](#): is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux.
- [C#](#): (C-Sharp) is a programming language developed by Microsoft that runs on the .NET Framework.
- [.NET Windows Forms](#): Smart clients are graphically rich applications that are easy to deploy and update, can work when they are connected to or disconnected from the Internet, and can access

resources on the local computer in a more secure manner than traditional Windows-based applications.

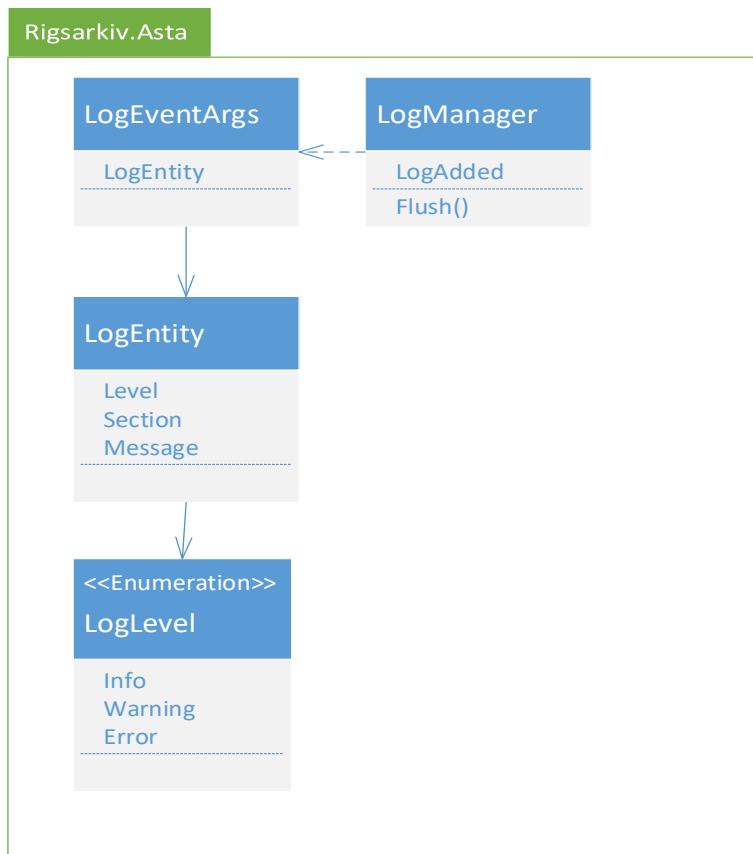
7. Class diagram

For all .js files there have been made code description which describes a files functionality and usage. As this is not OO-programming a class diagram is redundant.

The following illustrates program 2's components diagram.



Asta.dll is responsible for custom logs. By subscribing to LogManager event handler “LogAdded” possible different output displays for each LogEntity object can be implemented.



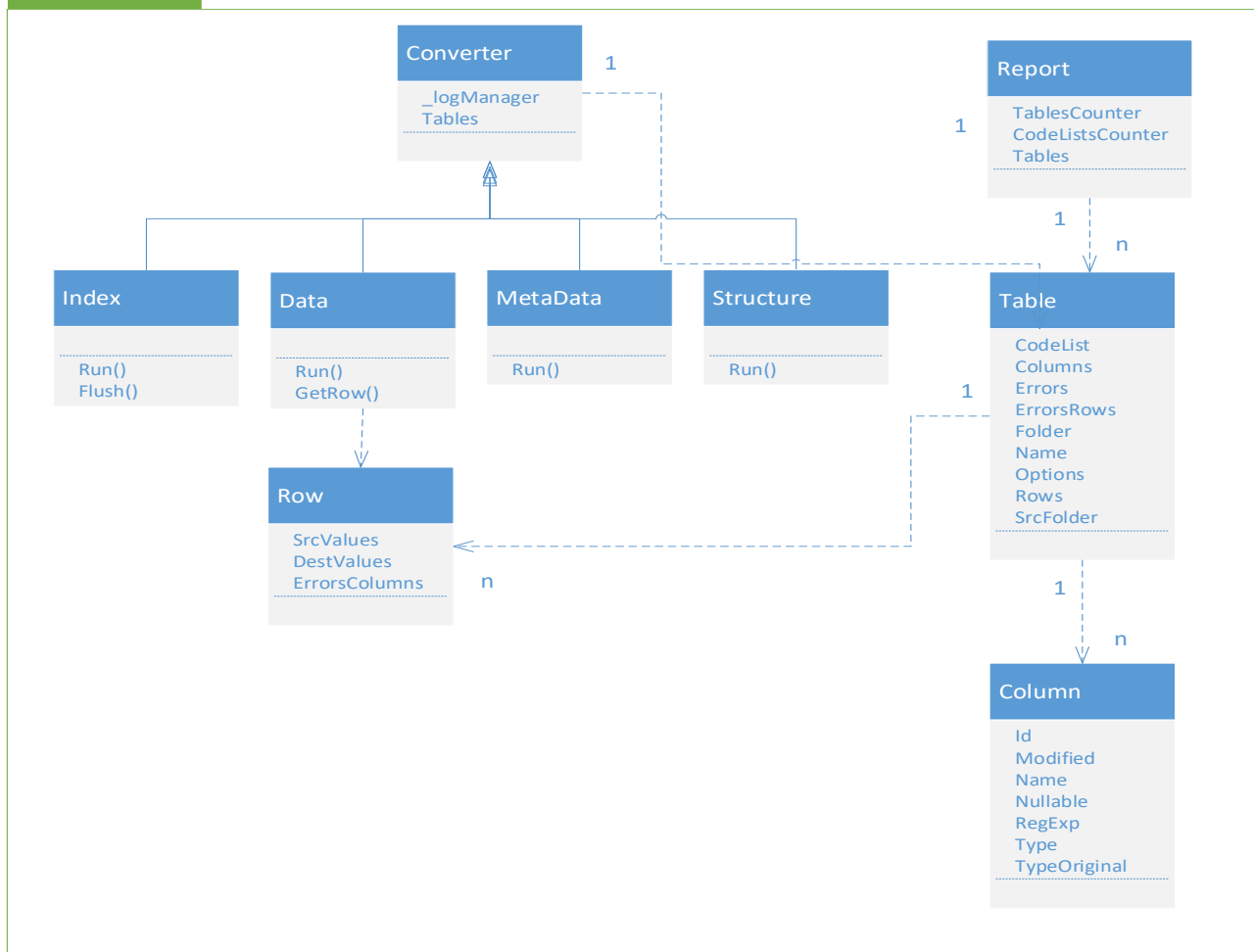
Athena.dll component has 4 inherited converter classes (Structure, MetaData, Data & Index) that's implement Run method. Each of these classes is responsible for partially convert action as followings:

- Structure: create AIP folder structure and copy embedded resource XSD files.
- MetaData: build tableIndex & researchIndex XML files with related code lists data tables
- Data: use stream writer to convert CSV to XML data table files.
- Index: Ensure files indices XML file and create report.

Tables Property in Report class updates through above converter's run methods ends with list of Table objects. Each object contains related conversion output data like Columns objects. Index Flush method generates report. Data GetRow function take Table object and row index return specific detailed Row object with before, after values & errors. The Converter structure takes the followings parameters:

- LogManager: responsible for add & flush custom logs.
- SIP json-file path
- AIP output path
- AIP folder name

Rigsarkiv.Athena



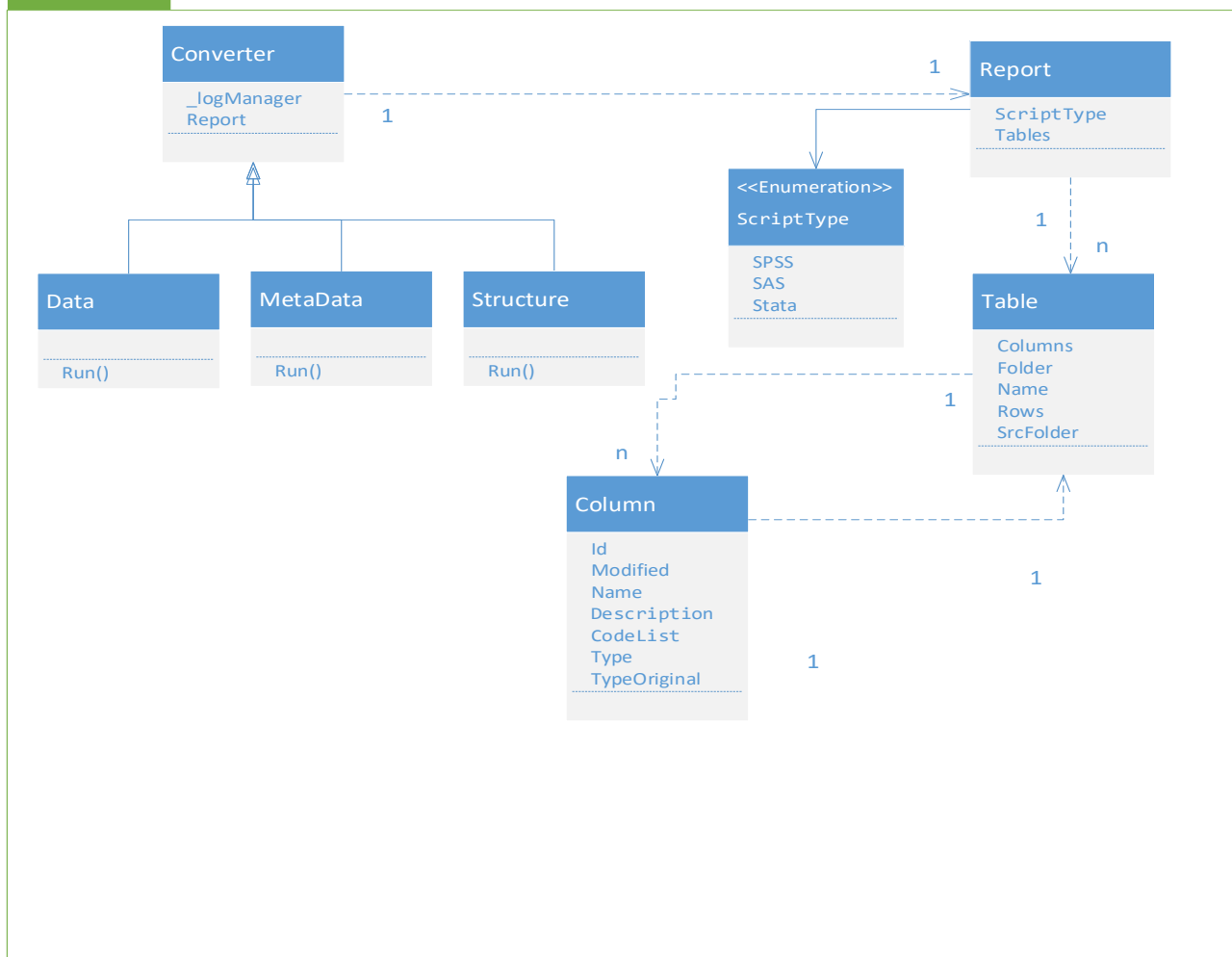
Styx.dll component has 3 inherited converter classes (Structure, MetaData & Data) that's implement Run method. Each of these classes is responsible for partially convert action as followings:

- Structure: create DIP folder structure and copy embedded script files based on script type.
- MetaData: build related texts files for (VARIABLE, VARIABLEBESKRIVELSE, KODELISTE & BRUGERKODE)
- Data: use stream writer to convert XML data to CSV files.

Tables Property in Report class updates through above converter's run methods ends with list of Table objects. Each object contains related conversion output data like Columns objects.

The Converter structure takes the followings parameters:

- LogManager: responsible for add & flush custom logs.
- AIP path
- DIP output path
- DIP folder name
- Script type (SPSS, SAS, Stata)



8. Browser and operating system version supported

Browser version support

Log files from ASTA is developed in HTML and was tested and worked on the following versions of the following browsers (as of September 2019):

- Google Chrome: 77.0.3865.90 (7.7.299) (64 bit)
- Mozilla Firefox: 69.0.2 (64 bit)
- Microsoft Internet Explorer: 11.0.145
- Microsoft Edge: 17.17134
- Safari (tested on Mac): 11.1.2 (13605.3.8) and 12.1.2

NOTE: The Internet Explorer is blocking (as a setting) embedded JavaScript from executing when opening a file, therefore there is need to click a checkbox and allow blocked content in order to make the log files work in this browser.

Operating system version support

ASTA was tested and worked on the following versions of the following operating systems:

- Windows 7 and 10
- macOS High Sierra

9. Languages

Hybris & Nemesis (program 1) are supporting multiple languages. The default language is Danish. JSON (<https://www.json.org>) files are used to save different languages texts. You can use online validation site (<https://jsonlint.com>).

File's name is using LCID standard (https://docs.microsoft.com/en-us/openspecs/office_standards/ms-oe376/6c085406-a698-4e12-9d4d-c3b0ee3dbc4a). Files are placed at electron/assets/languages and structured as array of sections. Each section has a name related to HTML page and keys array. Each key element has key/value pair. All Keys are unique and starting with related section name separating by "-" with extra text. A new language file required add file to release package ([Creation of packages and releases](#))

Followings are examples using Visual Studio Code.

Remember to keep each key name uniquely a cross language file. See da-DK.json example:



In most cases above a key is representing a HTML element id and value will be placed as element value on program starting event. This will be done by using reserved "languages" CSS class. See key's related HTML element example:

```

<> styx.html X
sections > <> styx.html > template.section-template > section#styx > header.section-header > div
1 <template class="section-template">
2   <section id="styx">
3     <header class="section-header">
4       <div class="section-wrapper">
5         <div class="header">
6           <h1 class="title languages" id="styx-Title"></h1>
7           <h1 class="title right">ASTA</h1>
8         </div>
9         <section>
10          <div class="formularContainer">
11            <h2 class="languages" id="styx-title-H2"></h2>
12            <div class="languages" id="styx-body-DIV"></div>
13          </div>
14        </section>
15      </div>
16    </header>
17  </section>
18 </template>

```

Key's value has variant possible contents from simple text to long HTML contents with menu and PDF links.

NOTE: Because a value is string typed remember to use backslash as escape character, see example:

```

{
  "key": "styx-body-DIV",
  "value": "<p>Funktionen \"Skab udleveringsformat\" er valgt og er fremkommet i separat vindue.</p>"
}

```

NOTE: Key's value has also Composite Formatting notation { index } (<https://docs.microsoft.com/en-us/dotnet/standard/base-types/composite-formatting>). These will be replace by self application so do not change them. See:

```

{
  "key": "nemesis-test-id",
  "value": "Testlog: {0}"
},
{
  "key": "nemesis-output-ValidateRows",
  "value": "<br/><br/>{0} af {1} rækker i {2} er valideret."
},

```

NOTE: Key's value has also PDF links references so do not change id attribute since it is used by application.

In order to update link reference to a new language just change self filename then added new file to electron/assets/documents folder. Of course you need also to add new PDF file to release package ([Creation of packages and releases](#)), see:

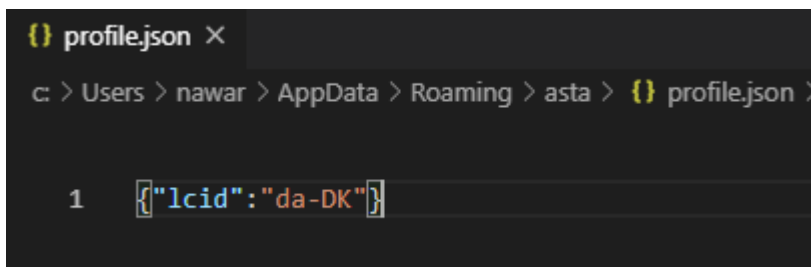


10. Profile

Hybris & Nemesis (program 1) is generating for first run JSON profile file:

- on macOS: ~/Library/Logs/asta/profile.json
- on Windows: %USERPROFILE%\AppData\Roaming\asta\ profile.json

File is saving user language id. This can be changed by user at the profile section.
Save action will restart application. See:



NOTE: You need to delete this file manually on ASTA uninstallation.