ADDML

Archival Data Description Markup Language

Generell del

Versjon PA 0.07
Sist oppdatert: 2010-09-16 TPD

Innledning

Dette dokumentet beskriver det norske arkivverkets standard for teknisk metadata - Archival Data Description Markup Language (ADDML) versjon 8.2.

Dokumentet består av følgende tre hoveddeler

- en innledende del med overordnede opplysninger
- en del som beskriver selve ADDML og de muligheter som ligger i standarden
- en del som beskriver Arkivverkets bruk av standarden.

I tillegg inneholder dokumentet også følgende vedlegg:

- et som beskriver de enkelte elementene i standarden
- et som inneholder selve DTDen
- et som inneholder et XML Schema som tilsvarer DTDen
- et som forklarer overgangen fra ADDMML 7.3 til ADDML 8.2.

ADDML er en standard for å beskrive samlinger av datafiler. En slik samling kalles et datasett og en fil som inneholder beskrivelsen av datasettet, kalles en datasettbeskrivelse.

Mål

ADDML er en standard for å beskrive samlinger av datafiler som er organisert som flate filer. (En flat fil i denne sammenheng betyr at filen er i ren tekst og internt organisert enten ved fast posisjonering eller ved tegnseparasjon.)

En slik samling av filer kalles også for et datasett.

En fil som inneholder beskrivelsen av datasettet kalles en datasettbeskrivelse.

Historie

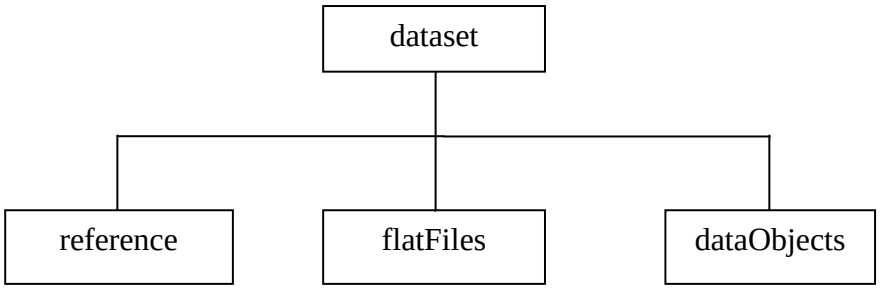
Hvordan benytte ADDML

ADDML benyttes for å beskrive datasett som avleveres eller deponeres i Arkivverket.

Del 1 – Beskrivelse av ADDML

Hva beskrives i ADDML 8.2. (Hovedstruktur)

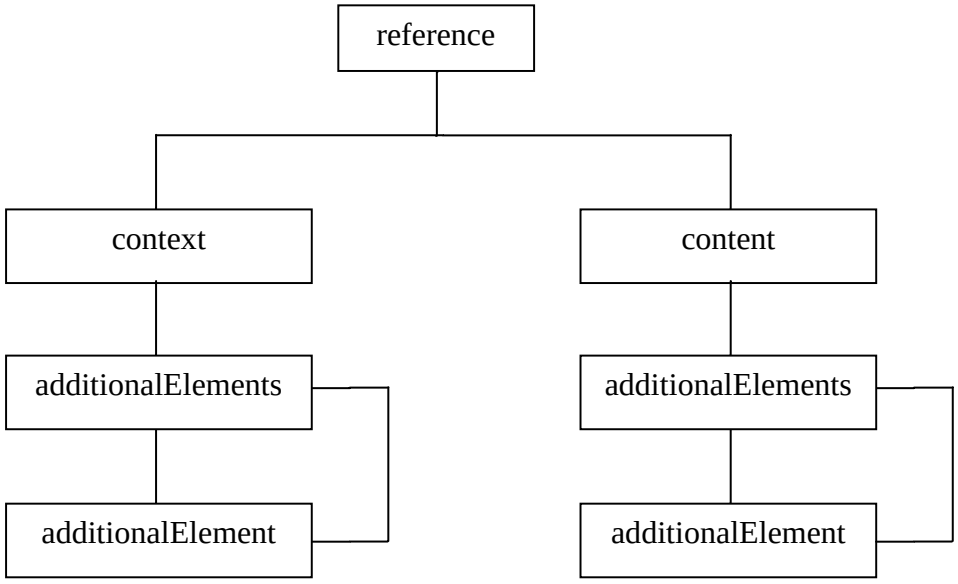
ADDML består av 3 hoveddeler. En del som beskriver tilhørende informasjon, dvs. informasjon om dataene i datasettet - reference. En del som er selve beskrivelsen av flate filer i detalj - flatFiles. En del som gir mulighet for å knytte opp andre filer enn flate filer til beskrivelsen - dataObjects.



Figur 1. Hoveddelene i ADDML.

I *reference* vil det kunne registreres bevaringsverdige metadata på kontekstuell og innholdsmessig nivå. I *flatFiles* vil man kunne registrere en detaljert beskrivelse av strukturen i datafilene dersom de enten er i fast format eller i tegnseparert format. Alle filer som ikke er i fast format eller tegnseparert format vil bli beskrevet i *dataObjects*. Dog vil det her ikke være mulig å gi en detaljert beskrivelse.

Tilhørende informasjon – bevaringsverdige metadata.



Figur 2. Oversikt over elementene i *reference*.

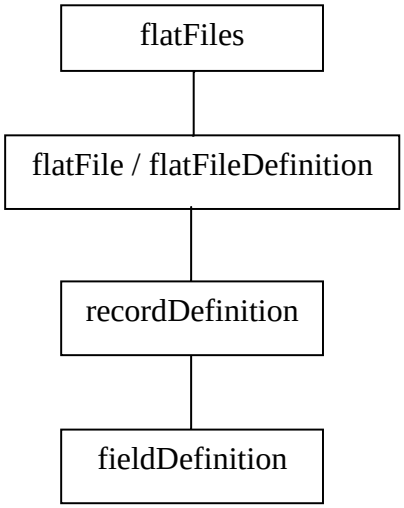
Med tilhørende informasjon menes informasjon om datasettet. Dette kan være opplysninger om hvem som har laget datasettet, hva slags type system det er hentet fra, bakgrunn for dataene i datasettet, osv. Dette er IKKE beskrivende informasjon av selve datasettet.

I ADDML er all slik informasjon samlet i *reference*. *reference* har to underelementer. Disse beskriver følgende:

- kontekstuell informasjon
- innholdsrelatert informasjon

Grupperingene har ingen faste felter, men det kan defineres egne felter som kan eller skal medfølge et datasett. Se eget avsnitt om generiske elementer.

Strukturen for å beskrive flate filer.



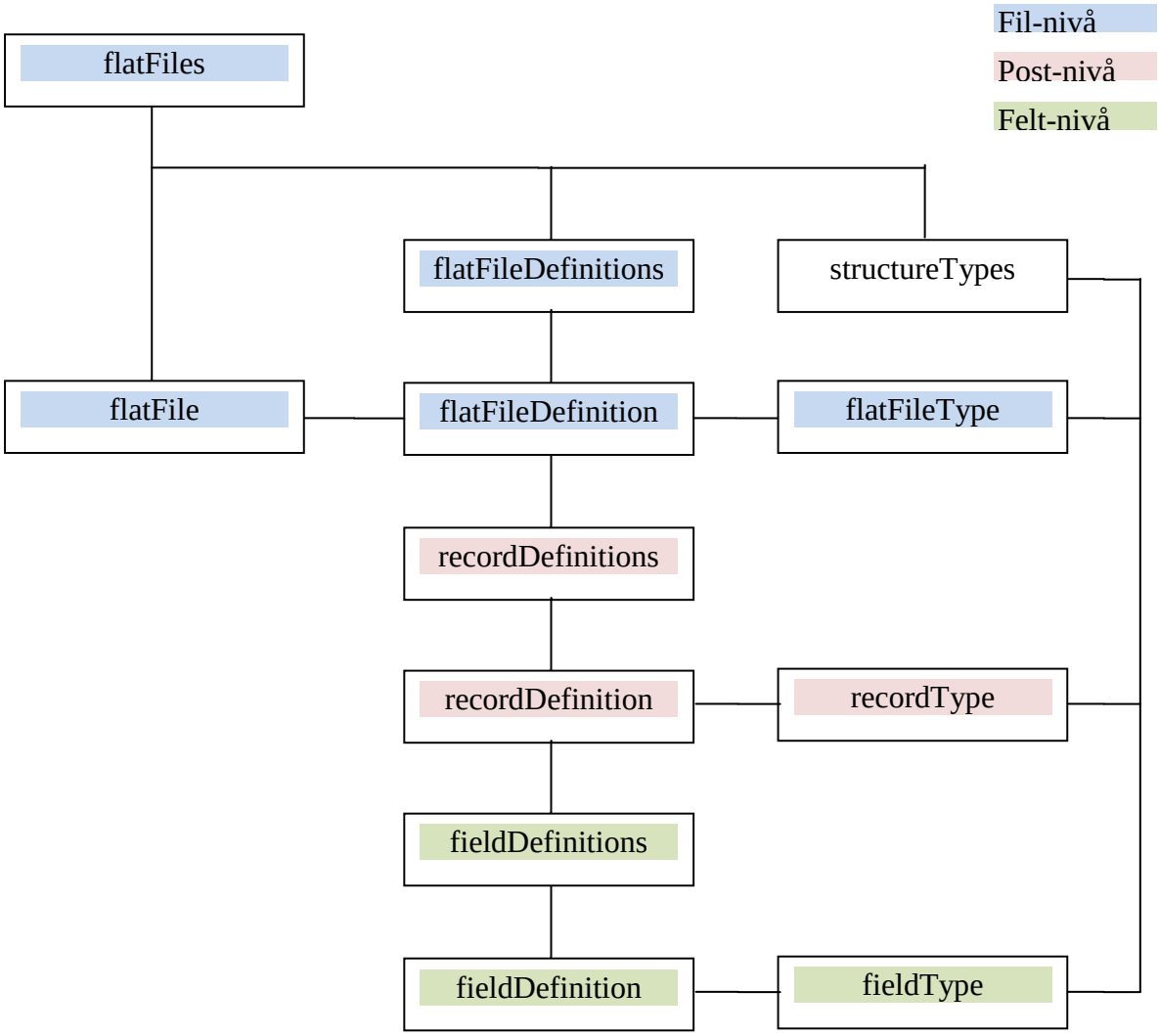
Figur 3. Oversikt over elementene i *dataset* (forenklet form).

Hovedstrukturen i det å beskrive en fil er som i tidligere versjoner av ADDML. Dog er den utvidet og gjort mer generell.

Den tidligere modellen fungerer fortsatt som en forenklet modell. Strukturen har et *flatFiles* (tidligere *structure*) som øverste nivå. Et *flatFiles* kan inneholde en eller flere *flatFile* (tidligere bare *file*). Tilsvarende kan en *flatFile* inneholde en eller flere *recordDefinition* (tidligere *recType*). Og deretter vil så *recordDefinition* inneholde en eller flere *fieldDefinition* (tidligere *field*).

I en vanlig relasjonell database vil det normalt ikke være behov for post-nivået. I andre sammenhenger kan imidlertid en fil inneholde mange forskjellige typer av poster som man ønsker å splitte opp til hver sin tabell. Koblingen mot en database vil derfor være mot post-nivået og ikke mot fil-nivået. Av denne grunnen er derfor også alle informasjoner om nøkler lagt på post-nivået.

Ved overgangen til versjon 8 ble hele denne strukturen revidert og vurdert på nytt. Dette innebar at hvert nivå ble delt opp i tre deler, samt at det for hver ble lagt på et multiplums-nivå med unntak av *dataset*. De tre delene inneholder gjennomgående informasjoner om den fysiske representasjonen av nivået, den definisjonsmessige og en overordnet type. Det er imidlertid i dag ikke funnet behov for den fysiske representasjonen på annet enn fil-nivå. Skulle et slikt behov melde seg senere vil det være en smal sak å legge disse på.

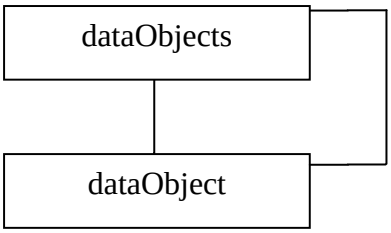


Figur 4. Oversikt over elementene i *dataset* (komplett form).

Dette betyr at strukturen er blitt mer kompleks enn i tidligere versjoner, men samtidig mer fleksibel og generell for bruk.

Andre filer enn flate filer.

Som nevnt over inneholder strukturen kun definering av flate filer med fast eller tegnseparert format. Andre filtyper kan ikke beskrives i detalj vha. ADDML 8.2-elementer. Dog er det mulig å knytte andre typer filer opp mot datasettet som defineres i en ADDML-fil. Dette gjøres ved å definere disse filene som logiske objekter ved å bruke elementene *dataObjects* og *dataObject*. På et logisk objekt kan det så knyttes opp en del egenskaper for å forklare hva slags filer dette er. (Filene kan være datafiler i xml-format, dtd eller xml-skjema, dokumentfiler, bildefiler, lydfiler, videofiler, osv.)



Figur 5. Oversikt over elementene i *dataObjects*.

Generiske elementer.

I de forskjellige beskrivelsene over har det dukket opp flere steder elementer som danner en loop. Det gjelder både for *additionalElements – additionalElement* og *dataObjects – dataObject*. I disse tilfellene er det snakk om en generisk struktur hvor den som benytter standarden selv kan bygge opp en hierarkisk struktur med de nevnte elementene.

Egenskaper.

For mange elementer er det i tillegg muligheten til å legge på egenskaper i form av attributtet *properties*. Også egenskaper har som tilleggselementer en generisk struktur ved *properties – property*.

Prosesser.

Standarden gir i tillegg brukerne muligheten til å definere egne operasjoner som skal utføres på informasjonen. Dette gjøres ved bruk av elementet *processes* som kan inneholdet et sett av elementene *process*, som definerer den enkelte operasjonen. Eksempler på operasjoner som kan tenkes er kontroller (f.eks. kontrollere sjekksum, koder), konverteringer (f.eks. pakke opp pakkede felt, endre fra EBCDIC til ASCII eller UTF-8), osv.

Når det gjelder generiske elementer, egenskaper og prosesser er det helt opp til brukerne av standarden å definere sine egne behov. Standarden gir bare muligheten i form av å definere disse som generiske elementer / attributter.

Gjennomgang av elementene i ADDML 8.2.

Herunder følger en beskrivelse for de viktigste elementene (vil ikke nødvendigvis følge DTD`en eller XML-schema'et).

addml

addml er toppnivået i strukturen. Dette elementet skal eksistere en og kun en gang i henhold til reglene for XML.

dataset

dataset er hovednivået i beskrivelsen. Dette tilsvarer et arkivuttrekk. Imidlertid kan en og samme beskrivelse også inneholde flere *dataset*. Dette for at det skal være mulig å samle beskrivelser når de skal benyttes sammen, for eksempel i en brukssituasjon.

reference

reference er et samlenivå for administrative informasjon om datasettet. Dette nivået skal alltid være med for ethvert *dataset*.

context og content

context inneholder informasjon av kontekstuell art om avleveringen. *content* inneholder informasjon av innholdsmessig art om avleveringen. Både *context* og *content* inneholder muligheter til å definere egne elementer for bruk.

flatFiles og flatFile

flatFiles er en overbygging av filstrukturen. De enkelte filene gjenfinnes i *flatFile*, mens *flatFiles* samler de sammen til en enhet. *flatFile* inneholder informasjon om en enkelt fil på det fysiske planet. Det finnes her referanse til *flatFileDefinition*, en referanse som kan være mange til en.

flatFileDefinitions og flatFileDefinition

Som med beskrivelsen av den fysiske siden, er det også på det overordnede planet en oppdeling av definisjonsnivået i et samleelement og et definert element for hver fil. Disse elementene er *flatFileDefinitions* og *flatFileDefinition* respektive. Fra *flatFileDefinition* finnes det en referanse til *flatFileType*.

structureTypes

For å minske graden av redundant informasjon i datasettbeskrivelsen og for å forenkle registreringen av den, er det innført typer på de tre hovednivåene – fil, post og felt. Disse typene er samlet under elementet *structureTypes*. En type kan defineres slik at flere fil-, post- eller feltdefinisjoner benytter seg av samme fil-, post- eller felttype respektive. Dermed vil all informasjon på typenivå kun registreres en gang.

flatFileTypes og flatFileType

For typene på filnivå benyttes elementene *flatFileTypes* og *flatFileType* – ett samlenivå og ett detaljnivå som vanlig. På filnivå kan det angis et navn på typen, en beskrivelse (hvis man ønsker), samt hvilket karaktersett som er benyttet og om filen er i fast format eller tegnseparert format.

recordTypes og recordType

For typene på postnivå benyttes elementene *recordTypes* og *recordType*, igjen med et samlenivå og et detaljnivå. På postnivå kan det angis et navn på typen, en beskrivelse (hvis man ønsker), samt om feltene er trimmet, altså om ledende nuller og etterfølgende blanke er fjernet i feltene.

fieldTypes og fieldType

For typene på feltnivå benyttes elementene *fieldTypes* og *fieldType*, og som vanlig med et samlenivå og et detaljnivå. På feltnivå kan det angis et navn på typen, en beskrivelse (hvis man ønsker), samt datatypen for feltet, feltformatet, hvilken justering feltet har, om feltet er fylt ut med blanke tegn, hva slags pakking feltet har (dersom det er pakket) og om det er benyttet spesielle tegn for nullverdier.

queries og query

Tanken med disse elementene var å kunne ta vare på SQL-setninger. I denne versjonen er det lagt opp til å kunne ta vare på queries av alle slag, ikke bare fra SQL.

Det er ikke lagt opp til at ADDML i seg selv skal kunne utnytte disse queriene. Men det er selvfølgelig fritt opp til hver applikasjon som benytter seg av en ADDML-fil til å kunne gjøre dette.

processes og process

På tilsvarende måter som med queries kan det også defineres prosesser i en ADDML-fil. Med prosesser menes operasjoner som en eller annet applikasjon skal utføre. Om det er ønskelig kan man benytte prosesser på en annen måte.Elementet *processes* er samlenivået, mens *process* er detaljnivået, helt i stil med hvordan det er ellers i ADDML-standarden.

flatFileProcesses, recordProcesses og fieldProcesses

Disse elementene representerer prosesser på de forskjellige delene av strukturen i en ADDML-fil som beskriver flate filer. Prosess-strukturen følger samme struktur som strukturen med flat fil-, post- og feltdefinisjonene. Selve prosessangivelsene knyttes til ønsket sted i prosess-strukturen

recordDefinitions og recordDefinition

I likhet med som for filer, er det også en tilsvarende struktur på definisjonsnivået for poster. Det er en oppdeling av definisjonsnivået i et samleelement og et definert element for hver post. Disse elementene er *recordDefinitions* og *recordDefinition* respektive. Fra *recordDefinition* finnes det en referanse til *recordType*.

keys og key

Som med det andre er selvfølgelig også nøkler definert med et samlenivå og et detaljnivå. For hver nøkkel vil det angis hva slags type nøkkel det er – primærnøkkel, sekundærnøkkel eller fremmednøkkel.

fieldDefinitions og fieldDefinition

I likhet med filer og poster, er det også en tilsvarende struktur på definisjonsnivået for felter. Med en oppdeling av definisjonsnivået i et samleelement og et definert element for hvert felt. Disse elementene er *fieldDefinitions* og *fieldDefinition* respektive. Fra *fieldDefinition* finnes det en referanse til *fieldType*.

fieldParts

I noen tilfeller er det ønskelig å kunne dele opp et element i mindre deler, samtidig som man også vil ha muligheten til å referere til det hele. Elementet *fieldParts* er ment å dekke dette behovet.

codes og code

Ved hjelp av samleelementet *codes* og detaljelementet *code*, kan koder som kan forekomme i felt i datafilene, tas med som en del av de enkelte feltdefinisjonene. Elementet *code* vil angi kodeverdi og en beskrivelse av denne.

dataObjects og dataObject

Dette er elementer hvor man kan lage en egen hierarkisk struktur for filer som ikke er flate filer. Man må selv bygge opp strukturen på det viset en selv føler blir best mulig.

Hva er nytt i versjon 8.2.

Det er mange ting som er nytt i versjon 8.2.

Foreløpig kan nevnes:

- ADDML-filer skal heretter ha filendelse ~~aml~~ xml. (Egentlig Bestemmelsene, ikke ver. 8.2 – *Må endres i Bestemmelsene*)
- Flere uavhengige datasett kan beskrives i den samme datasettbeskrivelsen.
- Elementet *additionalElements* gjør det mulig for den som [eksempel på hvem] tar ADDML i bruk å definere egne informasjonselementer for elementene *context* og *content* under *reference*.
- Mulighet for å definere egne felt under kontekst og innhold. Disse feltene kan så overføres automatisk som beskrivelse av avleveringen inn i ASTA, Arkis 2, e.l.
- Datafiler i XML-form kan ikke lenger beskrives i detalj vha. ADDML-elementer. Beskrivelsen (strukturen) kan kun beskrive flate filer i fast eller tegnseparert format.
- Det er mulig å registrere tilleggsegenskaper om de flate filene som inngår i datasettbeskrivelsen ved hjelp av elementet *properties*.
  properties kan defineres i en hierarkisk struktur.
  Det er opptil den som bruker ADDML å bestemme hvilke egenskaper som kan registreres om en flat fil.
  [Eksempel]
  Et eksempel på tilleggsegenskaper er filens fysiske navn og plassering, sjekksum, samt hvilken sjekksumalgoritme som er benyttet til å beregne sjekksummen.
- [Om strukturen – *flatFile – flatFileDefinition – recordDefinition - fieldDefinition*]
- [Om typer]
- Strukturbeskrivelsen er splittet opp i typer og definisjoner på de forskjellige nivåene. I tillegg er det også på noen nivåer innført egenskaper. Dette gjør beskrivelsen mer komplisert, men samtidig også mer generell.
  Type-elementene inneholder generell informasjon som kan benyttes på flere definisjoner.
  Definisjons-elementene inneholder spesifikke informasjoner knyttet til den enkelte definisjonen.
- Selv om ADDML primært er tenkt benyttet til å beskrive samlinger av flate filer, kan informasjon om og referanse til alle slags filer inngå ved hjelp av elementene *dataObjects* og *dataObject*.
  [Eksempel]

- Mulighet for å linke inn i datasettet andre filer enn de som beskrives i strukturdelen. Dette gjøres vha. logiske objekter. Et logisk objekt vil ha egenskaper, og de kan defineres i en trestruktur.
- Angivelsen av prosesser som kan utføres på de flate filene, er gjort generell slik at det er opp til den som bruker ADDML å bestemme navnet på prosessene og betydningen av å angi dem i en datasettbeskrivelse.
  [Eksempel]

[Mer detaljert oversikt og beskrivelse vil bli laget senere.]

Overgang fra versjon 7.3 til 8.2.