

Manual: Användning av XML-filer med dataelement från PREMIS för digitalt bevarande

Denna manual beskriver hur XML-filer med dataelement från PREMIS används för att dokumentera digitala objekt i ett bevarandesammanhang. Formatet följer standarden PREMIS enligt dataelementkatalogen och används för att säkerställa spårbarhet, autenticitet och långsiktig tillgänglighet.

XML-filerna ska ses som exempel av hur ett tillägg av tekniska bevarandemetadata till övrig beskrivning kan se ut enligt dataelementkatalogen. XML-filens innehåll kan sammanfogas med de eller den XML-fil som används för ingest, det vill säga arkiveringsmomentet i en e-arkivslösning. Filen som används för detta ändamål innehåller metadata som åtföljer informationen som ska arkiveras och omfattar metadata för övergripande struktur för informationen enligt verksamhetens arkivredovisning, såsom geografiska uppgifter, administrativa uppgifter som processbenämning, diarienummer, datumangivelser och så vidare. Detta innebär att dataelementen kan göras sökbara i ett e-arkiv, både för administratörer och användare, beroende på vad e-arkivslösningen tillåter.

Det är viktigt att observera vilka dataelement som är obligatoriska och inte samt vilka som är repeterbara. Lika viktigt är det att uppmärksamma vilka dataelement som ska hämta sina värden från värdelistor. Detta beskrivs i dataelementkatalogen. Att följa dessa regler gör att den tekniska bevarandemetadatan garanterar ett långtidsbevarande då de grundläggande dataelementen är obligatoriska och beskriver informationen i ett plattformsoberoende format.

Nedanstående beskrivning inkluderar varje övergripande behållare för begriplighet. För den semantiska ordningen, se exempelfilerna.

Det finns två exempelfiler:

- En enklare XML-fil med en övergripande objektkategori som består av endast en fil
- En sammansatt XML-fil med en övergripande objektkategori som består av två filer

Några viktiga regler och rekommendationer:

Identifierare

- Alla objekt, händelser och aktörer ska ha unika UUID
- Samma UUID ska användas i alla referenser

Konsistens

- Alla relationer ska vara dubbelriktade (till exempel objekt, "object" och händelse, "event")

Spårbarhet

- Alla förändringar ska dokumenteras som händelser, "events"

Skalbarhet

- Lägg till fler filer genom att:
 - skapa nytt fil-objekt
 - lägga till i representationens relationship-container
 - koppla till relevanta händelser, "events"

1. Övergripande struktur

En PREMIS XML-fil innehåller dokumentation av tre huvudkomponenter. Dessa tre komponenter används genom en detaljerad dubbelkoppling av underliggande dataelement för att säkra funktionaliteten i de värden som anges och kvaliteten på den spårning som behövs för en hållbar arkivering. De tre komponenterna är:

Del	Beskrivning
Objekt, "object"	<i>Det som bevaras (intellektuell entitet och filer).</i>
Händelse, "event"	<i>Händelser som påverkar objekt</i>
Aktör, "agent"	<i>Aktör som utför händelser</i>

2. Objekt – beskrivning av innehåll

Den information som ska arkiveras består av en eller, som oftast, av en samling filer i ett sammanhang som tillsammans utgör en enhet, ett objekt. Samlingar av filer kan arkiveras som ett arkivpaket för att visa på att deras inbördes relation ska bevaras. För att kunna beskriva vad den övergripande relationen och funktionaliteten hos informationen är används en beskrivning av denna. På samma sätt används objekt att beskriva de ingående filerna i objektet för en enhetlig metadatastruktur.

2.1. Beskrivningen av innehållet

Objektet, "object", ska först identifieras genom ett UUID. I beskrivningen av den information som ska hanteras kan en övergripande objektkategori användas för att beskriva densamma. Här ska man välja mellan värden i en värdelista. En av dessa val, "representation", har valts i den mer avancerade XML-filen där det beskrivs hur två filer hanteras. Denna objektkategori, representation, är alltså den övergripande enheten och kan sägas utgöra det arkivpaket som ska arkiveras.

Exempel:

```
<premis:object>
  <premis:objectIdentifier>
    <premis:objectIdentifierType>UUID</premis:objectIdentifierType>
    <premis:objectIdentifierValue>UUID_REPRESENTATION</premis:objectIdentifierValue>
  </premis:objectIdentifier>
  <premis:objectCategory>representation</premis:objectCategory>
</premis:object>
```

2.2. Fil

På samma sätt beskrivs varje fil som ingår i objektet som ett eget objekt med tillhörande beskrivning. Här upprepas objektets identifiering via UUID men nu med objektkategorin "file".

Exempel:

```
<premis:object>
  <premis:objectIdentifier>
    <premis:objectIdentifierType>UUID</premis:objectIdentifierType>
    <premis:objectIdentifierValue>UUID_FIL_1</premis:objectIdentifierValue>
  </premis:objectIdentifier>
```

```
<premis:objectCategory>file</premis:objectCategory>
</premis:object>
```

Filens egenskaper beskrivs genom ett flertal dataelement. Några av de viktigaste dataelementen för filens beskrivning är:

- filnamn (originalName)
- lagringsplats (storage)
- filstorlek (size)
- format (formatName)

3. Relationer – koppling mellan objekt

För att beskriva kopplingen mellan komponenterna används relationer, ”relationships”, och dess underliggande dataelement genom dubbelkoppling. Ett exempel är ett objekt som kan innehålla en eller flera filer. Det övergripande objektet i exempelfilen med flera filer, ”representation”, kopplas till de två filerna via strukturella relationer. Det finns även här värdelistor som värden ska väljas från. Dessa element är dessa:

Exempel:

```
<premis:object>
  <premis:relationship>
    <premis:relationshipType>structural</premis:relationshipType>
    <premis:relationshipSubType>Has part</premis:relationshipSubType>
  </premis:relationship>
</premis:object>
```

Detta objekt beskrivs här med ordet ”structural” vilket betyder att det finns en strukturell ordning, i detta exempel en filstruktur. Genom nästa värde, ”Has part”, får vi redan på att det finns underliggande delar, alltså filer. Varje del, både objektet och filerna, refereras till med sitt unika UUID. Så här ser referensen till en av filerna ut:

Exempel:

```
<premis:object>
  <premis:relationship>
    <premis:relatedObjectIdentifier>
      <premis:relatedObjectIdentifierType>UUID</premis:relatedObjectIdentifierType>
      <premis:relatedObjectIdentifierValue>UUID_FIL_1</premis:relatedObjectIdentifierValue>
    </premis:relatedObjectIdentifier>
  </premis:relationship>
</premis:object>
```

4. Händelser – dokumentation av händelser

Händelser, ”events”, beskriver vad som har hänt med objekten, alltså i vilka händelser filerna varit hanterade. Detta kan röra sig om en mängd händelser, till exempel konverteringar, komprimeringar, datafångst, kryptering, autentitetskontroller, arkivering eller migrering. Här ska värden hämtas från en värdelista.

Varje händelse identifieras med ett UUID, som här en händelse om handlar om integritetskontroll, ”fixity check”:

Exempel:

```
<premis:event>
  <premis:eventIdentifier>
    <premis:eventIdentifierType>UUID</premis:eventIdentifierType>
    <premis:eventIdentifierValue>UUID_EVENT_FIXITY</premis:eventIdentifierValue>
  </premis:eventIdentifier>
</premis:event>
```

4.1. Exemplet ”Ingest” (arkivering)

I den mer avancerade exempelfilen med två filer beskrivs metadata för en händelse som består av arkiveringen i en e-arkivslösning. Detta kallas ”ingest”, och beskriver när man tillför en fil till ett e-arkiv. Detta moment föregås av en viss process som är olika för olika e-arkivslösningar. Själva momentet heter ”ingestion” i denna värdelista:

Exempel:

```
<premis:event>
  <premis:eventType>ingestion</premis:eventType>
  <premis:eventDateTime>ÅÅÅÅ-MM-DDThh:mm:ss+01:00</premis:eventDateTime>
</premis:event>
```

4.2. Exemplet ”Fixity check” (integritetskontroll)

I den mer avancerade exempelfilen med två filer beskrivs ytterligare metadata för en annan händelse, ”fixity check”, det vill säga en integritetskontroll. Denna kontroll som görs utanför e-arkivslösningen kontrollerar att filen inte har förändrats. Den anges på följande vis:

Exempel:

```
<premis:event>
  <premis:eventType>fixity check</premis:eventType>
  <premis:eventDateTime>ÅÅÅÅ-MM-DDThh:mm:ss+01:00</premis:eventDateTime>
</premis:event>
```

5. Koppling mellan objekt och händelse

Det är viktigt att ange kopplingar mellan objektet, ”object”, och de händelser, ”event”, som skett. Även detta sker alltid i två riktningar. Alla objekt som påverkas av en händelse ska länkas. Här används objektets och händelsens UUID konsekvent för identifiering.

5.1. Från objekt till händelse

Det är viktigt att ange vilka händelser ett objekt varit hanterat i. Detta görs i objekt-containern via en UUID. För att koppla från objektet till händelsen används följande, som här till en ”ingest”-händelse:

Exempel:

```
<premis:object>
```

```

<premis:linkingEventIdentifier>
  <premis:linkingEventIdentifierType>UUID</premis:linkingEventIdentifierType>

<premis:linkingEventIdentifierValue>UUID_EVENT_INGEST</premis:linkingEventIdentifierValue>
</premis:linkingEventIdentifier>
</premis:object>

```

5.2. Från händelse till objekt

Det är också viktigt att koppla händelsen till objektet. Detta görs i händelse-containern via UUID. Här anges också vad resultat blivit, här "outcome". För att koppla från objektet till händelsen används följande, som här till huvudobjektet, representationen:

Exempel:

```

<premis:event>
  <premis:linkingObjectIdentifier>
    <premis:linkingObjectIdentifierType>UUID</premis:linkingObjectIdentifierType>

  <premis:linkingObjectIdentifierValue>UUID_REPRESENTATION</premis:linkingObjectIdentifierValue>
    <premis:linkingObjectRole>outcome</premis:linkingObjectRole>
  </premis:linkingObjectIdentifier>
</premis:event>

```

6. Agent – ansvarig aktör

För att få en fullgod beskrivning är det också möjligt att ange vem eller vad som har utfört händelserna. Det kan här anges en programvara eller en person, t.ex. en administratör.

Även här ska händelsen kopplas till aktören som utfört händelsen och aktören ska kopplas till händelsen. Aktören dokumenteras i containern agent:

Exempel:

```

<premis:agent>
  <premis:agentIdentifier>
    <premis:agentIdentifierType>UUID</premis:agentIdentifierType>
    <premis:agentIdentifierValue>UUID_AGENT</premis:agentIdentifierValue>
  </premis:agentIdentifier>
  <premis:agentName>TEXT</premis:agentName>
  <premis:agentType>TEXT</premis:agentType>
</premis:agent>

```

För att sedan länka aktören till en händelse används koppling i event-containern där beskrivning ges:

Exempel:

```

<premis:event>
  <premis:linkingAgentIdentifier>
    <premis:linkingAgentIdentifierType>UUID</premis:linkingAgentIdentifierType>
    <premis:linkingAgentIdentifierValue>UUID_AGENT</premis:linkingAgentIdentifierValue>
  </premis:linkingAgentIdentifier>

```

<premis:linkingAgentRole>executing program</premis:linkingAgentRole>
</premis:linkingAgentIdentifier>
</premis:event>