Vel bevart!

Rapport fra samarbeidsprosjektet DIAS
– Digital arkivpakkestruktur

Del 2:

Vedlegg 3 - 7 til prosjektrapport

Riksarkivaren

8. juni 2012

Innhold[1]

[1] Spesifikasjonene er også publisert på http://www.arkivverket.no/standarder/dias (lest 03.06.1012)

Vedlegg 3:

DIAS-METS.xsd

METS *(Metadata Encoding & Transmission Standard)* spesifiserer de overordnede elementene i en OAIS-basert arkivpakke: organisering, struktur og innholdsoversikt for pakkeinformasjon. METS er dokumentert på nettstedet *http://www.loc.gov/standards/mets/*

DIAS-METS er DIAS-prosjektets tilpasning av METS for bruk i en felles arkivpakke-struktur for norske statlige og kommunale arkivdepoter. Innbygd i arkivpakkestrukturen, og følgelig i DIAS-METS, brukes følgende andre standarder: EAD og EAC-CPF for arkiv- og aktørbeskrivelse, DIAS-PREMIS for bevaringsmetadata og ADDML for teknisk struktur-beskrivelse. Hver av disse underordnede standardene i arkivpakkestrukturen har et eget xml-skjema.

DIAS-PREMIS, ADDML, EAD og EAC-CPF tilknyttes DIAS-METS som ”plug-ins”. De kan være innkapslet i DIAS-METS, plasseres som utenforliggende objekter som DIAS-METS refererer, eller eventuelt bygge på en kombinasjon av disse to prinsippene.

DIAS-METS er tilgjengelig som xsd-fil på adressen:
http://schema.arkivverket.no/METS/v1.9/DIAS_METS.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This is the adapted schema to profile DIAS -->
<!-- This is the adapted schema to profile SWEIP -->
<!-- METS: Metadata Encoding and Transmission Standard -->
<!-- Copyright © 2001, 2002, 2003, 2004, 2005, 2006, 2008, 2009, 2010 Digital Library Federation -->
- <!--
    Prepared for the Digital Library Federation by Jerome McDonough, New York University,
   with the assistance of Michael Alexander (British Library), Joachim Bauer (Content Conversion Specialists, Germany), Rick Beaubien
-->
<!-- Febuary 2010 -->
<!-- Version 1.9 -->
<!-- Change History -->
<!-- April 23, 2001: Alpha Draft completed -->
<!-- June 7, 2001: Beta completed -->
- <!--
    6/7/2001 Beta Changes:
          1. add 'Time' as a possible time code value, as well as TCF.
          2. Make dmdSec ID attribute required; make ID attribute optional on MDRef/MDWrap.
          3. Add 'Label' attribute to StructMap, along with 'Type'.
          4. Add DDI and FGDC as potential metadata schemes to enumeration.
          5. Enable an "otherMDtype" attribute for MDWrap/MDRef and any other element where
              there's an 'other' in the enumerated possibilities.
          6. Add a "profile" attribute to METS element.
          7. Revised mptr declaration so that it's like FLocat/MDRef (and not like XLink)
          8. Extend internal documentation of <area> attributes.
          9. Add "other" to the possible set of LOCTYPEs.
          10. Change ADMIDS to ADMID on FileGrp.
          11. Change "N" to "Order" on <div> element.
          12. Change "Number" to "order label" on <div> element
          13. Add createdate and lastmoddate attributes to mets element.
          14. Allow <div> and <area> elements to link to administrative metadata sections.
          15. Normalize attribute pointing facilities for file element and mdRef.
          16. Provide a LOCTYPE of "other" and an "otherloctype" attribute for pointing to external files.
          17. Drop PDI from enumeration of LOCTYPES.
          18. Make MDTYPE required in mdRef and mdWrap.
          19. Rename preservationMD to digiprovMD.
          20. Add optional CHECKSUM attribute to FContent element.
          21. Modularize declarations of fileGrpType and mdSecType attributes and enumerations to
                simplify maintenance.
          22. Add TYPE attribute to structMap.
          23. Declare structMap element using structMapType rather than direct declaration.
          24. Add area element as possible subelement to <div>, along with par and seq.
          25. Change mdSec model to ALL, to enable differing order of mdRef/mdWrap elements.
          26. Extend documentation on <par> and <seq> elements.

-->
<!-- October 22, 2001: Gamma completed -->
- <!--
    10/22/2001 Gamma changes:
          1. Added optional fileSec element beneath METS root element to contain fileGrps.
          2. Created subsidiary schema file xlink.xsd for XLink attributes, restored XLink attributes
          to mptr element, and added XLink support to mdRef and FLocat.
          3. Created new element metsHdr to handle metadata regarding METS document
          itself (analogous to TEI Header). Moved CREATEDATE and LASTMODDATE attributes
          to metsHdr, and added new RECORDSTATUS attribute. Added new subsidiary elements
          agent and altRecordID to metsHdr.
          4. Made CREATEDATE and LASTMODDATE attributes type xsd:dateTime to allow more precise
          recording of when work was done.
          5. Changed all attributes using data type of xsd:binary to xsd:base64Binary to conform to final
          W3C schema recommendations.
          6. Cleaned up annotations/documentation.

-->
<!-- December 19, 2001: Epsilon and PROTOFINAL completed -->
- <!--
    12/19/2001 Epsilon changes:
          1. Changed sequence operator for StructMap so that only 1 root div element is permitted.
          2. Add new roles to agent element's role attribute and support for extensible 'other' role.
          3. Add support for extensible 'other' type attribute on agent element.
          4. Yet more documentation clean up.
          5. Relocate CHECKSUM attribute from FContent to File element.
          6. Change the file element's CREATED attribute and fileGroup's VERSDATE attribute to
          a type of xsd:dateTime
          7. Change attribute name DMD for div element to DMDID for consistency's sake.
          8. Added new behaviorSec for support of referencing executable code from METS object

-->
<!-- February 8, 2002: Zeta bug fix to final -->
- <!--
    2/8/2002 Zeta changes:

          1. Eliminated redundant VRA in metadata type enumeration.
          2. Changed mdWrap content model, adding xmlData element to eliminate
                ambiguous content model

-->
<!-- June 3, 2002: Version 1.1 -->
- <!--
    6/3/2002 v1.1 changes:

          1. Add new structLink section for recording hyperlinks between media represented by structMap nodes.
          2. Allow a <par> element to
          contain a <seq>
-->
<!-- Dec. 27, 2002: Version 1.2 -->
- <!--
    12/27/2002 v1.2 changes:
   1. Add "USE" attribute to FileGrp, File, FLocat and FContent;
   2. Make FLocat repeatable;
   3. Have FContent mimic mdWrap in using separate binData/xmlData sections;
   4. Copyright statement added;
   5. Allow both FLocat and Fcontent in single file element;
   6. Allow behaviorSec elements to group through GROUPID attribute;
```

```
    7. allow descriptive and administrative metadata sections to be grouped through GROUPID attribute;
    8. allow <file> element to point to descriptive metadata via DMDID attribute;
    9. allow descriptive metadata and all forms of administrative metadata to point to administrative metadata via ADMID attribute;
    10. CREATED and STATUS attributes added to all desc. and adm. metadata sections; and
    11. clean up documentation in elements to reflect reality.
  -->
  <!-- May 8, 2003: Version 1.3 -->
- <!--
    05/05/2003 v1.3 changes:

    1. Change "2. OBJID: a primary identifier assigned to the original source document" to "2. OBJID: a primary identifier assigned to
    2. Add MODS to MDTYPEs.
    3. Modify <file> attributes so that instead of just CHECKSUM we have CHECKSUM and CHECKSUMTYPE, where CHECKSUMTYPE is a controlled
        HAVAL, MD5, SHA-1, SHA-256, SHA-384, SHA-512, TIGER, WHIRLPOOL
    4.Alter BehaviorSec to make it recursive, and add a new behavior element to wrap mechanism and interfaceDef elements.
  -->
  <!-- May 1, 2004: Version 1.4 -->
- <!--
    05/01/2003 v1.4 changes:

    1. Moved attribute documentation out of element documentation
    (thank you, Brian Tingle).
    2. New CONTENTIDS attribute (and URIs simpleType) added to div, fptr,
    mptr and area elements for mapping MPEG21 DII Identifier values
    3. XLink namespace URI changed to conform with XLink recommendation.
    4. ID Attribute added to FContent.
    5. ID Attribute addedt to structLink.
    6. ID Attribute added to smLink.
    7. "LOM" added as metadata type.

  -->
  <!-- April 12, 2005: Version 1.5 -->
- <!--
    04/12/2005 v1.5 changes:

    1. Made file element recursive to deal with PREMIS Onion Layer model and
    support XFDU-ish unpacking specification.
    2. Add <stream> element beneath <file> to allow linking of metadata to
    subfile structures.
    3. Modify structLink TO and FROM attributes to put them in XLink namespace.
    4. Make processContents "lax" for all xsd:any elements.

  -->
  <!-- October 18, 2006: Version 1.6 -->
- <!--
    10/18/2006 v1.6 changes:

    1. add ID to stream and transformFile
    2. add ADMID to metsHdr
    3. make smLink/@xlink:to and smLink/@xlink:from required

  -->
  <!-- October 16, 2007/ Jan 20, 2008: Version 1.7 -->
- <!--
    10/16/2007 01/30/2008 v 1.7 changes:

    1. create parType complex type to allow a seq to contain a par
    2. create FILECORE attribute group with MIMETYPE, SIZE, CHECKSUM, CHECKSUMTYPE;
        change fileType, mdWrapType and mdRefType use the attribute group, so mdType and mdRef end
        up with new SIZE, CHECKSUM, and CHECKSUMTYPE attributes (file does not change)
    20080130
    2a. CREATED added to FILECORE
    3. PREMIS:OBJECT PREMIS:AGENT PREMIS:RIGHTS PREMIS:EVENT added to MDTYPE value enumeration
  -->
  <!-- April 2009: Version 1.8 -->
- <!--
    Version 1.8 changes:
            1. Add CRC32, Adler-32, MNP to the enumerated values constraining CHECKSUMTYPE to align with MIX messageDigestAlgorithm con
            2. Add TEXTMD and METSRIGHTS to the enumeration values constraining MDTYPE.
            3. Add an MDTYPEVERSION attribute as a companion to the MDTYPE attribute in the mdRef and mdWrap elements.
            4. ID and STRUCTID attributes on the behavior element made optional. Depending on whether the behavior applies to a transf
            5. Documentation aligned with the METS Primer, and corrected.
            6. xml:lang="en" atttribute value added to every <documentation> element
            7. xlink:extendedLink support added to the <structLink> element by means of a new <smLinkGrp> element, and its child <smLoc
  -->
  <!-- February 2010: Version 1.9 -->
- <!--
    Version 1.9 Changes:
            1. Added a <metsDocumentID> element to the <metsHdr> for recording a unique identifier for the METS document itself where t
            2. Added "ISO 19115:2003 NAP" to the enumerated values for the MDTYPE attribute in the METADATA attribute group.
            3. Added "XPTR" to the enumerated values for the BETYPE attribute on the areaType data type
            4. Added BEGIN, END and BETYPE attributes to the <file> and <stream> elements for specifying the location of a nested file
  -->
- <xsd:schema targetNamespace="http://arkivverket.no/standarder/METS" xmlns="http://arkivverket.no/standarder/METS"
    xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
    <xsd:import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlink.xsd" />
  - <xsd:element name="mets">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">METS: Metadata Encoding and Transmission Standard. METS is intended to provide a
          standardized XML format for transmission of complex digital library objects between systems. As such, it can be seen as
          filling a role similar to that defined for the Submission Information Package (SIP), Archival Information Package (AIP) and
          Dissemination Information Package (DIP) in the Reference Model for an Open Archival Information System. The root
          element <mets> establishes the container for the information being stored and/or transmitted by the
          standard.</xsd:documentation>
      </xsd:annotation>
    - <xsd:complexType>
      - <xsd:complexContent>
          <xsd:extension base="metsType" />
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:element>
```

```
- <xsd:complexType name="metsType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">metsType: Complex Type for METS Sections A METS document consists of seven possible subsidiary sections: metsHdr (METS document header), dmdSec (descriptive metadata section), amdSec (administrative metadata section), fileGrp (file inventory group), structLink (structural map linking), structMap (structural map) and behaviorSec (behaviors section).</xsd:documentation>
    </xsd:annotation>
  - <xsd:sequence>
    - <xsd:element name="metsHdr">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">The mets header element <metsHdr> captures metadata about the METS document itself, not the digital object the METS document encodes. Although it records a more limited set of metadata, it is very similar in function and purpose to the headers employed in other schema such as the Text Encoding Initiative (TEI) or in the Encoded Archival Description (EAD).</xsd:documentation>
          <xsd:documentation xml:lang="en">Made element mandatory instead of optional, Karin Bredenberg minOccurs="0"</xsd:documentation>
          <xsd:documentation xml:lang="en">At least two agents must be present, Karin Bredenberg (originaly 1)
          </xsd:documentation>
        </xsd:annotation>
      - <xsd:complexType>
        - <xsd:sequence>
          - <xsd:element name="agent" minOccurs="6" maxOccurs="unbounded">
            - <xsd:annotation>
                <xsd:documentation xml:lang="en">agent: The agent element <agent> provides for various parties and their roles with respect to the METS record to be documented.</xsd:documentation>
                <xsd:documentation xml:lang="en">At least one agent must be present, Karin Bredenberg</xsd:documentation>
                <xsd:documentation xml:lang="en">In DIAS 6 agents must be present, Karin Bredenberg</xsd:documentation>
              </xsd:annotation>
            - <xsd:complexType>
              - <xsd:sequence>
                - <xsd:element name="name" type="xsd:string">
                  - <xsd:annotation>
                      <xsd:documentation xml:lang="en">The element <name> can be used to record the full name of the document agent.</xsd:documentation>
                    </xsd:annotation>
                  </xsd:element>
                - <xsd:element name="note" type="xsd:string" minOccurs="0" maxOccurs="unbounded">
                  - <xsd:annotation>
                      <xsd:documentation xml:lang="en">The <note> element can be used to record any additional information regarding the agent's activities with respect to the METS document.</xsd:documentation>
                    </xsd:annotation>
                  </xsd:element>
                </xsd:sequence>
              - <xsd:attribute name="ID" type="xsd:ID" use="optional">
                - <xsd:annotation>
                    <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
                  </xsd:annotation>
                </xsd:attribute>
              - <xsd:attribute name="ROLE" use="required">
                - <xsd:annotation>
                    <xsd:documentation xml:lang="en">ROLE (string/R): Specifies the function of the agent with respect to the METS record. The allowed values are: CREATOR: The person(s) or institution(s) responsible for the METS document. EDITOR: The person(s) or institution(s) that prepares the metadata for encoding. ARCHIVIST: The person(s) or institution(s) responsible for the document/collection. PRESERVATION: The person(s) or institution(s) responsible for preservation functions. DISSEMINATOR: The person(s) or institution(s) responsible for dissemination functions. CUSTODIAN: The person(s) or institution(s) charged with the oversight of a document/collection. IPOWNER: Intellectual Property Owner: The person(s) or institution holding copyright, trade or service marks or other intellectual property rights for the object. OTHER: Use OTHER if none of the preceding values pertains and clarify the type and location specifier being used in the OTHERROLE attribute (see below).</xsd:documentation>
                  </xsd:annotation>
                - <xsd:simpleType>
                  - <xsd:restriction base="xsd:string">
                      <xsd:enumeration value="CREATOR" />
                      <xsd:enumeration value="EDITOR" />
                      <xsd:enumeration value="ARCHIVIST" />
                      <xsd:enumeration value="PRESERVATION" />
                      <xsd:enumeration value="DISSEMINATOR" />
                      <xsd:enumeration value="CUSTODIAN" />
                      <xsd:enumeration value="IPOWNER" />
                      <xsd:enumeration value="OTHER" />
                    </xsd:restriction>
                  </xsd:simpleType>
                </xsd:attribute>
              - <xsd:attribute name="OTHERROLE" type="xsd:string" use="optional">
                - <xsd:annotation>
                    <xsd:documentation xml:lang="en">OTHERROLE (string/O): Denotes a role not contained in the allowed values set if OTHER is indicated in the ROLE attribute.</xsd:documentation>
                  </xsd:annotation>
                </xsd:attribute>
              - <xsd:attribute name="TYPE" use="required">
                - <xsd:annotation>
                    <xsd:documentation xml:lang="en">TYPE (string/O): is used to specify the type of AGENT. It must be one of the following values: INDIVIDUAL: Use if an individual has served as the agent. ORGANIZATION: Use if an institution, corporate body, association, non-profit enterprise, government, religious body, etc. has served as the agent. OTHER: Use OTHER if none of the preceding values pertain and clarify the type of agent specifier being used in the OTHERTYPE attribute</xsd:documentation>
                    <xsd:documentation xml:lang="en">Changed use from optional to mandatory, Karin Bredenberg</xsd:documentation>
```

```
        </xsd:annotation>
      - <xsd:simpleType>
        - <xsd:restriction base="xsd:string">
            <xsd:enumeration value="INDIVIDUAL" />
            <xsd:enumeration value="ORGANIZATION" />
            <xsd:enumeration value="OTHER" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    - <xsd:attribute name="OTHERTYPE" use="optional">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">OTHERTYPE (string/O): Specifies the type of agent when the value
            OTHER is indicated in the TYPE attribute.</xsd:documentation>
          <xsd:documentation xml:lang="en">Removed type, Karin Bredenberg
            type="xsd:string"</xsd:documentation>
          <xsd:documentation xml:lang="en">Added valuelist, Karin Bredenberg</xsd:documentation>
        </xsd:annotation>
      - <xsd:simpleType>
        - <xsd:restriction base="xsd:string">
            <xsd:enumeration value="SOFTWARE" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
- <xsd:element name="altRecordID" minOccurs="0" maxOccurs="unbounded">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">The alternative record identifier element <altRecordID> allows one to use
        alternative record identifier values for the digital object represented by the METS document; the primary
        record identifier is stored in the OBJID attribute in the root <mets> element.</xsd:documentation>
    </xsd:annotation>
  - <xsd:complexType>
    - <xsd:simpleContent>
      - <xsd:extension base="xsd:string">
        - <xsd:attribute name="ID" type="xsd:ID" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element
                within the METS document, and would allow the element to be referenced unambiguously
                from another element or document via an IDREF or an XPTR. For more information on using
                ID attributes for internal and external linking see Chapter 4 of the METS
                Primer.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute name="TYPE" type="xsd:string" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">TYPE (string/O): A description of the identifier type (e.g.,
                OCLC record number, LCCN, etc.).</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
- <xsd:element name="metsDocumentID">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">The metsDocument identifier element <metsDocumentID> allows a unique
        identifier to be assigned to the METS document itself. This may be different from the OBJID attribute value
        in the root <mets> element, which uniquely identifies the entire digital object represented by the METS
        document.</xsd:documentation>
      <xsd:documentation xml:lang="en">Made element mandatory, Karin Bredenberg (minOccurs="0")
        </xsd:documentation>
    </xsd:annotation>
  - <xsd:complexType>
    - <xsd:simpleContent>
      - <xsd:extension base="xsd:string">
        - <xsd:attribute name="ID" type="xsd:ID" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element
                within the METS document, and would allow the element to be referenced unambiguously
                from another element or document via an IDREF or an XPTR. For more information on using
                ID attributes for internal and external linking see Chapter 4 of the METS
                Primer.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute name="TYPE" type="xsd:string" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">TYPE (string/O): A description of the identifier
                type.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>
- <xsd:attribute name="ID" type="xsd:ID" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS
        document, and would allow the element to be referenced unambiguously from another element or document
        via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see
        Chapter 4 of the METS Primer.</xsd:documentation>
    </xsd:annotation>
</xsd:attribute>
- <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional">
  - <xsd:annotation>
```

```
            <xsd:documentation xml:lang="en">ADMID (IDREFS/O): Contains the ID attribute values of the <techMD>,
              <sourceMD>, <rightsMD> and/or <digiprovMD> elements within the <amdSec> of the METS document that
              contain administrative metadata pertaining to the METS document itself. For more information on using METS
              IDREFS and IDREF type attributes for internal linking, see Chapter 4 of the METS Primer.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      - <xsd:attribute name="CREATEDATE" type="xsd:dateTime" use="required">
        - <xsd:annotation>
            <xsd:documentation xml:lang="en">CREATEDATE (dateTime/O): Records the date/time the METS document was
              created.</xsd:documentation>
            <xsd:documentation xml:lang="en">Made attribut required instead of optional, Karin
              Bredenberg</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      - <xsd:attribute name="LASTMODDATE" type="xsd:dateTime" use="optional">
        - <xsd:annotation>
            <xsd:documentation xml:lang="en">LASTMODDATE (dateTime/O): Is used to indicate the date/time the METS
              document was last modified.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      - <xsd:attribute name="RECORDSTATUS" type="xsd:string" use="optional">
        - <xsd:annotation>
            <xsd:documentation xml:lang="en">RECORDSTATUS (string/O): Specifies the status of the METS document. It is
              used for internal processing purposes.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:complexType>
    </xsd:element>
  - <xsd:element name="dmdSec" type="mdSecType" minOccurs="0" maxOccurs="unbounded">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">A descriptive metadata section <dmdSec> records descriptive metadata pertaining to
          the METS object as a whole or one of its components. The <dmdSec> element conforms to same generic datatype as
          the <techMD>, <rightsMD>, <sourceMD> and <digiprovMD> elements, and supports the same sub-elements and
          attributes. A descriptive metadata element can either wrap the metadata (mdWrap) or reference it in an external
          location (mdRef) or both. METS allows multiple <dmdSec> elements; and descriptive metadata can be associated
          with any METS element that supports a DMDID attribute. Descriptive metadata can be expressed according to many
          current description standards (i.e., MARC, MODS, Dublin Core, TEI Header, EAD, VRA, FGDC, DDI) or a locally
          produced XML schema.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  - <xsd:element name="amdSec" type="amdSecType" minOccurs="0" maxOccurs="unbounded">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">The administrative metadata section <amdSec> contains the administrative
          metadata pertaining to the digital object, its components and any original source material from which the digital
          object is derived. The <amdSec> is separated into four sub-sections that accommodate technical metadata (techMD),
          intellectual property rights (rightsMD), analog/digital source metadata (sourceMD), and digital provenance metadata
          (digiprovMD). Each of these subsections can either wrap the metadata (mdWrap) or reference it in an external
          location (mdRef) or both. Multiple instances of the <amdSec> element can occur within a METS document and
          multiple instances of its subsections can occur in one <amdSec> element. This allows considerable flexibility in the
          structuring of the administrative metadata. METS does not define a vocabulary or syntax for encoding administrative
          metadata. Administrative metadata can be expressed within the amdSec sub-elements according to many current
          community defined standards, or locally produced XML schemas.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  - <xsd:element name="fileSec" minOccurs="0">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">The overall purpose of the content file section element <fileSec> is to provide an
          inventory of and the location for the content files that comprise the digital object being described in the METS
          document.</xsd:documentation>
      </xsd:annotation>
    - <xsd:complexType>
      - <xsd:sequence>
        - <xsd:element name="fileGrp" maxOccurs="unbounded">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">A sequence of file group elements <fileGrp> can be used group the digital
                files comprising the content of a METS object either into a flat arrangement or, because each file group
                element can itself contain one or more file group elements, into a nested (hierarchical) arrangement. In
                the case where the content files are images of different formats and resolutions, for example, one could
                group the image content files by format and create a separate <fileGrp> for each image format/resolution
                such as: -- one <fileGrp> for the thumbnails of the images -- one <fileGrp> for the higher resolution JPEGs
                of the image -- one <fileGrp> for the master archival TIFFs of the images For a text resource with a variety
                of content file types one might group the content files at the highest level by type, and then use the
                <fileGrp> element's nesting capabilities to subdivide a <fileGrp> by format within the type, such as: -- one
                <fileGrp> for all of the page images with nested <fileGrp> elements for each image format/resolution (tiff,
                jpeg, gif) -- one <fileGrp> for a PDF version of all the pages of the document -- one <fileGrp> for a TEI
                encoded XML version of the entire document or each of its pages. A <fileGrp> may contain zero or more
                <fileGrp> elements and or <file> elements.</xsd:documentation>
            </xsd:annotation>
          - <xsd:complexType>
            - <xsd:complexContent>
                <xsd:extension base="fileGrpType" />
              </xsd:complexContent>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      - <xsd:attribute name="ID" type="xsd:ID" use="optional">
        - <xsd:annotation>
            <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS
              document, and would allow the element to be referenced unambiguously from another element or document
              via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see
              Chapter 4 of the METS Primer.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:complexType>
    </xsd:element>
  - <xsd:element name="structMap" type="structMapType" maxOccurs="unbounded">
```

```
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">The structural map section <structMap> is the heart of a METS document. It provides a means for organizing the digital content represented by the <file> elements in the <fileSec> of the METS document into a coherent hierarchical structure. Such a hierarchical structure can be presented to users to facilitate their comprehension and navigation of the digital content. It can further be applied to any purpose requiring an understanding of the structural relationship of the content files or parts of the content files. The organization may be specified to any level of granularity (intellectual and or physical) that is desired. Since the <structMap> element is repeatable, more than one organization can be applied to the digital content represented by the METS document. The hierarchical structure specified by a <structMap> is encoded as a tree of nested <div> elements. A <div> element may directly point to content via child file pointer <fptr> elements (if the content is represented in the <fileSec<) or child METS pointer <mptr> elements (if the content is represented by an external METS document). The <fptr> element may point to a single whole <file> element that manifests its parent <div<, or to part of a <file> that manifests its <div<. It can also point to multiple files or parts of files that must be played/displayed either in sequence or in parallel to reveal its structural division. In addition to providing a means for organizing content, the <structMap> provides a mechanism for linking content at any hierarchical level with relevant descriptive and administrative metadata.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    - <xsd:element name="structLink" minOccurs="0">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">The structural link section element <structLink> allows for the specification of hyperlinks between the different components of a METS structure that are delineated in a structural map. This element is a container for a single, repeatable element, <smLink> which indicates a hyperlink between two nodes in the structural map. The <structLink> section in the METS document is identified using its XML ID attributes.</xsd:documentation>
        </xsd:annotation>
      - <xsd:complexType>
        - <xsd:complexContent>
            <xsd:extension base="structLinkType" />
          </xsd:complexContent>
        </xsd:complexType>
      </xsd:element>
    - <xsd:element name="behaviorSec" type="behaviorSecType" minOccurs="0" maxOccurs="unbounded">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">A behavior section element <behaviorSec> associates executable behaviors with content in the METS document by means of a repeatable behavior <behavior> element. This element has an interface definition <interfaceDef> element that represents an abstract definition of the set of behaviors represented by a particular behavior section. A <behavior> element also has a <mechanism> element which is used to point to a module of executable code that implements and runs the behavior defined by the interface definition. The <behaviorSec> element, which is repeatable as well as nestable, can be used to group individual behaviors within the structure of the METS document. Such grouping can be useful for organizing families of behaviors together or to indicate other relationships between particular behaviors.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  - <xsd:attribute name="ID" type="xsd:ID" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="OBJID" type="xsd:string" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">OBJID (string/O): Is the primary identifier assigned to the METS object as a whole. Although this attribute is not required, it is strongly recommended. This identifier is used to tag the entire METS object to external systems, in contrast with the ID identifier.</xsd:documentation>
        <xsd:documentation xml:lang="en">Made attribute mandatory instead of optional, Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="LABEL" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">LABEL (string/O): Is a simple title string used to identify the object/entity being described in the METS document for the user.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="TYPE" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">TYPE (string/O): Specifies the class or type of the object, e.g.: book, journal, stereograph, dataset, video, etc.</xsd:documentation>
        <xsd:documentation xml:lang="en">Made the attribut mandatory instead of optional, Karin Bredenberg</xsd:documentation>
        <xsd:documentation xml:lang="en">Added valuelist, Karin Bredenberg</xsd:documentation>
        <xsd:documentation xml:lang="en">Removde type, Karin Bredenberg type="xsd:string"</xsd:documentation>
        <xsd:documentation xml:lang="en">Added values from DIAS, Karin Bredenberg 2010-12-01</xsd:documentation>
      </xsd:annotation>
    - <xsd:simpleType>
      - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="SIP" />
          <xsd:enumeration value="AIP" />
          <xsd:enumeration value="DIP" />
          <xsd:enumeration value="AIC" />
          <xsd:enumeration value="AIU" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  - <xsd:attribute name="PROFILE" type="xsd:string" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">PROFILE (string/O): Indicates to which of the registered profile(s) the METS document conforms. For additional information about PROFILES see Chapter 5 of the METS Primer.</xsd:documentation>
        <xsd:documentation xml:lang="en">MAde attribute mandatory instead of optional, Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
- <xsd:complexType name="amdSecType">
```

```
- <xsd:annotation>
    <xsd:documentation xml:lang="en">amdSecType: Complex Type for Administrative Metadata Sections The administrative metadata section consists of four possible subsidiary sections: techMD (technical metadata for text/image/audio/video files), rightsMD (intellectual property rights metadata), sourceMD (analog/digital source metadata), and digiprovMD (digital provenance metadata, that is, the history of migrations/translations performed on a digital library object from it's original digital capture/encoding).</xsd:documentation>
  </xsd:annotation>
- <xsd:sequence>
  - <xsd:element name="techMD" type="mdSecType" minOccurs="0" maxOccurs="unbounded">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">A technical metadata element <techMD> records technical metadata about a component of the METS object, such as a digital content file. The <techMD> element conforms to same generic datatype as the <dmdSec>, <rightsMD>, <sourceMD> and <digiprovMD> elements, and supports the same sub-elements and attributes. A technical metadata element can either wrap the metadata (mdWrap) or reference it in an external location (mdRef) or both. METS allows multiple <techMD> elements; and technical metadata can be associated with any METS element that supports an ADMID attribute. Technical metadata can be expressed according to many current technical description standards (such as MIX and textMD) or a locally produced XML schema.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  - <xsd:element name="rightsMD" type="mdSecType" minOccurs="0" maxOccurs="unbounded">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">An intellectual property rights metadata element <rightsMD> records information about copyright and licensing pertaining to a component of the METS object. The <rightsMD> element conforms to same generic datatype as the <dmdSec>, <techMD>, <sourceMD> and <digiprovMD> elements, and supports the same sub-elements and attributes. A rights metadata element can either wrap the metadata (mdWrap) or reference it in an external location (mdRef) or both. METS allows multiple <rightsMD> elements; and rights metadata can be associated with any METS element that supports an ADMID attribute. Rights metadata can be expressed according current rights description standards (such as CopyrightMD and rightsDeclarationMD) or a locally produced XML schema.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  - <xsd:element name="sourceMD" type="mdSecType" minOccurs="0" maxOccurs="unbounded">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">A source metadata element <sourceMD> records descriptive and administrative metadata about the source format or media of a component of the METS object such as a digital content file. It is often used for discovery, data administration or preservation of the digital object. The <sourceMD> element conforms to same generic datatype as the <dmdSec>, <techMD>, <rightsMD>, and <digiprovMD> elements, and supports the same sub-elements and attributes. A source metadata element can either wrap the metadata (mdWrap) or reference it in an external location (mdRef) or both. METS allows multiple <sourceMD> elements; and source metadata can be associated with any METS element that supports an ADMID attribute. Source metadata can be expressed according to current source description standards (such as PREMIS) or a locally produced XML schema.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  - <xsd:element name="digiprovMD" type="mdSecType" minOccurs="0" maxOccurs="unbounded">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">A digital provenance metadata element <digiprovMD> can be used to record any preservation-related actions taken on the various files which comprise a digital object (e.g., those subsequent to the initial digitization of the files such as transformation or migrations) or, in the case of born digital materials, the files' creation. In short, digital provenance should be used to record information that allows both archival/library staff and scholars to understand what modifications have been made to a digital object and/or its constituent parts during its life cycle. This information can then be used to judge how those processes might have altered or corrupted the object's ability to accurately represent the original item. One might, for example, record master derivative relationships and the process by which those derivations have been created. Or the <digiprovMD> element could contain information regarding the migration/transformation of a file from its original digitization (e.g., OCR, TEI, etc.,)to its current incarnation as a digital object (e.g., JPEG2000). The <digiprovMD> element conforms to same generic datatype as the <dmdSec>, <techMD>, <rightsMD>, and <sourceMD> elements, and supports the same sub-elements and attributes. A digital provenance metadata element can either wrap the metadata (mdWrap) or reference it in an external location (mdRef) or both. METS allows multiple <digiprovMD> elements; and digital provenance metadata can be associated with any METS element that supports an ADMID attribute. Digital provenance metadata can be expressed according to current digital provenance description standards (such as PREMIS) or a locally produced XML schema.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
- <xsd:attribute name="ID" type="xsd:ID" use="required">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
      <xsd:documentation xml:lang="en">Changed use from optional to mandatory, Karin Bredenberg</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
- <xsd:complexType name="fileGrpType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">fileGrpType: Complex Type for File Groups The file group is used to cluster all of the digital files composing a digital library object in a hierarchical arrangement (fileGrp is recursively defined to enable the creation of the hierarchy). Any file group may contain zero or more file elements. File elements in turn can contain one or more FLocat elements (a pointer to a file containing content for this object) and/or a FContent element (the contents of the file, in either XML or Base64 encoding).</xsd:documentation>
    </xsd:annotation>
  - <xsd:choice>
      <xsd:element name="fileGrp" type="fileGrpType" minOccurs="0" maxOccurs="unbounded" />
    - <xsd:element name="file" minOccurs="0" maxOccurs="unbounded" type="fileType">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">The file element <file> provides access to the content files for the digital object being described by the METS document. A <file> element may contain one or more <FLocat> elements which provide pointers to a content file and/or a <FContent> element which wraps an encoded version of the file. Embedding files using <FContent> can be a valuable feature for exchanging digital objects between repositories or for archiving versions of digital objects for off-site storage. All <FLocat> and <FContent> elements should identify and/or contain identical copies of a single file. The <file> element is recursive, thus allowing sub-files or component files of a larger file to be listed in the inventory. Alternatively, by using the <stream> element, a smaller component of a file or of a related file can be placed within a <file> element. Finally, by using the <transformFile> element, it is possible to include within a <file> element a different version of a file that has undergone a transformation for some reason, such as format migration.</xsd:documentation>
```

```
      </xsd:annotation>
    </xsd:element>
  </xsd:choice>
- <xsd:attribute name="ID" type="xsd:ID" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="VERSDATE" type="xsd:dateTime" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">VERSDATE (dateTime/O): An optional dateTime attribute specifying the date this version/fileGrp of the digital object was created.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ADMID (IDREF/O): Contains the ID attribute values of the <techMD>, <sourceMD>, <rightsMD> and/or <digiprovMD> elements within the <amdSec> of the METS document applicable to all of the files in a particular file group. For more information on using METS IDREFS and IDREF type attributes for internal linking, see Chapter 4 of the METS Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="USE" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">USE (string/O): A tagging attribute to indicate the intended use of files within this file group (e.g., master, reference, thumbnails for image files). A USE attribute can be expressed at the<fileGrp> level, the <file> level, the <FLocat> level and/or the <FContent> level. A USE attribute value at the <fileGrp> level should pertain to all of the files in the <fileGrp>. A USE attribute at the <file> level should pertain to all copies of the file as represented by subsidiary <FLocat> and/or <FContent> elements. A USE attribute at the <FLocat> or <FContent> level pertains to the particular copy of the file that is either referenced (<FLocat>) or wrapped (<FContent>).</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
- <xsd:complexType name="structMapType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">structMapType: Complex Type for Structural Maps The structural map (structMap) outlines a hierarchical structure for the original object being encoded, using a series of nested div elements.</xsd:documentation>
    </xsd:annotation>
  - <xsd:sequence>
    - <xsd:element name="div" type="divType">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">The structural divisions of the hierarchical organization provided by a <structMap> are represented by division <div> elements, which can be nested to any depth. Each <div> element can represent either an intellectual (logical) division or a physical division. Every <div> node in the structural map hierarchy may be connected (via subsidiary <mptr> or <fptr> elements) to content files which represent that div's portion of the whole document.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  - <xsd:attribute name="ID" type="xsd:ID" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="TYPE" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">TYPE (string/O): Identifies the type of structure represented by the <structMap>. For example, a <structMap> that represented a purely logical or intellectual structure could be assigned a TYPE value of “logical” whereas a <structMap> that represented a purely physical structure could be assigned a TYPE value of “physical”. However, the METS schema neither defines nor requires a common vocabulary for this attribute. A METS profile, however, may well constrain the values for the <structMap> TYPE.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="LABEL" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">LABEL (string/O): Describes the <structMap> to viewers of the METS document. This would be useful primarily where more than one <structMap> is provided for a single object. A descriptive LABEL value, in that case, could clarify to users the purpose of each of the available structMaps.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
- <xsd:complexType name="divType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">divType: Complex Type for Divisions The METS standard represents a document structurally as a series of nested div elements, that is, as a hierarchy (e.g., a book, which is composed of chapters, which are composed of subchapters, which are composed of text). Every div node in the structural map hierarchy may be connected (via subsidiary mptr or fptr elements) to content files which represent that div's portion of the whole document. SPECIAL NOTE REGARDING DIV ATTRIBUTE VALUES: to clarify the differences between the ORDER, ORDERLABEL, and LABEL attributes for the <div> element, imagine a text with 10 roman numbered pages followed by 10 arabic numbered pages. Page iii would have an ORDER of "3", an ORDERLABEL of "iii" and a LABEL of "Page iii", while page 3 would have an ORDER of "13", an ORDERLABEL of "3" and a LABEL of "Page 3".</xsd:documentation>
    </xsd:annotation>
  - <xsd:sequence>
    - <xsd:element name="mptr" minOccurs="0" maxOccurs="unbounded">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">Like the <fptr> element, the METS pointer element <mptr> represents digital content that manifests its parent <div> element. Unlike the <fptr>, which either directly or indirectly points to content represented in the <fileSec> of the parent METS document, the <mptr> element points to content represented by an external METS document. Thus, this element allows multiple discrete and separate METS documents to be organized at a higher level by a separate METS document. For example, METS documents representing the individual issues in
```

```
the series of a journal could be grouped together and organized by a higher level METS document that represents the entire journal series. Each of the <div> elements in the <structMap> of the METS document representing the journal series would point to a METS document representing an issue. It would do so via a child <mptr> element. Thus the <mptr> element gives METS users considerable flexibility in managing the depth of the <structMap> hierarchy of individual METS documents. The <mptr> element points to an external METS document by means of an xlink:href attribute and associated XLink attributes.</xsd:documentation>
    </xsd:annotation>
  - <xsd:complexType>
    - <xsd:attribute name="ID" type="xsd:ID" use="optional">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
      <xsd:attributeGroup ref="LOCATION" />
      <xsd:attributeGroup ref="xlink:simpleLink" />
    - <xsd:attribute name="CONTENTIDS" type="URIs" use="optional">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">CONTENTIDS (URI/O): Content IDs for the content represented by the <mptr> (equivalent to DIDL DII or Digital Item Identifier, a unique external ID).</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
- <xsd:element name="fptr" minOccurs="0" maxOccurs="unbounded">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">The <fptr> or file pointer element represents digital content that manifests its parent <div> element. The content represented by an <fptr> element must consist of integral files or parts of files that are represented by <file> elements in the <fileSec>. Via its FILEID attribute, an <fptr> may point directly to a single integral <file> element that manifests a structural division. However, an <fptr> element may also govern an <area> element, a <par>, or a <seq> which in turn would point to the relevant file or files. A child <area> element can point to part of a <file> that manifests a division, while the <par> and <seq> elements can point to multiple files or parts of files that together manifest a division. More than one <fptr> element can be associated with a <div> element. Typically sibling <fptr> elements represent alternative versions, or manifestations, of the same content</xsd:documentation>
    </xsd:annotation>
  - <xsd:complexType>
    - <xsd:choice>
      - <xsd:element name="par" type="parType" minOccurs="0">
        - <xsd:annotation>
            <xsd:documentation xml:lang="en">The <par> or parallel files element aggregates pointers to files, parts of files, and/or sequences of files or parts of files that must be played or displayed simultaneously to manifest a block of digital content represented by an <fptr> element. This might be the case, for example, with multi-media content, where a still image might have an accompanying audio track that comments on the still image. In this case, a <par> element would aggregate two <area> elements, one of which pointed to the image file and one of which pointed to the audio file that must be played in conjunction with the image. The <area> element associated with the image could be further qualified with SHAPE and COORDS attributes if only a portion of the image file was pertinent and the <area> element associated with the audio file could be further qualified with BETYPE, BEGIN, EXTTYPE, and EXTENT attributes if only a portion of the associated audio file should be played in conjunction with the image.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      - <xsd:element name="seq" type="seqType" minOccurs="0">
        - <xsd:annotation>
            <xsd:documentation xml:lang="en">The sequence of files element <seq> aggregates pointers to files, parts of files and/or parallel sets of files or parts of files that must be played or displayed sequentially to manifest a block of digital content. This might be the case, for example, if the parent <div> element represented a logical division, such as a diary entry, that spanned multiple pages of a diary and, hence, multiple page image files. In this case, a <seq> element would aggregate multiple, sequentially arranged <area> elements, each of which pointed to one of the image files that must be presented sequentially to manifest the entire diary entry. If the diary entry started in the middle of a page, then the first <area> element (representing the page on which the diary entry starts) might be further qualified, via its SHAPE and COORDS attributes, to specify the specific, pertinent area of the associated image file.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      - <xsd:element name="area" type="areaType" minOccurs="0">
        - <xsd:annotation>
            <xsd:documentation xml:lang="en">The area element <area> typically points to content consisting of just a portion or area of a file represented by a <file> element in the <fileSec>. In some contexts, however, the <area> element can also point to content represented by an integral file. A single <area> element would appear as the direct child of a <fptr> element when only a portion of a <file>, rather than an integral <file>, manifested the digital content represented by the <fptr>. Multiple <area> elements would appear as the direct children of a <par> element or a <seq> element when multiple files or parts of files manifested the digital content represented by an <fptr> element. When used in the context of a <par> or <seq> element an <area> element can point either to an integral file or to a segment of a file as necessary.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:choice>
    - <xsd:attribute name="ID" type="xsd:ID" use="optional">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    - <xsd:attribute name="FILEID" type="xsd:IDREF" use="required">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">FILEID (IDREF/O): An optional attribute that provides the XML ID identifying the <file> element that links to and/or contains the digital content represented by the <fptr>. A <fptr> element should only have a FILEID attribute value if it does not have a child <area>, <par> or <seq> element. If it has a child element, then the responsibility for pointing to the relevant content falls to this child element or its descendants.</xsd:documentation>
          <xsd:documentation xml:lang="en">Changed use from optional to mandatory, Karin
```

```
            Bredenberg</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      - <xsd:attribute name="CONTENTIDS" type="URIs" use="optional">
        - <xsd:annotation>
            <xsd:documentation xml:lang="en">CONTENTIDS (URI/O): Content IDs for the content represented by the <fptr>
              (equivalent to DIDL DII or Digital Item Identifier, a unique external ID).</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:complexType>
    </xsd:element>
  - <xsd:element name="div" type="divType" minOccurs="0" maxOccurs="unbounded">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">In DIAS there are at least 4 div elements present but it cant be expressed in the
          schema. Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
- <xsd:attribute name="ID" type="xsd:ID" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and
        would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR.
        For more information on using ID attributes for internal and external linking see Chapter 4 of the METS
        Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="ORDER" type="xsd:integer" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ORDER (integer/O): A representation of the div's order among its siblings (e.g., its
        absolute, numeric sequence). For an example, and clarification of the distinction between ORDER and ORDERLABEL, see
        the description of the ORDERLABEL attribute.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="ORDERLABEL" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ORDERLABEL (string/O): A representation of the div's order among its siblings (e.g.,
        "xii"), or of any non-integer native numbering system. It is presumed that this value will still be machine actionable
        (e.g., it would support 'go to page ___' function), and it should not be used as a replacement/substitute for the LABEL
        attribute. To understand the differences between ORDER, ORDERLABEL and LABEL, imagine a text with 10 roman
        numbered pages followed by 10 arabic numbered pages. Page iii would have an ORDER of "3", an ORDERLABEL of "iii"
        and a LABEL of "Page iii", while page 3 would have an ORDER of "13", an ORDERLABEL of "3" and a LABEL of "Page
        3".</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="LABEL" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">LABEL (string/O): An attribute used, for example, to identify a <div> to an end user
        viewing the document. Thus a hierarchical arrangement of the <div> LABEL values could provide a table of contents to
        the digital content represented by a METS document and facilitate the users' navigation of the digital object. Note that a
        <div> LABEL should be specific to its level in the structural map. In the case of a book with chapters, the book <div>
        LABEL should have the book title and the chapter <div>; LABELs should have the individual chapter titles, rather than
        having the chapter <div> LABELs combine both book title and chapter title . For further of the distinction between LABEL
        and ORDERLABEL see the description of the ORDERLABEL attribute.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="DMDID" type="xsd:IDREFS" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">DMDID (IDREFS/O): Contains the ID attribute values identifying the <dmdSec>,
        elements in the METS document that contain or link to descriptive metadata pertaining to the structural division
        represented by the current <div> element. For more information on using METS IDREFS and IDREF type attributes for
        internal linking, see Chapter 4 of the METS Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ADMID (IDREFS/O): Contains the ID attribute values identifying the <rightsMD>,
        <sourceMD>, <techMD> and/or <digiprovMD> elements within the <amdSec> of the METS document that contain or
        link to administrative metadata pertaining to the structural division represented by the <div> element. Typically the
        <div> ADMID attribute would be used to identify the <rightsMD> element or elements that pertain to the <div>, but it
        could be used anytime there was a need to link a <div> with pertinent administrative metadata. For more information on
        using METS IDREFS and IDREF type attributes for internal linking, see Chapter 4 of the METS Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="TYPE" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">TYPE (string/O): An attribute that specifies the type of structural division that the <div>
        element represents. Possible <div> TYPE attribute values include: chapter, article, page, track, segment, section etc.
        METS places no constraints on the possible TYPE values. Suggestions for controlled vocabularies for TYPE may be found
        on the METS website.</xsd:documentation>
      <xsd:documentation xml:lang="en">In DIAS an attribute list added via a restriction list type moved to the list
        type="xsd:string". Karin Bredenberg</xsd:documentation>
    </xsd:annotation>
  - <xsd:simpleType>
    - <xsd:restriction base="xsd:string">
        <xsd:enumeration value="preservationmetadata" />
        <xsd:enumeration value="preservationdata" />
        <xsd:enumeration value="technicalmetadata" />
        <xsd:enumeration value="depotoperation" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
- <xsd:attribute name="CONTENTIDS" type="URIs" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">CONTENTIDS (URI/O): Content IDs for the content represented by the <div> (equivalent
        to DIDL DII or Digital Item Identifier, a unique external ID).</xsd:documentation>
```

```
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute ref="xlink:label">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">xlink:label - an xlink label to be referred to by an smLink element</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
- <xsd:complexType name="parType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">parType: Complex Type for Parallel Files The <par> or parallel files element aggregates pointers to files, parts of files, and/or sequences of files or parts of files that must be played or displayed simultaneously to manifest a block of digital content represented by an <fptr> element.</xsd:documentation>
    </xsd:annotation>
  - <xsd:choice maxOccurs="unbounded">
      <xsd:element name="area" type="areaType" minOccurs="0" />
      <xsd:element name="seq" type="seqType" minOccurs="0" />
    </xsd:choice>
  - <xsd:attribute name="ID" type="xsd:ID" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
- <xsd:complexType name="seqType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">seqType: Complex Type for Sequences of Files The seq element should be used to link a div to a set of content files when those files should be played/displayed sequentially to deliver content to a user. Individual <area> subelements within the seq element provide the links to the files or portions thereof.</xsd:documentation>
    </xsd:annotation>
  - <xsd:choice maxOccurs="unbounded">
      <xsd:element name="area" type="areaType" minOccurs="0" />
      <xsd:element name="par" type="parType" minOccurs="0" />
    </xsd:choice>
  - <xsd:attribute name="ID" type="xsd:ID" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
- <xsd:complexType name="areaType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">areaType: Complex Type for Area Linking The area element provides for more sophisticated linking between a div element and content files representing that div, be they text, image, audio, or video files. An area element can link a div to a point within a file, to a one-dimension segment of a file (e.g., text segment, image line, audio/video clip), or a two-dimensional section of a file (e.g, subsection of an image, or a subsection of the video display of a video file. The area element has no content; all information is recorded within its various attributes.</xsd:documentation>
    </xsd:annotation>
  - <xsd:attribute name="ID" type="xsd:ID" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="FILEID" type="xsd:IDREF" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">FILEID (IDREF/R): An attribute which provides the XML ID value that identifies the <file> element in the <fileSec> that then points to and/or contains the digital content represented by the <area> element. It must contain an ID value represented in an ID attribute associated with a <file> element in the <fileSec> element in the same METS document.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="SHAPE" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">SHAPE (string/O): An attribute that can be used as in HTML to define the shape of the relevant area within the content file pointed to by the <area> element. Typically this would be used with image content (still image or video frame) when only a portion of an integal image map pertains. If SHAPE is specified then COORDS must also be present. SHAPE should be used in conjunction with COORDS in the manner defined for the shape and coords attributes on an HTML4 <area> element. SHAPE must contain one of the following values: RECT CIRCLE POLY</xsd:documentation>
      </xsd:annotation>
    - <xsd:simpleType>
      - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="RECT" />
          <xsd:enumeration value="CIRCLE" />
          <xsd:enumeration value="POLY" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  - <xsd:attribute name="COORDS" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">COORDS (string/O): Specifies the coordinates in an image map for the shape of the pertinent area as specified in the SHAPE attribute. While technically optional, SHAPE and COORDS must both appear together to define the relevant area of image content. COORDS should be used in conjunction with SHAPE in the manner defined for the COORDs and SHAPE attributes on an HTML4 <area> element. COORDS must be a comma delimited string of integer value pairs representing coordinates (plus radius in the case of CIRCLE) within an image map. Number of coordinates pairs depends on shape: RECT: x1, y1, x2, y2; CIRC: x1, y1; POLY: x1, y1, x2, y2, x3, y3 . .
```

```
    .</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="BEGIN" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">BEGIN (string/O): An attribute that specifies the point in the content file where the
        relevant section of content begins. It can be used in conjunction with either the END attribute or the EXTENT attribute as
        a means of defining the relevant portion of the referenced file precisely. It can only be interpreted meaningfully in
        conjunction with the BETYPE or EXTTYPE, which specify the kind of beginning/ending point values or beginning/extent
        values that are being used. The BEGIN attribute can be used with or without a companion END or EXTENT element. In
        this case, the end of the content file is assumed to be the end point.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="END" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">END (string/O): An attribute that specifies the point in the content file where the
        relevant section of content ends. It can only be interpreted meaningfully in conjunction with the BETYPE, which specifies
        the kind of ending point values being used. Typically the END attribute would only appear in conjunction with a BEGIN
        element.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="BETYPE" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">BETYPE: Begin/End Type. BETYPE (string/O): An attribute that specifies the kind of
        BEGIN and/or END values that are being used. For example, if BYTE is specified, then the BEGIN and END point values
        represent the byte offsets into a file. If IDREF is specified, then the BEGIN element specifies the ID value that identifies
        the element in a structured text file where the relevant section of the file begins; and the END value (if present) would
        specify the ID value that identifies the element with which the relevant section of the file ends. Must be one of the
        following values: BYTE IDREF SMIL MIDI SMPTE-25 SMPTE-24 SMPTE-DF30 SMPTE-NDF30 SMPTE-DF29.97 SMPTE-
        NDF29.97 TIME TCF XPTR</xsd:documentation>
    </xsd:annotation>
  - <xsd:simpleType>
    - <xsd:restriction base="xsd:string">
        <xsd:enumeration value="BYTE" />
        <xsd:enumeration value="IDREF" />
        <xsd:enumeration value="SMIL" />
        <xsd:enumeration value="MIDI" />
        <xsd:enumeration value="SMPTE-25" />
        <xsd:enumeration value="SMPTE-24" />
        <xsd:enumeration value="SMPTE-DF30" />
        <xsd:enumeration value="SMPTE-NDF30" />
        <xsd:enumeration value="SMPTE-DF29.97" />
        <xsd:enumeration value="SMPTE-NDF29.97" />
        <xsd:enumeration value="TIME" />
        <xsd:enumeration value="TCF" />
        <xsd:enumeration value="XPTR" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
- <xsd:attribute name="EXTENT" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">EXTENT (string/O): An attribute that specifies the extent of the relevant section of the
        content file. Can only be interpreted meaningfully in conjunction with the EXTTYPE which specifies the kind of value that
        is being used. Typically the EXTENT attribute would only appear in conjunction with a BEGIN element and would not be
        used if the BEGIN point represents an IDREF.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="EXTTYPE" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">EXTTYPE (string/O): An attribute that specifies the kind of EXTENT values that are being
        used. For example if BYTE is specified then EXTENT would represent a byte count. If TIME is specified the EXTENT would
        represent a duration of time. EXTTYPE must be one of the following values: BYTE SMIL MIDI SMPTE-25 SMPTE-24 SMPTE-
        DF30 SMPTE-NDF30 SMPTE-DF29.97 SMPTE-NDF29.97 TIME TCF.</xsd:documentation>
    </xsd:annotation>
  - <xsd:simpleType>
    - <xsd:restriction base="xsd:string">
        <xsd:enumeration value="BYTE" />
        <xsd:enumeration value="SMIL" />
        <xsd:enumeration value="MIDI" />
        <xsd:enumeration value="SMPTE-25" />
        <xsd:enumeration value="SMPTE-24" />
        <xsd:enumeration value="SMPTE-DF30" />
        <xsd:enumeration value="SMPTE-NDF30" />
        <xsd:enumeration value="SMPTE-DF29.97" />
        <xsd:enumeration value="SMPTE-NDF29.97" />
        <xsd:enumeration value="TIME" />
        <xsd:enumeration value="TCF" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
- <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ADMID (IDREFS/O): Contains the ID attribute values identifying the <rightsMD>,
        <sourceMD>, <techMD> and/or <digiprovMD> elements within the <amdSec> of the METS document that contain or
        link to administrative metadata pertaining to the content represented by the <area> element. Typically the <area>
        ADMID attribute would be used to identify the <rightsMD> element or elements that pertain to the <area>, but it could
        be used anytime there was a need to link an <area> with pertinent administrative metadata. For more information on
        using METS IDREFS and IDREF type attributes for internal linking, see Chapter 4 of the METS Primer</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="CONTENTIDS" type="URIs" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">CONTENTIDS (URI/O): Content IDs for the content represented by the <area>
        (equivalent to DIDL DII or Digital Item Identifier, a unique external ID).</xsd:documentation>
    </xsd:annotation>
```

```
    </xsd:attribute>
  </xsd:complexType>
- <xsd:complexType name="structLinkType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">structLinkType: Complex Type for Structural Map Linking The Structural Map Linking section
        allows for the specification of hyperlinks between different components of a METS structure delineated in a structural map.
        structLink contains a single, repeatable element, smLink. Each smLink element indicates a hyperlink between two nodes in
        the structMap. The structMap nodes recorded in smLink are identified using their XML ID attribute
        values.</xsd:documentation>
    </xsd:annotation>
  - <xsd:choice maxOccurs="unbounded">
    - <xsd:element name="smLink">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">The Structural Map Link element <smLink> identifies a hyperlink between two nodes
            in the structural map. You would use <smLink>, for instance, to note the existence of hypertext links between web
            pages, if you wished to record those links within METS. NOTE: <smLink> is an empty element. The location of the
            <smLink> element to which the <smLink> element is pointing MUST be stored in the xlink:href
            attribute.</xsd:documentation>
        </xsd:annotation>
      - <xsd:complexType>
        - <xsd:attribute name="ID" type="xsd:ID" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS
                document, and would allow the element to be referenced unambiguously from another element or document
                via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see
                Chapter 4 of the METS Primer.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute ref="xlink:arcrole" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">xlink:arcrole - the role of the link, as per the xlink specification. See
                http://www.w3.org/TR/xlink/</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute ref="xlink:title" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">xlink:title - a title for the link (if needed), as per the xlink specification. See
                http://www.w3.org/TR/xlink/</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute ref="xlink:show" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">xlink:show - see the xlink specification at
                http://www.w3.org/TR/xlink/</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute ref="xlink:actuate" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">xlink:actuate - see the xlink specification at
                http://www.w3.org/TR/xlink/</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute ref="xlink:to" use="required">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">xlink:to - the value of the label for the element in the structMap you are
                linking to.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute ref="xlink:from" use="required">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">xlink:from - the value of the label for the element in the structMap you are
                linking from.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        </xsd:complexType>
      </xsd:element>
    - <xsd:element name="smLinkGrp">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">The structMap link group element <smLinkGrp> provides an implementation of
            xlink:extendLink, and provides xlink compliant mechanisms for establishing xlink:arcLink type links between 2 or
            more <div> elements in <structMap> element(s) occurring within the same METS document or different METS
            documents. The smLinkGrp could be used as an alternative to the <smLink> element to establish a one-to-one link
            between <div> elements in the same METS document in a fully xlink compliant manner. However, it can also be used
            to establish one-to-many or many-to-many links between <div> elements. For example, if a METS document contains
            two <structMap> elements, one of which represents a purely logical structure and one of which represents a purely
            physical structure, the <smLinkGrp> element would provide a means of mapping a <div> representing a logical
            entity (for example, a newspaper article) with multiple <div> elements in the physical <structMap> representing the
            physical areas that together comprise the logical entity (for example, the <div> elements representing the page
            areas that together comprise the newspaper article).</xsd:documentation>
        </xsd:annotation>
      - <xsd:complexType>
        - <xsd:sequence>
          - <xsd:element name="smLocatorLink" minOccurs="2" maxOccurs="unbounded">
            - <xsd:annotation>
                <xsd:documentation xml:lang="en">The structMap locator link element <smLocatorLink> is of xlink:type
                  "locator". It provides a means of identifying a <div> element that will participate in one or more of the
                  links specified by means of <smArcLink> elements within the same <smLinkGrp>. The participating <div>
                  element that is represented by the <smLocatorLink> is identified by means of a URI in the associate
                  xlink:href attribute. The lowest level of this xlink:href URI value should be a fragment identifier that
                  references the ID value that identifies the relevant <div> element. For example, "xlink:href='#div20'"
                  where "div20" is the ID value that identifies the pertinent <div> in the current METS document. Although
                  not required by the xlink specification, an <smLocatorLink> element will typically include an xlink:label
                  attribute in this context, as the <smArcLink> elements will reference these labels to establish the from and
                  to sides of each arc link.</xsd:documentation>
              </xsd:annotation>
            - <xsd:complexType>
```

```
- <xsd:attribute name="ID" type="xsd:ID">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the
        METS document, and would allow the element to be referenced unambiguously from another
        element or document via an IDREF or an XPTR. For more information on using ID attributes for
        internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attributeGroup ref="xlink:locatorLink" />
 </xsd:complexType>
</xsd:element>
- <xsd:element name="smArcLink" minOccurs="1" maxOccurs="unbounded">
 - <xsd:complexType>
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">The structMap arc link element <smArcLink> is of xlink:type "arc" It
        can be used to establish a traversal link between two <div> elements as identified by <smLocatorLink>
        elements within the same smLinkGrp element. The associated xlink:from and xlink:to attributes identify
        the from and to sides of the arc link by referencing the xlink:label attribute values on the participating
        smLocatorLink elements.</xsd:documentation>
    </xsd:annotation>
  - <xsd:attribute name="ID" type="xsd:ID">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the
          METS document, and would allow the element to be referenced unambiguously from another
          element or document via an IDREF or an XPTR. For more information on using ID attributes for
          internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attributeGroup ref="xlink:arcLink" />
  - <xsd:attribute name="ARCTYPE" type="xsd:string">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ARCTYPE (string/O):The ARCTYPE attribute provides a means of
          specifying the relationship between the <div> elements participating in the arc link, and hence the
          purpose or role of the link. While it can be considered analogous to the xlink:arcrole attribute, its
          type is a simple string, rather than anyURI. ARCTYPE has no xlink specified meaning, and the
          xlink:arcrole attribute should be used instead of or in addition to the ARCTYPE attribute when full
          xlink compliance is desired with respect to specifying the role or purpose of the arc
          link.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ADMID (IDREFS/O): Contains the ID attribute values identifying the
          <sourceMD>, <techMD>, <digiprovMD> and/or <rightsMD> elements within the <amdSec> of the
          METS document that contain or link to administrative metadata pertaining to <smArcLink>.
          Typically the <smArcLink> ADMID attribute would be used to identify one or more <sourceMD>
          and/or <techMD> elements that refine or clarify the relationship between the xlink:from and
          xlink:to sides of the arc. For more information on using METS IDREFS and IDREF type attributes for
          internal linking, see Chapter 4 of the METS Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
 </xsd:element>
</xsd:sequence>
<xsd:attribute name="ID" type="xsd:ID" />
- <xsd:attribute name="ARCLINKORDER" default="unordered">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ARCLINKORDER (enumerated string/O): ARCLINKORDER is used to indicate
        whether the order of the smArcLink elements aggregated by the smLinkGrp element is significant. If the order
        is significant, then a value of "ordered" should be supplied. Value defaults to "unordered" Note that the
        ARLINKORDER attribute has no xlink specified meaning.</xsd:documentation>
    </xsd:annotation>
  - <xsd:simpleType>
    - <xsd:restriction base="xsd:string">
        <xsd:enumeration value="ordered" />
        <xsd:enumeration value="unordered" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attributeGroup ref="xlink:extendedLink" />
 </xsd:complexType>
</xsd:element>
</xsd:choice>
- <xsd:attribute name="ID" type="xsd:ID" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and
        would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR.
        For more information on using ID attributes for internal and external linking see Chapter 4 of the METS
        Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
- <xsd:complexType name="behaviorSecType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">behaviorSecType: Complex Type for Behavior Sections Behaviors are executable code which
        can be associated with parts of a METS object. The behaviorSec element is used to group individual behaviors within a
        hierarchical structure. Such grouping can be useful to organize families of behaviors together or to indicate other
        relationships between particular behaviors.</xsd:documentation>
    </xsd:annotation>
  - <xsd:sequence>
      <xsd:element name="behaviorSec" type="behaviorSecType" minOccurs="0" maxOccurs="unbounded" />
    - <xsd:element name="behavior" type="behaviorType" minOccurs="0" maxOccurs="unbounded">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">A behavior element <behavior> can be used to associate executable behaviors with
            content in the METS document. This element has an interface definition <interfaceDef> element that represents an
```

```
            abstract definition of a set of behaviors represented by a particular behavior. A <behavior> element also has a behavior
            mechanism <mechanism> element, a module of executable code that implements and runs the behavior defined
            abstractly by the interface definition.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  - <xsd:attribute name="ID" type="xsd:ID" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and
          would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR.
          For more information on using ID attributes for internal and external linking see Chapter 4 of the METS
          Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="CREATED" type="xsd:dateTime" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">CREATED (dateTime/O): Specifies the date and time of creation for the
          <behaviorSec></xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="LABEL" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">LABEL (string/O): A text description of the behavior section.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
- <xsd:complexType name="behaviorType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">behaviorType: Complex Type for Behaviors A behavior can be used to associate executable
        behaviors with content in the METS object. A behavior element has an interface definition element that represents an
        abstract definition of the set of behaviors represented by a particular behavior. A behavior element also has an behavior
        mechanism which is a module of executable code that implements and runs the behavior defined abstractly by the interface
        definition.</xsd:documentation>
    </xsd:annotation>
  - <xsd:sequence>
    - <xsd:element name="interfaceDef" type="objectType" minOccurs="0">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">The interface definition <interfaceDef> element contains a pointer to an abstract
            definition of a single behavior or a set of related behaviors that are associated with the content of a METS object. The
            interface definition object to which the <interfaceDef> element points using xlink:href could be another digital
            object, or some other entity, such as a text file which describes the interface or a Web Services Description Language
            (WSDL) file. Ideally, an interface definition object contains metadata that describes a set of behaviors or methods. It
            may also contain files that describe the intended usage of the behaviors, and possibly files that represent different
            expressions of the interface definition.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    - <xsd:element name="mechanism" type="objectType">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">A mechanism element <mechanism> contains a pointer to an executable code
            module that implements a set of behaviors defined by an interface definition. The <mechanism> element will be a
            pointer to another object (a mechanism object). A mechanism object could be another METS object, or some other
            entity (e.g., a WSDL file). A mechanism object should contain executable code, pointers to executable code, or
            specifications for binding to network services (e.g., web services).</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  - <xsd:attribute name="ID" type="xsd:ID" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and
          would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. In
          the case of a <behavior> element that applies to a <transformFile> element, the ID value must be present and would be
          referenced from the transformFile/@TRANSFORMBEHAVIOR attribute. For more information on using ID attributes for
          internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="STRUCTID" type="xsd:IDREFS" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">STRUCTID (IDREFS/O): An XML IDREFS attribute used to link a <behavior> to one or
          more <div> elements within a <structMap> in the METS document. The content to which the STRUCTID points is
          considered input to the executable behavior mechanism defined for the behavior. If the <behavior> applies to one or
          more <div> elements, then the STRUCTID attribute must be present.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="BTYPE" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">BTYPE (string/O): The behavior type provides a means of categorizing the related
          behavior.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="CREATED" type="xsd:dateTime" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">CREATED (dateTime/O): The dateTime of creation for the behavior.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="LABEL" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">LABEL (string/O): A text description of the behavior.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="GROUPID" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">GROUPID (string/O): An identifier that establishes a correspondence between the given
          behavior and other behaviors, typically used to facilitate versions of behaviors.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
```

```
  - <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ADMID (IDREFS/O): An optional attribute listing the XML ID values of administrative
          metadata sections within the METS document pertaining to this behavior.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
- <xsd:complexType name="objectType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">objectType: complexType for interfaceDef and mechanism elements The mechanism and
        behavior elements point to external objects--an interface definition object or an executable code object respectively--which
        together constitute a behavior that can be applied to one or more <div> elements in a <structMap>.</xsd:documentation>
    </xsd:annotation>
  - <xsd:attribute name="ID" type="xsd:ID" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and
          would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR.
          For more information on using ID attributes for internal and external linking see Chapter 4 of the METS
          Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="LABEL" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">LABEL (string/O): A text description of the entity represented.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attributeGroup ref="LOCATION" />
    <xsd:attributeGroup ref="xlink:simpleLink" />
  </xsd:complexType>
- <xsd:complexType name="mdSecType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">mdSecType: Complex Type for Metadata Sections A generic framework for pointing
        to/including metadata within a METS document, a la Warwick Framework.</xsd:documentation>
    </xsd:annotation>
  - <xsd:all>
    - <xsd:element name="mdRef" minOccurs="0">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">The metadata reference element <mdRef> element is a generic element used
            throughout the METS schema to provide a pointer to metadata which resides outside the METS document. NB:
            <mdRef> is an empty element. The location of the metadata must be recorded in the xlink:href attribute,
            supplemented by the XPTR attribute as needed.</xsd:documentation>
        </xsd:annotation>
      - <xsd:complexType>
        - <xsd:attribute name="ID" type="xsd:ID" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS
                document, and would allow the element to be referenced unambiguously from another element or document
                via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see
                Chapter 4 of the METS Primer.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
          <xsd:attributeGroup ref="LOCATION" />
          <xsd:attributeGroup ref="xlink:simpleLink" />
          <xsd:attributeGroup ref="METADATA" />
          <xsd:attributeGroup ref="FILECORE" />
        - <xsd:attribute name="LABEL" type="xsd:string" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">LABEL (string/O): Provides a label to display to the viewer of the METS
                document that identifies the associated metadata.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute name="XPTR" type="xsd:string" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">XPTR (string/O): Locates the point within a file to which the <mdRef>
                element refers, if applicable.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        </xsd:complexType>
      </xsd:element>
    - <xsd:element name="mdWrap" minOccurs="0">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">A metadata wrapper element <mdWrap> provides a wrapper around metadata
            embedded within a METS document. The element is repeatable. Such metadata can be in one of two forms: 1) XML-
            encoded metadata, with the XML-encoding identifying itself as belonging to a namespace other than the METS
            document namespace. 2) Any arbitrary binary or textual form, PROVIDED that the metadata is Base64 encoded and
            wrapped in a <binData> element within the internal descriptive metadata element.</xsd:documentation>
        </xsd:annotation>
      - <xsd:complexType>
        - <xsd:choice>
          - <xsd:element name="binData" type="xsd:base64Binary" minOccurs="0">
            - <xsd:annotation>
                <xsd:documentation xml:lang="en">The binary data wrapper element <binData> is used to contain Base64
                  encoded metadata.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
          - <xsd:element name="xmlData" minOccurs="0">
            - <xsd:annotation>
                <xsd:documentation xml:lang="en">The xml data wrapper element <xmlData> is used to contain XML encoded
                  metadata. The content of an <xmlData> element can be in any namespace or in no namespace. As
                  permitted by the XML Schema Standard, the processContents attribute value for the metadata in an
                  <xmlData> is set to "lax". Therefore, if the source schema and its location are identified by means of an
                  XML schemaLocation attribute, then an XML processor will validate the elements for which it can find
                  declarations. If a source schema is not identified, or cannot be found at the specified schemaLocation, then
                  an XML validator will check for well-formedness, but otherwise skip over the elements appearing in the
                  <xmlData> element.</xsd:documentation>
```

```
            </xsd:annotation>
          - <xsd:complexType>
            - <xsd:sequence>
                <xsd:any namespace="##any" maxOccurs="unbounded" processContents="lax" />
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:choice>
      - <xsd:attribute name="ID" type="xsd:ID" use="optional">
        - <xsd:annotation>
            <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS
              document, and would allow the element to be referenced unambiguously from another element or document
              via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see
              Chapter 4 of the METS Primer.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
        <xsd:attributeGroup ref="METADATA" />
        <xsd:attributeGroup ref="FILECORE_MDWRAP" />
      - <xsd:attribute name="LABEL" type="xsd:string" use="optional">
        - <xsd:annotation>
            <xsd:documentation xml:lang="en">LABEL: an optional string attribute providing a label to display to the viewer of
              the METS document identifying the metadata.</xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:complexType>
    </xsd:element>
  </xsd:all>
- <xsd:attribute name="ID" type="xsd:ID" use="required">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ID (ID/R): This attribute uniquely identifies the element within the METS document, and
        would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR.
        The ID attribute on the <dmdSec>, <techMD>, <sourceMD>, <rightsMD> and <digiprovMD> elements (which are all of
        mdSecType) is required, and its value should be referenced from one or more DMDID attributes (when the ID identifies a
        <dmdSec> element) or ADMID attributes (when the ID identifies a <techMD>, <sourceMD>, <rightsMD> or
        <digiprovMD> element) that are associated with other elements in the METS document. The following elements support
        references to a <dmdSec> via a DMDID attribute: <file>, <stream>, <div>. The following elements support references to
        <techMD>, <sourceMD>, <rightsMD> and <digiprovMD> elements via an ADMID attribute: <metsHdr>, <dmdSec>,
        <techMD>, <sourceMD>, <rightsMD>, <digiprovMD>, <fileGrp>, <file>, <stream>, <div>, <area>, <behavior>. For
        more information on using ID attributes for internal and external linking see Chapter 4 of the METS
        Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="GROUPID" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">GROUPID (string/O): This identifier is used to indicate that different metadata sections
        may be considered as part of a group. Two metadata sections with the same GROUPID value are to be considered part of
        the same group. For example this facility might be used to group changed versions of the same metadata if previous
        versions are maintained in a file for tracking purposes.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ADMID (IDREFS/O): Contains the ID attribute values of the <digiprovMD>, <techMD>,
        <sourceMD> and/or <rightsMD> elements within the <amdSec> of the METS document that contain administrative
        metadata pertaining to the current mdSecType element. Typically used in this context to reference preservation
        metadata (digiprovMD) which applies to the current metadata. For more information on using METS IDREFS and IDREF
        type attributes for internal linking, see Chapter 4 of the METS Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="CREATED" type="xsd:dateTime" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">CREATED (dateTime/O): Specifies the date and time of creation for the
        metadata.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="STATUS" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">STATUS (string/O): Indicates the status of this metadata (e.g., superseded, current,
        etc.).</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
- <xsd:complexType name="fileType">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">fileType: Complex Type for Files The file element provides access to content files for a METS
        object. A file element may contain one or more FLocat elements, which provide pointers to a content file, and/or an FContent
        element, which wraps an encoded version of the file. Note that ALL FLocat and FContent elements underneath a single file
        element should identify/contain identical copies of a single file.</xsd:documentation>
    </xsd:annotation>
  - <xsd:sequence>
    - <xsd:element name="FLocat">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">The file location element <FLocat> provides a pointer to the location of a content file.
            It uses the XLink reference syntax to provide linking information indicating the actual location of the content file,
            along with other attributes specifying additional linking information. NOTE: <FLocat> is an empty element. The
            location of the resource pointed to MUST be stored in the xlink:href attribute.</xsd:documentation>
          <xsd:documentation xml:lang="en">Changed use from optional to mandatory, Karin Bredenberg minOccurs="0"
            maxOccurs="unbounded"</xsd:documentation>
        </xsd:annotation>
      - <xsd:complexType>
        - <xsd:attribute name="ID" type="xsd:ID" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS
                document, and would allow the element to be referenced unambiguously from another element or document
                via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see
```

```
        Chapter 4 of the METS Primer.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attributeGroup ref="LOCATION" />
  - <xsd:attribute name="USE" type="xsd:string" use="optional">
    - <xsd:annotation>
      - <xsd:documentation xml:lang="en">USE (string/O): A tagging attribute to indicate the intended use of the specific
        copy of the file represented by the <FLocat> element (e.g., service master, archive master). A USE attribute
        can be expressed at the<fileGrp> level, the <file> level, the <FLocat> level and/or the <FContent> level. A
        USE attribute value at the <fileGrp> level should pertain to all of the files in the <fileGrp>. A USE attribute at
        the <file> level should pertain to all copies of the file as represented by subsidiary <FLocat> and/or
        <FContent> elements. A USE attribute at the <FLocat> or <FContent> level pertains to the particular copy of
        the file that is either referenced (<FLocat>) or wrapped (<FContent>).</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attributeGroup ref="xlink:simpleLink" />
  </xsd:complexType>
</xsd:element>
- <xsd:element name="FContent" minOccurs="0">
  - <xsd:annotation>
    - <xsd:documentation xml:lang="en">The file content element <FContent> is used to identify a content file contained
      internally within a METS document. The content file must be either Base64 encoded and contained within the
      subsidiary <binData> wrapper element, or consist of XML information and be contained within the subsidiary
      <xmlData> wrapper element.</xsd:documentation>
    </xsd:annotation>
  - <xsd:complexType>
    - <xsd:choice>
      - <xsd:element name="binData" type="xsd:base64Binary" minOccurs="0">
        - <xsd:annotation>
          - <xsd:documentation xml:lang="en">A binary data wrapper element <binData> is used to contain a Base64
            encoded file.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      - <xsd:element name="xmlData" minOccurs="0">
        - <xsd:annotation>
          - <xsd:documentation xml:lang="en">An xml data wrapper element <xmlData> is used to contain an XML
            encoded file. The content of an <xmlData> element can be in any namespace or in no namespace. As
            permitted by the XML Schema Standard, the processContents attribute value for the metadata in an
            <xmlData> element is set to "lax". Therefore, if the source schema and its location are identified by means
            of an xsi:schemaLocation attribute, then an XML processor will validate the elements for which it can find
            declarations. If a source schema is not identified, or cannot be found at the specified schemaLocation, then
            an XML validator will check for well-formedness, but otherwise skip over the elements appearing in the
            <xmlData> element.</xsd:documentation>
          </xsd:annotation>
        - <xsd:complexType>
          - <xsd:sequence>
            - <xsd:any namespace="##any" maxOccurs="unbounded" processContents="lax" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    - <xsd:attribute name="ID" type="xsd:ID" use="optional">
      - <xsd:annotation>
        - <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS
          document, and would allow the element to be referenced unambiguously from another element or document
          via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see
          Chapter 4 of the METS Primer.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    - <xsd:attribute name="USE" type="xsd:string" use="optional">
      - <xsd:annotation>
        - <xsd:documentation xml:lang="en">USE (string/O): A tagging attribute to indicate the intended use of the specific
          copy of the file represented by the <FContent> element (e.g., service master, archive master). A USE attribute
          can be expressed at the<fileGrp> level, the <file> level, the <FLocat> level and/or the <FContent> level. A
          USE attribute value at the <fileGrp> level should pertain to all of the files in the <fileGrp>. A USE attribute at
          the <file> level should pertain to all copies of the file as represented by subsidiary <FLocat> and/or
          <FContent> elements. A USE attribute at the <FLocat> or <FContent> level pertains to the particular copy of
          the file that is either referenced (<FLocat>) or wrapped (<FContent>).</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:complexType>
  </xsd:element>
- <xsd:element name="stream" minOccurs="0" maxOccurs="unbounded">
  - <xsd:annotation>
    - <xsd:documentation xml:lang="en">A component byte stream element <stream> may be composed of one or more
      subsidiary streams. An MPEG4 file, for example, might contain separate audio and video streams, each of which is
      associated with technical metadata. The repeatable <stream> element provides a mechanism to record the existence
      of separate data streams within a particular file, and the opportunity to associate <dmdSec> and <amdSec> with
      those subsidiary data streams if desired.</xsd:documentation>
    </xsd:annotation>
  - <xsd:complexType>
    - <xsd:complexContent>
      - <xsd:restriction base="xsd:anyType">
        - <xsd:attribute name="ID" type="xsd:ID" use="optional">
          - <xsd:annotation>
            - <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the
              METS document, and would allow the element to be referenced unambiguously from another element or
              document via an IDREF or an XPTR. For more information on using ID attributes for internal and
              external linking see Chapter 4 of the METS Primer.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute name="streamType" type="xsd:string" use="optional">
          - <xsd:annotation>
            - <xsd:documentation xml:lang="en">streamType (string/O): The IANA MIME media type for the
              bytestream.</xsd:documentation>
            </xsd:annotation>
```

```
      </xsd:attribute>
    - <xsd:attribute name="OWNERID" type="xsd:string" use="optional">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">OWNERID (string/O): Used to provide a unique identifier (which could include a URI) assigned to the file. This identifier may differ from the URI used to retrieve the file.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    - <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">ADMID (IDREFS/O): Contains the ID attribute values of the <techMD>, <sourceMD>, <rightsMD> and/or <digiprovMD> elements within the <amdSec> of the METS document that contain administrative metadata pertaining to the bytestream. For more information on using METS IDREFS and IDREF type attributes for internal linking, see Chapter 4 of the METS Primer.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    - <xsd:attribute name="DMDID" type="xsd:IDREFS" use="optional">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">DMDID (IDREFS/O): Contains the ID attribute values identifying the <dmdSec>, elements in the METS document that contain or link to descriptive metadata pertaining to the content file stream represented by the current <stream> element. For more information on using METS IDREFS and IDREF type attributes for internal linking, see Chapter 4 of the METS Primer.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    - <xsd:attribute name="BEGIN" type="xsd:string" use="optional">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">BEGIN (string/O): An attribute that specifies the point in the parent <file> where the current <stream> begins. It can be used in conjunction with the END attribute as a means of defining the location of the stream within its parent file. However, the BEGIN attribute can be used with or without a companion END attribute. When no END attribute is specified, the end of the parent file is assumed also to be the end point of the stream. The BEGIN and END attributes can only be interpreted meaningfully in conjunction with a BETYPE attribute, which specifies the kind of beginning/ending point values that are being used.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    - <xsd:attribute name="END" type="xsd:string" use="optional">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">END (string/O): An attribute that specifies the point in the parent <file> where the <stream> ends. It can only be interpreted meaningfully in conjunction with the BETYPE, which specifies the kind of ending point values being used. Typically the END attribute would only appear in conjunction with a BEGIN attribute.</xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    - <xsd:attribute name="BETYPE" use="optional">
      - <xsd:annotation>
          <xsd:documentation xml:lang="en">BETYPE: Begin/End Type. BETYPE (string/O): An attribute that specifies the kind of BEGIN and/or END values that are being used. Currently BYTE is the only valid value that can be used in conjunction with nested <file> or <stream> elements.</xsd:documentation>
        </xsd:annotation>
      - <xsd:simpleType>
        - <xsd:restriction base="xsd:string">
            <xsd:enumeration value="BYTE" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
</xsd:element>
- <xsd:element name="transformFile" minOccurs="0" maxOccurs="unbounded">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">The transform file element <transformFile> provides a means to access any subsidiary files listed below a <file> element by indicating the steps required to "unpack" or transform the subsidiary files. This element is repeatable and might provide a link to a <behavior> in the <behaviorSec> that performs the transformation.</xsd:documentation>
    </xsd:annotation>
  - <xsd:complexType>
    - <xsd:complexContent>
      - <xsd:restriction base="xsd:anyType">
        - <xsd:attribute name="ID" type="xsd:ID" use="optional">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">ID (ID/O): This attribute uniquely identifies the element within the METS document, and would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR. For more information on using ID attributes for internal and external linking see Chapter 4 of the METS Primer.</xsd:documentation>
            </xsd:annotation>
          </xsd:attribute>
        - <xsd:attribute name="TRANSFORMTYPE" use="required">
          - <xsd:annotation>
              <xsd:documentation xml:lang="en">TRANSFORMTYPE (string/R): Is used to indicate the type of transformation needed to render content of a file accessible. This may include unpacking a file into subsidiary files/streams. The controlled value constraints for this XML string include "decompression" and "decryption". Decompression is defined as the action of reversing data compression, i.e., the process of encoding information using fewer bits than an unencoded representation would use by means of specific encoding schemas. Decryption is defined as the process of restoring data that has been obscured to make it unreadable without special knowledge (encrypted data) to its original form.</xsd:documentation>
            </xsd:annotation>
          - <xsd:simpleType>
            - <xsd:restriction base="xsd:string">
                <xsd:enumeration value="decompression" />
                <xsd:enumeration value="decryption" />
              </xsd:restriction>
            </xsd:simpleType>
```

```
            </xsd:attribute>
          - <xsd:attribute name="TRANSFORMALGORITHM" type="xsd:string" use="required">
            - <xsd:annotation>
                <xsd:documentation xml:lang="en">TRANSFORM-ALGORITHM (string/R): Specifies the decompression or
                  decryption routine used to access the contents of the file. Algorithms for compression can be either
                  loss-less or lossy.</xsd:documentation>
              </xsd:annotation>
            </xsd:attribute>
          - <xsd:attribute name="TRANSFORMKEY" type="xsd:string" use="optional">
            - <xsd:annotation>
                <xsd:documentation xml:lang="en">TRANSFORMKEY (string/O): A key to be used with the transform
                  algorithm for accessing the file's contents.</xsd:documentation>
              </xsd:annotation>
            </xsd:attribute>
          - <xsd:attribute name="TRANSFORMBEHAVIOR" type="xsd:IDREF" use="optional">
            - <xsd:annotation>
                <xsd:documentation xml:lang="en">TRANSFORMBEHAVIOR (string/O): An IDREF to a behavior element for
                  this transformation.</xsd:documentation>
              </xsd:annotation>
            </xsd:attribute>
          - <xsd:attribute name="TRANSFORMORDER" type="xsd:positiveInteger" use="required">
            - <xsd:annotation>
                <xsd:documentation xml:lang="en">TRANSFORMORDER (postive-integer/R): The order in which the
                  instructions must be followed in order to unpack or transform the container file.</xsd:documentation>
              </xsd:annotation>
            </xsd:attribute>
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="file" type="fileType" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
- <xsd:attribute name="ID" type="xsd:ID" use="required">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ID (ID/R): This attribute uniquely identifies the element within the METS document, and
        would allow the element to be referenced unambiguously from another element or document via an IDREF or an XPTR.
        Typically, the ID attribute value on a <file> element would be referenced from one or more FILEID attributes (which are
        of type IDREF) on <fptr>and/or <area> elements within the <structMap>. Such references establish links between
        structural divisions (<div> elements) and the specific content files or parts of content files that manifest them. For more
        information on using ID attributes for internal and external linking see Chapter 4 of the METS
        Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="SEQ" type="xsd:int" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">SEQ (integer/O): Indicates the sequence of this <file> relative to the others in its
        <fileGrp>.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attributeGroup ref="FILECORE" />
- <xsd:attribute name="OWNERID" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">OWNERID (string/O): A unique identifier assigned to the file by its owner. This may be a
        URI which differs from the URI used to retrieve the file.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="ADMID" type="xsd:IDREFS" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">ADMID (IDREFS/O): Contains the ID attribute values of the <techMD>, <sourceMD>,
        <rightsMD> and/or <digiprovMD> elements within the <amdSec> of the METS document that contain administrative
        metadata pertaining to the file. For more information on using METS IDREFS and IDREF type attributes for internal
        linking, see Chapter 4 of the METS Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="DMDID" type="xsd:IDREFS" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">DMDID (IDREFS/O): Contains the ID attribute values identifying the <dmdSec>,
        elements in the METS document that contain or link to descriptive metadata pertaining to the content file represented by
        the current <file> element. For more information on using METS IDREFS and IDREF type attributes for internal linking,
        see Chapter 4 of the METS Primer.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="GROUPID" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">GROUPID (string/O): An identifier that establishes a correspondence between this file
        and files in other file groups. Typically, this will be used to associate a master file in one file group with the derivative
        files made from it in other file groups.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="USE" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">USE (string/O): A tagging attribute to indicate the intended use of all copies of the file
        aggregated by the <file> element (e.g., master, reference, thumbnails for image files). A USE attribute can be expressed
        at the<fileGrp> level, the <file> level, the <FLocat> level and/or the <FContent> level. A USE attribute value at the
        <fileGrp> level should pertain to all of the files in the <fileGrp>. A USE attribute at the <file> level should pertain to all
        copies of the file as represented by subsidiary <FLocat> and/or <FContent> elements. A USE attribute at the <FLocat>
        or <FContent> level pertains to the particular copy of the file that is either referenced (<FLocat>) or wrapped
        (<FContent>).</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="BEGIN" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">BEGIN (string/O): An attribute that specifies the point in the parent <file> where the
        current <file> begins. When used in conjunction with a <file> element, this attribute is only meaningful when this
        element is nested, and its parent <file> element represents a container file. It can be used in conjunction with the END
```

```
      attribute as a means of defining the location of the current file within its parent file. However, the BEGIN attribute can be
      used with or without a companion END attribute. When no END attribute is specified, the end of the parent file is
      assumed also to be the end point of the current file. The BEGIN and END attributes can only be interpreted meaningfully
      in conjunction with a BETYPE attribute, which specifies the kind of beginning/ending point values that are being
      used.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="END" type="xsd:string" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">END (string/O): An attribute that specifies the point in the parent <file> where the
      current, nested <file> ends. It can only be interpreted meaningfully in conjunction with the BETYPE, which specifies the
      kind of ending point values being used. Typically the END attribute would only appear in conjunction with a BEGIN
      attribute.</xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
- <xsd:attribute name="BETYPE" use="optional">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">BETYPE: Begin/End Type. BETYPE (string/O): An attribute that specifies the kind of
      BEGIN and/or END values that are being used. Currently BYTE is the only valid value that can be used in conjunction with
      nested <file> or <stream> elements.</xsd:documentation>
    </xsd:annotation>
  - <xsd:simpleType>
    - <xsd:restriction base="xsd:string">
        <xsd:enumeration value="BYTE" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>
- <xsd:simpleType name="URIs">
    <xsd:list itemType="xsd:anyURI" />
</xsd:simpleType>
- <xsd:attributeGroup name="METADATA">
  - <xsd:attribute name="MDTYPE" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">MDTYPE (string/R): Is used to indicate the type of the associated metadata. It must
        have one of the following values: MARC: any form of MARC record MODS: metadata in the Library of Congress MODS
        format EAD: Encoded Archival Description finding aid DC: Dublin Core NISOIMG: NISO Technical Metadata for Digital Still
        Images LC-AV: technical metadata specified in the Library of Congress A/V prototyping project VRA: Visual Resources
        Association Core TEIHDR: Text Encoding Initiative Header DDI: Data Documentation Initiative FGDC: Federal Geographic
        Data Committee metadata LOM: Learning Object Model PREMIS: PREservation Metadata: Implementation Strategies
        PREMIS:OBJECT: PREMIS Object entiry PREMIS:AGENT: PREMIS Agent entity PREMIS:RIGHTS: PREMIS Rights entity
        PREMIS:EVENT: PREMIS Event entity TEXTMD: textMD Technical metadata for text METSRIGHTS: Rights Declaration
        Schema ISO 19115:2003 NAP: North American Profile of ISO 19115:2003 descriptive metadata OTHER: metadata in a
        format not specified above</xsd:documentation>
      </xsd:annotation>
    - <xsd:simpleType>
      - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="MARC" />
          <xsd:enumeration value="MODS" />
          <xsd:enumeration value="EAD" />
          <xsd:enumeration value="DC" />
          <xsd:enumeration value="NISOIMG" />
          <xsd:enumeration value="LC-AV" />
          <xsd:enumeration value="VRA" />
          <xsd:enumeration value="TEIHDR" />
          <xsd:enumeration value="DDI" />
          <xsd:enumeration value="FGDC" />
          <xsd:enumeration value="LOM" />
          <xsd:enumeration value="PREMIS" />
          <xsd:enumeration value="PREMIS:OBJECT" />
          <xsd:enumeration value="PREMIS:AGENT" />
          <xsd:enumeration value="PREMIS:RIGHTS" />
          <xsd:enumeration value="PREMIS:EVENT" />
          <xsd:enumeration value="TEXTMD" />
          <xsd:enumeration value="METSRIGHTS" />
          <xsd:enumeration value="ISO 19115:2003 NAP" />
          <xsd:enumeration value="OTHER" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  - <xsd:attribute name="OTHERMDTYPE" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">OTHERMDTYPE (string/O): Specifies the form of metadata in use when the value OTHER
        is indicated in the MDTYPE attribute.</xsd:documentation>
        <xsd:documentation xml:lang="en">Removed type, Karin Bredenberg type="xsd:string"</xsd:documentation>
        <xsd:documentation xml:lang="en">Added valuelist, Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    - <xsd:simpleType>
      - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="ADDML" />
          <xsd:enumeration value="EAC-F" />
          <xsd:enumeration value="EAC-CPF" />
          <xsd:enumeration value="EAG" />
          <xsd:enumeration value="METS" />
          <xsd:enumeration value="EAC" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  - <xsd:attribute name="MDTYPEVERSION" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">MDTYPEVERSION(string/O): Provides a means for recording the version of the type of
        metadata (as recorded in the MDTYPE or OTHERMDTYPE attribute) that is being used. This may represent the version of
        the underlying data dictionary or metadata model rather than a schema version.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
```

```
  </xsd:attributeGroup>
- <xsd:attributeGroup name="LOCATION">
  - <xsd:attribute name="LOCTYPE" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">LOCTYPE (string/R): Specifies the locator type used in the xlink:href attribute. Valid
          values for LOCTYPE are: ARK URN URL PURL HANDLE DOI OTHER</xsd:documentation>
        <xsd:documentation xml:lang="en">Only value permitted is URL, Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    - <xsd:simpleType>
      - <xsd:restriction base="xsd:string">
          <!-- <xsd:enumeration value="ARK"/> -->
          <!-- <xsd:enumeration value="URN"/> -->
          <xsd:enumeration value="URL" />
          <!-- <xsd:enumeration value="PURL"/> -->
          <!-- <xsd:enumeration value="HANDLE"/> -->
          <!-- <xsd:enumeration value="DOI"/> -->
          <!-- <xsd:enumeration value="OTHER"/> -->
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  - <xsd:attribute name="OTHERLOCTYPE" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">OTHERLOCTYPE (string/O): Specifies the locator type when the value OTHER is used in
          the LOCTYPE attribute. Although optional, it is strongly recommended when OTHER is used.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:attributeGroup>
- <xsd:attributeGroup name="FILECORE">
  - <xsd:attribute name="MIMETYPE" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">MIMETYPE (string/O): The IANA MIME media type for the associated file or wrapped
          content. Some values for this attribute can be found on the IANA website.</xsd:documentation>
        <xsd:documentation xml:lang="en">Changed use from optional to mandatory, Karin Bredenberg</xsd:documentation>
        <xsd:documentation xml:lang="en">In DIAS an value list is added. Original type type="xsd:string" moved to an restriction.
          Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    - <xsd:simpleType>
      - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="text/txt" />
          <xsd:enumeration value="text/xml" />
          <xsd:enumeration value="image/jpg" />
          <xsd:enumeration value="image/pdf" />
          <xsd:enumeration value="image/tiff" />
          <xsd:enumeration value="audio/mp3" />
          <xsd:enumeration value="video/mpg" />
          <xsd:enumeration value="package/tar" />
          <xsd:enumeration value="application/x-tar" />
          <xsd:enumeration value="application/xml" />
          <xsd:enumeration value="text/plain" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  - <xsd:attribute name="SIZE" type="xsd:long" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">SIZE (long/O): Specifies the size in bytes of the associated file or wrapped
          content.</xsd:documentation>
        <xsd:documentation xml:lang="en">Changed use from optional to mandatory, Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="CREATED" type="xsd:dateTime" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">CREATED (dateTime/O): Specifies the date and time of creation for the associated file or
          wrapped content.</xsd:documentation>
        <xsd:documentation xml:lang="en">Changed use from optional to mandatory, Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="CHECKSUM" type="xsd:string" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">CHECKSUM (string/O): Provides a checksum value for the associated file or wrapped
          content.</xsd:documentation>
        <xsd:documentation xml:lang="en">Changed use from optional to mandatory, Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="CHECKSUMTYPE" use="required">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">CHECKSUMTYPE (enumerated string/O): Specifies the checksum algorithm used to
          produce the value contained in the CHECKSUM attribute. CHECKSUMTYPE must contain one of the following values: Adler-
          32 CRC32 HAVAL MD5 MNP SHA-1 SHA-256 SHA-384 SHA-512 TIGER WHIRLPOOL</xsd:documentation>
        <xsd:documentation xml:lang="en">Changed use from optional to mandatory, Karin Bredenberg</xsd:documentation>
        <xsd:documentation xml:lang="en">Only MD5 and SHA- accepted, Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    - <xsd:simpleType>
      - <xsd:restriction base="xsd:string">
          <!--                                <xsd:enumeration value="Adler-32"/> -->
          <!--                                <xsd:enumeration value="CRC32"/> -->
          <!--                                <xsd:enumeration value="HAVAL"/> -->
          <xsd:enumeration value="MD5" />
          <!--                                <xsd:enumeration value="MNP"/> -->
          <xsd:enumeration value="SHA-1" />
          <xsd:enumeration value="SHA-256" />
          <xsd:enumeration value="SHA-384" />
          <xsd:enumeration value="SHA-512" />
          <!--                                <xsd:enumeration value="TIGER"/> -->
          <!--                                <xsd:enumeration value="WHIRLPOOL"/> -->
```

```
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:attributeGroup>
- <xsd:attributeGroup name="FILECORE_MDWRAP">
  - <xsd:annotation>
      <xsd:documentation xml:lang="en">Added copy to be able to have diffrent mandatory statements in mdwrap and mdref, Karin
        Bredenberg</xsd:documentation>
    </xsd:annotation>
  - <xsd:attribute name="MIMETYPE" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">MIMETYPE (string/O): The IANA MIME media type for the associated file or wrapped
          content. Some values for this attribute can be found on the IANA website.</xsd:documentation>
        <xsd:documentation xml:lang="en">In DIAS an value list is added. Original type type="xsd:string" moved to an restriction.
          Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    - <xsd:simpleType>
      - <xsd:restriction base="xsd:string">
          <xsd:enumeration value="text/txt" />
          <xsd:enumeration value="text/xml" />
          <xsd:enumeration value="image/jpg" />
          <xsd:enumeration value="image/pdf" />
          <xsd:enumeration value="image/tiff" />
          <xsd:enumeration value="audio/mp3" />
          <xsd:enumeration value="video/mpg" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  - <xsd:attribute name="SIZE" type="xsd:long" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">SIZE (long/O): Specifies the size in bytes of the associated file or wrapped
          content.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="CREATED" type="xsd:dateTime" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">CREATED (dateTime/O): Specifies the date and time of creation for the associated file or
          wrapped content.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="CHECKSUM" type="xsd:string" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">CHECKSUM (string/O): Provides a checksum value for the associated file or wrapped
          content.</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  - <xsd:attribute name="CHECKSUMTYPE" use="optional">
    - <xsd:annotation>
        <xsd:documentation xml:lang="en">CHECKSUMTYPE (enumerated string/O): Specifies the checksum algorithm used to
          produce the value contained in the CHECKSUM attribute. CHECKSUMTYPE must contain one of the following values: Adler-
          32 CRC32 HAVAL MD5 MNP SHA-1 SHA-256 SHA-384 SHA-512 TIGER WHIRLPOOL</xsd:documentation>
        <xsd:documentation xml:lang="en">Only MD5 and SHA- accepted, Karin Bredenberg</xsd:documentation>
      </xsd:annotation>
    - <xsd:simpleType>
      - <xsd:restriction base="xsd:string">
          <!--                                <xsd:enumeration value="Adler-32"/> -->
          <!--                                <xsd:enumeration value="CRC32"/> -->
          <!--                                <xsd:enumeration value="HAVAL"/> -->
          <xsd:enumeration value="MD5" />
          <!--                                <xsd:enumeration value="MNP"/> -->
          <xsd:enumeration value="SHA-1" />
          <xsd:enumeration value="SHA-256" />
          <xsd:enumeration value="SHA-384" />
          <xsd:enumeration value="SHA-512" />
          <!--                                <xsd:enumeration value="TIGER"/> -->
          <!--                                <xsd:enumeration value="WHIRLPOOL"/> -->
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:attributeGroup>
</xsd:schema>
```

Vedlegg 4:

DIAS-PREMIS.xsd

PREMIS *(Preservation Metadata: Implementation Strategies)* spesifiserer bevarings-beskrivende metadata i en arkivpakke. Den dokumenterer hendelsene og håndterings-historikken gjennom arkivpakkens livssyklus fra og med mottak i arkivdepot. PREMIS er dokumentert på nettstedet *http://www.loc.gov/standards/premis/*

DIAS-PREMIS er DIAS-prosjektets tilpasning av PREMIS for bruk i en felles arkivpakke-struktur for norske statlige og kommunale arkivdepoter.

DIAS-PREMIS er tilgjengelig som xsd-fil på adressen:
http://schema.arkivverket.no/PREMIS/v2.0/DIAS_PREMIS.xsd

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--

    ********************************************************************************************
                                    PREMIS Preservation Metadata Schema
                                                Version 2.0
                                               July 17, 2008
    ********************************************************************************************

    editor: Ray Denenberg, Library of Congress; via XML Spy. Send comments to rden@loc.gov


    ********************************************************************************************
                                          Changes:

    ************
    May 1:
    ***********
    Under objectCharacteristicsComplexType
                        <xs:element ref="format" minOccurs="1" maxOccurs="1"/>
    was changed to:
                        <xs:element ref="format" minOccurs="1" maxOccurs="unbounded"/>

    i.e. <format> of <objectCharacteristics> was made repeatable.

    This change is for consistency with the data dictionary, version 2.0, i n which <format> was changed from non-repeatable to repeata

    This schema, version 2.0,  is intended to be completely in synch with the data dictionary, version 2.0.s
    ********************************************************************************************
  -->
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns="http://arkivverket.no/standarder/PREMIS" targetNamespace="http://arkivverket.no/standarder/PREMIS"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  - <!--

       Import XLink
    -->
    <xs:import namespace="http://www.w3.org/1999/xlink" schemaLocation="http://www.loc.gov/standards/xlink/xlink.xsd" />
  - <!--


      An instance is
      (1) One or more of <object>, <event>, <agent>, <rights> all wrapped within a <premis> container; or
      (2) any one of  <object>, <event>, <agent>, <rights>  by itself.

      Thus the root element is one of the following:  <premis>, <object>, <event>, <agent>, <rights>

      ********************************************************************************************
      ************************ Root element declarations ****************************************
      ********************************************************************************************
    -->
    <xs:element name="premis" type="premisComplexType" />
    <xs:element name="object" type="objectComplexType" />
    <xs:element name="event" type="eventComplexType" />
    <xs:element name="agent" type="agentComplexType" />
    <xs:element name="rights" type="rightsComplexType" />
  - <!--



      ********************************************************************************************
      ************************ definitions of complex types for the root elements)*****************
      ********************************************************************************************

      ********************************** premisComplexType
    -->
  - <xs:complexType name="premisComplexType">
    - <xs:sequence>
        <xs:element ref="object" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="event" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="agent" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="rights" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="version" type="versionSimpleType" use="required" />
    </xs:complexType>
  - <!--


      ********************************************************************************************
      ************************objectComplexType  (and the three major <object>  category definitions) *********************
      ********************************************************************************************
    -->
    <xs:complexType name="objectComplexType" abstract="true" />
  - <!--
      ***************
      The three "types":  'file', 'representation', and 'bitstream'. These are the values for the xsi:type attribute in an instance.
      For an object of type file: <object xsi:type="file">  will mean that the complexType "file" will be validated.
      For an object of type representation: <object xsi:type="representation">  will mean that the complexType "representation" will be
      For an object of type bitstream: <object xsi:type="bitstream">  will mean that the complexType "bitstream" will be validated.
      *****************

      ******* file
    -->
  - <xs:complexType name="file">
    - <xs:complexContent>
      - <xs:extension base="objectComplexType">
        - <xs:sequence>
            <xs:element ref="objectIdentifier" minOccurs="1" maxOccurs="unbounded" />
            <!--  Data dictionary lists objectCategory here, values:  'file', 'representation', or  'bitstream'. It is
            omitted and instead the mechanism described in the preceding comment is used to signify the category of the object. Using thi
            <xs:element ref="preservationLevel" minOccurs="0" maxOccurs="unbounded" />
```

```
        <xs:element ref="significantProperties" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="objectCharacteristics" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="originalName" minOccurs="0" maxOccurs="1" />
        <xs:element ref="storage" minOccurs="0" maxOccurs="unbounded" />
        <!--  note conflict with data dictionary, where storage is listed erroneously as mandatory. This will be cited as
        errata and corrected in the next version of the data dictionary  -->
        <xs:element ref="environment" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="signatureInformation" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="relationship" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="linkingEventIdentifier" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="linkingIntellectualEntityIdentifier" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="linkingRightsStatementIdentifier" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="xmlID" type="xs:ID" />
      <xs:attribute name="version" type="versionSimpleType" use="optional" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
- <!--
    ********** representation

-->
- <xs:complexType name="representation">
  - <xs:complexContent>
    - <xs:extension base="objectComplexType">
      - <xs:sequence>
        <xs:element ref="objectIdentifier" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="preservationLevel" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="significantProperties" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="originalName" minOccurs="0" maxOccurs="1" />
        <xs:element ref="environment" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="relationship" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="linkingEventIdentifier" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="linkingIntellectualEntityIdentifier" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="linkingRightsStatementIdentifier" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="xmlID" type="xs:ID" />
      <xs:attribute name="version" type="versionSimpleType" use="optional" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
- <!--
    ********** bitstream

-->
- <xs:complexType name="bitstream">
  - <xs:complexContent>
    - <xs:extension base="objectComplexType">
      - <xs:sequence>
        <xs:element ref="objectIdentifier" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="significantProperties" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="objectCharacteristics" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="storage" minOccurs="0" maxOccurs="unbounded" />
        <!--  note conflict with data dictionary, where storage is listed erroneously as mandatory. This will be cited as
        errata and corrected in the next version of the data dictionary  -->
        <xs:element ref="environment" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="signatureInformation" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="relationship" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="linkingEventIdentifier" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="linkingIntellectualEntityIdentifier" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="linkingRightsStatementIdentifier" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="xmlID" type="xs:ID" />
      <xs:attribute name="version" type="versionSimpleType" use="optional" />
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
- <!--


    **********************************eventComplexType

-->
- <xs:complexType name="eventComplexType">
  - <xs:sequence>
    <xs:element ref="eventIdentifier" minOccurs="1" maxOccurs="1" />
    <xs:element ref="eventType" minOccurs="1" maxOccurs="1" />
    <xs:element ref="eventDateTime" minOccurs="1" maxOccurs="1" />
    <xs:element ref="eventDetail" minOccurs="0" maxOccurs="1" />
    <xs:element ref="eventOutcomeInformation" minOccurs="0" maxOccurs="unbounded" />
    <xs:element ref="linkingAgentIdentifier" minOccurs="0" maxOccurs="unbounded" />
    <xs:element ref="linkingObjectIdentifier" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="xmlID" type="xs:ID" />
  <xs:attribute name="version" type="versionSimpleType" use="optional" />
</xs:complexType>
- <!--


    ************************  agentComplexType

-->
- <xs:complexType name="agentComplexType">
  - <xs:sequence>
    <xs:element ref="agentIdentifier" minOccurs="1" maxOccurs="unbounded" />
```

```
      <xs:element ref="agentName" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="agentType" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="xmlID" type="xs:ID" />
    <xs:attribute name="version" type="versionSimpleType" use="optional" />
  </xs:complexType>
- <!--


     ****************** rightsComplexType

  -->
- <xs:complexType name="rightsComplexType">
  - <xs:choice minOccurs="1" maxOccurs="unbounded">
      <xs:element ref="rightsStatement" />
      <xs:element ref="rightsExtension" />
    </xs:choice>
    <xs:attribute name="xmlID" type="xs:ID" />
    <xs:attribute name="version" type="versionSimpleType" use="optional" />
  </xs:complexType>
- <!--
     ********************************************************************************************
     ************************subsidiary  complexType definitions  ********************
     ********************************************************************************************

     ******* agentIdentifierComplexType
  -->
- <xs:complexType name="agentIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="agentIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="agentIdentifierValue" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
  <!-- ***************contentLocationComplexType
    -->
- <xs:complexType name="contentLocationComplexType">
  - <xs:sequence>
      <xs:element ref="contentLocationType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="contentLocationValue" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
  <!-- ******** copyrightInformationComplexType
    -->
- <xs:complexType name="copyrightInformationComplexType">
  - <xs:sequence>
      <xs:element ref="copyrightStatus" minOccurs="1" maxOccurs="1" />
      <xs:element ref="copyrightJurisdiction" minOccurs="1" maxOccurs="1" />
      <xs:element ref="copyrightStatusDeterminationDate" minOccurs="0" maxOccurs="1" />
      <xs:element ref="copyrightNote" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <!-- ***************creatingApplicationComplexType
    -->
- <xs:complexType name="creatingApplicationComplexType">
  - <xs:choice>
    - <xs:sequence>
        <xs:element ref="creatingApplicationName" minOccurs="1" maxOccurs="1" />
        <xs:element ref="creatingApplicationVersion" minOccurs="0" maxOccurs="1" />
        <xs:element ref="dateCreatedByApplication" minOccurs="0" maxOccurs="1" />
        <xs:element ref="creatingApplicationExtension" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
    - <xs:sequence>
        <xs:element ref="creatingApplicationVersion" minOccurs="1" maxOccurs="1" />
        <xs:element ref="dateCreatedByApplication" minOccurs="0" maxOccurs="1" />
        <xs:element ref="creatingApplicationExtension" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
    - <xs:sequence>
        <xs:element ref="dateCreatedByApplication" minOccurs="1" maxOccurs="1" />
        <xs:element ref="creatingApplicationExtension" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:element ref="creatingApplicationExtension" minOccurs="1" maxOccurs="unbounded" />
    </xs:choice>
    <!-- All of the  elements individually are optional, but at least one must occur. And those occuring must occur in the
    specified order.And some are non-repeatable. XML schema doesn't provide an easy way to define such a construct.  If  sequencing an
  </xs:complexType>
- <!--

     ***************dependencyComplexType
     See comment for "creatingApplicationComplexType"
  -->
- <xs:complexType name="dependencyComplexType">
  - <xs:choice>
    - <xs:sequence>
        <xs:element ref="dependencyName" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="dependencyIdentifier" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:element ref="dependencyIdentifier" minOccurs="1" maxOccurs="unbounded" />
    </xs:choice>
  </xs:complexType>
- <!--

     ***************dependencyIdentifierComplexType
  -->
```

```
- <xs:complexType name="dependencyIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="dependencyIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="dependencyIdentifierValue" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
- <!--
     ***************environmentComplexType
    See comment for "creatingApplicationComplexType"
  -->
- <xs:complexType name="environmentComplexType">
  - <xs:choice>
      <!--    -->
    - <xs:sequence>
        <xs:element ref="environmentCharacteristic" minOccurs="1" maxOccurs="1" />
        <xs:element ref="environmentPurpose" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="environmentNote" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="dependency" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="software" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="hardware" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="environmentExtension" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
      <!--    -->
    - <xs:sequence>
        <xs:element ref="environmentPurpose" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="environmentNote" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="dependency" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="software" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="hardware" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="environmentExtension" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
      <!--    -->
    - <xs:sequence>
        <xs:element ref="environmentNote" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="dependency" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="software" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="hardware" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="environmentExtension" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
      <!--    -->
    - <xs:sequence>
        <xs:element ref="dependency" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="software" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="hardware" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="environmentExtension" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
      <!--    -->
    - <xs:sequence>
        <xs:element ref="software" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="hardware" minOccurs="0" maxOccurs="unbounded" />
        <xs:element ref="environmentExtension" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
      <!--    -->
    - <xs:sequence>
        <xs:element ref="hardware" minOccurs="1" maxOccurs="unbounded" />
        <xs:element ref="environmentExtension" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
      <xs:element ref="environmentExtension" minOccurs="1" maxOccurs="1" />
    </xs:choice>
  </xs:complexType>
  <!--    ****eventIdentifierComplexType
    -->
- <xs:complexType name="eventIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="eventIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="eventIdentifierValue" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
- <!--
       ****eventOutcomeDetailComplexType
    See comment for "creatingApplicationComplexType"
  -->
- <xs:complexType name="eventOutcomeDetailComplexType">
  - <xs:choice>
    - <xs:sequence>
        <xs:element ref="eventOutcomeDetailNote" minOccurs="1" maxOccurs="1" />
        <xs:element ref="eventOutcomeDetailExtension" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
      <xs:element ref="eventOutcomeDetailExtension" minOccurs="1" maxOccurs="1" />
    </xs:choice>
  </xs:complexType>
- <!--
       ****eventOutcomeInformationComplexType
    See comment for "creatingApplicationComplexType"
  -->
- <xs:complexType name="eventOutcomeInformationComplexType">
  - <xs:choice>
    - <xs:sequence>
        <xs:element ref="eventOutcome" minOccurs="1" maxOccurs="1" />
        <xs:element ref="eventOutcomeDetail" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:element ref="eventOutcomeDetail" minOccurs="1" maxOccurs="unbounded" />
```

```
    </xs:choice>
  </xs:complexType>
- <!--

     *******fixityComplexType
  -->
- <xs:complexType name="fixityComplexType">
  - <xs:sequence>
      <xs:element ref="messageDigestAlgorithm" minOccurs="1" maxOccurs="1" />
      <xs:element ref="messageDigest" minOccurs="1" maxOccurs="1" />
      <xs:element ref="messageDigestOriginator" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
- <!--

     *******formatComplexType
  -->
- <xs:complexType name="formatComplexType">
  - <xs:sequence>
    - <xs:choice>
        <!--  one or both of formatDesignation and/or formatRegistry required; followed optionally by formatNote  -->
      - <xs:sequence>
          <xs:element ref="formatDesignation" minOccurs="1" maxOccurs="1" />
          <xs:element ref="formatRegistry" minOccurs="0" maxOccurs="1" />
        </xs:sequence>
        <xs:element ref="formatRegistry" minOccurs="1" maxOccurs="1" />
      </xs:choice>
      <xs:element ref="formatNote" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
- <!--

     *******formatDesignationComplexType
  -->
- <xs:complexType name="formatDesignationComplexType">
  - <xs:sequence>
      <xs:element ref="formatName" minOccurs="1" maxOccurs="1" />
      <xs:element ref="formatVersion" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
- <!--

     *******formatRegistryComplexType

  -->
- <xs:complexType name="formatRegistryComplexType">
  - <xs:sequence>
      <xs:element ref="formatRegistryName" minOccurs="1" maxOccurs="1" />
      <xs:element ref="formatRegistryKey" minOccurs="1" maxOccurs="1" />
      <xs:element ref="formatRegistryRole" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
- <!--

     ***************hardwareComplexType
  -->
- <xs:complexType name="hardwareComplexType">
  - <xs:sequence>
      <xs:element ref="hwName" minOccurs="1" maxOccurs="1" />
      <xs:element ref="hwType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="hwOtherInformation" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
- <!--

     ***************inhibitorsComplexType
  -->
- <xs:complexType name="inhibitorsComplexType">
  - <xs:sequence>
      <xs:element ref="inhibitorType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="inhibitorTarget" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="inhibitorKey" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
  <!-- ******* licenseIdentifierComplexType
    -->
- <xs:complexType name="licenseIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="licenseIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="licenseIdentifierValue" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
- <!--
     ******* licenseInformationComplexType
     See comment for "creatingApplicationComplexType"
  -->
- <xs:complexType name="licenseInformationComplexType">
  - <xs:choice>
    - <xs:sequence>
        <xs:element ref="licenseIdentifier" minOccurs="1" maxOccurs="1" />
        <xs:element ref="licenseTerms" minOccurs="0" maxOccurs="1" />
        <xs:element ref="licenseNote" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
    - <xs:sequence>
```

```
        <xs:element ref="licenseTerms" minOccurs="1" maxOccurs="1" />
        <xs:element ref="licenseNote" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:element ref="licenseNote" minOccurs="1" maxOccurs="unbounded" />
    </xs:choice>
  </xs:complexType>
  <!-- ******* linkingAgentIdentifierComplexType
    -->
- <xs:complexType name="linkingAgentIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="linkingAgentIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="linkingAgentIdentifierValue" minOccurs="1" maxOccurs="1" />
      <xs:element ref="linkingAgentRole" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="LinkAgentXmlID" type="xs:IDREF" use="optional" />
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
- <!--

     ***************"linkingEventIdentifierComplexType
  -->
- <xs:complexType name="linkingEventIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="linkingEventIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="linkingEventIdentifierValue" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="LinkEventXmlID" type="xs:IDREF" use="optional" />
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
  <!-- *******linkingObjectIdentifierComplexType
    -->
- <xs:complexType name="linkingObjectIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="linkingObjectIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="linkingObjectIdentifierValue" minOccurs="1" maxOccurs="1" />
      <xs:element ref="linkingObjectRole" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="LinkObjectXmlID" type="xs:IDREF" use="optional" />
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
- <!--

     ***************linkingIntellectualEntityIdentifierComplexType
  -->
- <xs:complexType name="linkingIntellectualEntityIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="linkingIntellectualEntityIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="linkingIntellectualEntityIdentifierValue" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
- <!--

     ***************linkingRightsStatementIdentifierComplexType
  -->
- <xs:complexType name="linkingRightsStatementIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="linkingRightsStatementIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="linkingRightsStatementIdentifierValue" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="LinkPermissionStatementXmlID" type="xs:IDREF" use="optional" />
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
- <!--

     ***************objectCharacteristicsComplexType
  -->
- <xs:complexType name="objectCharacteristicsComplexType">
  - <xs:sequence>
      <xs:element ref="compositionLevel" minOccurs="1" maxOccurs="1" />
      <xs:element ref="fixity" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="size" minOccurs="0" maxOccurs="1" />
      <xs:element ref="format" minOccurs="1" maxOccurs="unbounded" />
      <xs:element ref="creatingApplication" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="inhibitors" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="objectCharacteristicsExtension" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
- <!--

     *******objectIdentifierComplexType
  -->
- <xs:complexType name="objectIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="objectIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="objectIdentifierValue" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
- <!--

     ***************originalNameComplexType
  -->
- <xs:complexType name="originalNameComplexType">
  - <xs:simpleContent>
```

```
    - <xs:extension base="xs:string">
        <xs:attributeGroup ref="xlink:simpleLink" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
- <!--

    ***************preservationLevelComplexType
  -->
- <xs:complexType name="preservationLevelComplexType">
  - <xs:sequence>
      <xs:element ref="preservationLevelValue" minOccurs="1" maxOccurs="1" />
      <xs:element ref="preservationLevelRole" minOccurs="0" maxOccurs="1" />
      <xs:element ref="preservationLevelRationale" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="preservationLevelDateAssigned" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
- <!--

    ***************relatedEventIdentificationComplexType
  -->
- <xs:complexType name="relatedEventIdentificationComplexType">
  - <xs:sequence>
      <xs:element ref="relatedEventIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="relatedEventIdentifierValue" minOccurs="1" maxOccurs="1" />
      <xs:element ref="relatedEventSequence" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="RelEventXmlID" type="xs:IDREF" use="optional" />
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
- <!--

    ***************relatedObjectIdentificationComplexType
  -->
- <xs:complexType name="relatedObjectIdentificationComplexType">
  - <xs:sequence>
      <xs:element ref="relatedObjectIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="relatedObjectIdentifierValue" minOccurs="1" maxOccurs="1" />
      <xs:element ref="relatedObjectSequence" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
    <xs:attribute name="RelObjectXmlID" type="xs:IDREF" use="optional" />
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
- <!--

    ***************relationshipComplexType
  -->
- <xs:complexType name="relationshipComplexType">
  - <xs:sequence>
      <xs:element ref="relationshipType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="relationshipSubType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="relatedObjectIdentification" minOccurs="1" maxOccurs="unbounded" />
      <xs:element ref="relatedEventIdentification" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <!-- ******** rightsGrantedComplexType
    -->
- <xs:complexType name="rightsGrantedComplexType">
  - <xs:sequence>
      <xs:element ref="act" minOccurs="1" maxOccurs="1" />
      <xs:element ref="restriction" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="termOfGrant" minOccurs="1" maxOccurs="1" />
      <xs:element ref="rightsGrantedNote" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
- <!--

    ******* rightsStatementComplexType
  -->
- <xs:complexType name="rightsStatementComplexType">
  - <xs:sequence>
      <xs:element ref="rightsStatementIdentifier" minOccurs="1" maxOccurs="1" />
      <xs:element ref="rightsBasis" minOccurs="1" maxOccurs="1" />
      <xs:element ref="copyrightInformation" minOccurs="0" maxOccurs="1" />
      <xs:element ref="licenseInformation" minOccurs="0" maxOccurs="1" />
      <xs:element ref="statuteInformation" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="rightsGranted" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="linkingObjectIdentifier" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="linkingAgentIdentifier" minOccurs="0" maxOccurs="unbounded" />
      <!-- The above two, <linkingObjectIdentifier> and <linkingAgentIdentifier>, are different than defined in the data
      dictionary. This will be cited as errata and corrected in the next version of the data dictionary. -->
    </xs:sequence>
  </xs:complexType>
  <!-- *******rightsStatementIdentifierComplexType
    -->
- <xs:complexType name="rightsStatementIdentifierComplexType">
  - <xs:sequence>
      <xs:element ref="rightsStatementIdentifierType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="rightsStatementIdentifierValue" minOccurs="1" maxOccurs="1" />
    </xs:sequence>
    <xs:attributeGroup ref="xlink:simpleLink" />
  </xs:complexType>
- <!--
```

```
    ***************signatureComplexType
  -->
- <xs:complexType name="signatureComplexType">
  - <xs:sequence>
      <xs:element ref="signatureEncoding" minOccurs="1" maxOccurs="1" />
      <xs:element ref="signer" minOccurs="0" maxOccurs="1" />
      <xs:element ref="signatureMethod" minOccurs="1" maxOccurs="1" />
      <xs:element ref="signatureValue" minOccurs="1" maxOccurs="1" />
      <xs:element ref="signatureValidationRules" minOccurs="1" maxOccurs="1" />
      <xs:element ref="signatureProperties" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="keyInformation" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
- <!--

    ***************signatureInformationComplexType
    See comment for "creatingApplicationComplexType"
  -->
- <xs:complexType name="signatureInformationComplexType">
  - <xs:choice>
    - <xs:sequence>
        <xs:element ref="signature" minOccurs="1" maxOccurs="1" />
        <xs:element ref="signatureInformationExtension" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:element ref="signatureInformationExtension" minOccurs="1" maxOccurs="unbounded" />
    </xs:choice>
  </xs:complexType>
- <!--

    ***************significantPropertiesComplexType
    See comment for "creatingApplicationComplexType"

  -->
- <xs:complexType name="significantPropertiesComplexType">
  - <xs:choice>
    - <xs:sequence>
        <xs:element ref="significantPropertiesType" minOccurs="1" maxOccurs="1" />
        <xs:element ref="significantPropertiesValue" minOccurs="0" maxOccurs="1" />
        <xs:element ref="significantPropertiesExtension" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
    - <xs:sequence>
        <xs:element ref="significantPropertiesValue" minOccurs="1" maxOccurs="1" />
        <xs:element ref="significantPropertiesExtension" minOccurs="0" maxOccurs="unbounded" />
      </xs:sequence>
      <xs:element ref="significantPropertiesExtension" minOccurs="1" maxOccurs="unbounded" />
    </xs:choice>
  </xs:complexType>
- <!--

    ***************softwareComplexType
  -->
- <xs:complexType name="softwareComplexType">
  - <xs:sequence>
      <xs:element ref="swName" minOccurs="1" maxOccurs="1" />
      <xs:element ref="swVersion" minOccurs="0" maxOccurs="1" />
      <xs:element ref="swType" minOccurs="1" maxOccurs="1" />
      <xs:element ref="swOtherInformation" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="swDependency" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <!-- ******* statuteInformationComplexType
    -->
- <xs:complexType name="statuteInformationComplexType">
  - <xs:sequence>
      <xs:element ref="statuteJurisdiction" minOccurs="1" maxOccurs="1" />
      <xs:element ref="statuteCitation" minOccurs="1" maxOccurs="1" />
      <xs:element ref="statuteInformationDeterminationDate" minOccurs="0" maxOccurs="1" />
      <xs:element ref="statuteNote" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
- <!--

    ***************storageComplexType
    See comment for "creatingApplicationComplexType"
  -->
- <xs:complexType name="storageComplexType">
  - <xs:choice>
    - <xs:sequence>
        <xs:element ref="contentLocation" minOccurs="1" maxOccurs="1" />
        <xs:element ref="storageMedium" minOccurs="0" maxOccurs="1" />
      </xs:sequence>
      <xs:element ref="storageMedium" minOccurs="1" maxOccurs="1" />
    </xs:choice>
  </xs:complexType>
- <!--

    ***************termOfGrantComplexType
  -->
- <xs:complexType name="termOfGrantComplexType">
  - <xs:sequence>
      <xs:element ref="startDate" minOccurs="1" maxOccurs="1" />
      <xs:element ref="endDate" minOccurs="0" maxOccurs="1" />
    </xs:sequence>
  </xs:complexType>
- <!--
```

```
    ********************************************************************************************
    ********************************Element Declarations ***************************************
    ********************************************************************************************

    ****** string type element declarations

  -->
  <xs:element name="act" type="xs:string" />
  <xs:element name="agentIdentifierType" type="xs:string" />
  <xs:element name="agentIdentifierValue" type="xs:string" />
  <xs:element name="agentName" type="xs:string" />
- <xs:element name="agentType">
  - <xs:annotation>
      <xs:documentation xml:lang="en">Added valuelist TPD 2011-06-10</xs:documentation>
    </xs:annotation>
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:restriction base="xs:anyType">
        - <xs:simpleType>
          - <xs:restriction base="xs:token">
              <xs:enumeration value="person" />
              <xs:enumeration value="organization" />
              <xs:enumeration value="software" />
            </xs:restriction>
          </xs:simpleType>
        </xs:restriction>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
- <xs:element name="contentLocationType">
  - <xs:annotation>
      <xs:documentation xml:lang="en">Added valuelist TPD 2011-06-10</xs:documentation>
    </xs:annotation>
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:restriction base="xs:anyType">
        - <xs:simpleType>
          - <xs:restriction base="xs:token">
              <xs:enumeration value="SIP" />
              <xs:enumeration value="AIP" />
              <xs:enumeration value="AIU" />
              <xs:enumeration value="DIP" />
            </xs:restriction>
          </xs:simpleType>
        </xs:restriction>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="contentLocationValue" type="xs:string" />
  <xs:element name="copyrightStatus" type="xs:string" />
  <xs:element name="copyrightJurisdiction" type="xs:string" />
  <xs:element name="copyrightNote" type="xs:string" />
  <xs:element name="creatingApplicationName" type="xs:string" />
  <xs:element name="creatingApplicationVersion" type="xs:string" />
  <xs:element name="dependencyIdentifierType" type="xs:string" />
  <xs:element name="dependencyIdentifierValue" type="xs:string" />
  <xs:element name="dependencyName" type="xs:string" />
  <xs:element name="environmentCharacteristic" type="xs:string" />
  <xs:element name="environmentNote" type="xs:string" />
  <xs:element name="environmentPurpose" type="xs:string" />
  <xs:element name="eventDetail" type="xs:string" />
  <xs:element name="eventIdentifierType" type="xs:string" />
  <xs:element name="eventIdentifierValue" type="xs:string" />
  <xs:element name="eventOutcome" type="xs:string" />
  <xs:element name="eventOutcomeDetailNote" type="xs:string" />
  <xs:element name="formatName" type="xs:string" />
  <xs:element name="formatNote" type="xs:string" />
  <xs:element name="formatRegistryName" type="xs:string" />
  <xs:element name="formatRegistryKey" type="xs:string" />
  <xs:element name="formatRegistryRole" type="xs:string" />
  <xs:element name="formatVersion" type="xs:string" />
  <xs:element name="hwOtherInformation" type="xs:string" />
  <xs:element name="hwName" type="xs:string" />
  <xs:element name="hwType" type="xs:string" />
  <xs:element name="inhibitorKey" type="xs:string" />
  <xs:element name="inhibitorTarget" type="xs:string" />
  <xs:element name="inhibitorType" type="xs:string" />
  <xs:element name="licenseIdentifierType" type="xs:string" />
  <xs:element name="licenseIdentifierValue" type="xs:string" />
  <xs:element name="licenseNote" type="xs:string" />
  <xs:element name="licenseTerms" type="xs:string" />
  <xs:element name="linkingAgentIdentifierType" type="xs:string" />
  <xs:element name="linkingAgentIdentifierValue" type="xs:string" />
  <xs:element name="linkingAgentRole" type="xs:string" />
  <xs:element name="linkingEventIdentifierType" type="xs:string" />
  <xs:element name="linkingEventIdentifierValue" type="xs:string" />
  <xs:element name="linkingIntellectualEntityIdentifierType" type="xs:string" />
  <xs:element name="linkingIntellectualEntityIdentifierValue" type="xs:string" />
  <xs:element name="linkingObjectIdentifierType" type="xs:string" />
  <xs:element name="linkingObjectRole" type="xs:string" />
  <xs:element name="linkingObjectIdentifierValue" type="xs:string" />
  <xs:element name="linkingRightsStatementIdentifierType" type="xs:string" />
```

```
<xs:element name="linkingRightsStatementIdentifierValue" type="xs:string" />
<xs:element name="messageDigest" type="xs:string" />
<xs:element name="messageDigestAlgorithm" type="xs:string" />
<xs:element name="messageDigestOriginator" type="xs:string" />
<xs:element name="objectIdentifierType" type="xs:string" />
<xs:element name="objectIdentifierValue" type="xs:string" />
<xs:element name="preservationLevelValue" type="xs:string" />
<xs:element name="preservationLevelRole" type="xs:string" />
<xs:element name="preservationLevelRationale" type="xs:string" />
<xs:element name="relatedEventIdentifierType" type="xs:string" />
<xs:element name="relatedEventIdentifierValue" type="xs:string" />
<xs:element name="relatedObjectIdentifierType" type="xs:string" />
<xs:element name="relatedObjectIdentifierValue" type="xs:string" />
<xs:element name="relationshipType" type="xs:string" />
<xs:element name="relationshipSubType" type="xs:string" />
<xs:element name="restriction" type="xs:string" />
- <xs:element name="rightsBasis">
  - <xs:annotation>
      <xs:documentation xml:lang="en">Added valuelist TPD 2011-06-10</xs:documentation>
    </xs:annotation>
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:restriction base="xs:anyType">
        - <xs:simpleType>
          - <xs:restriction base="xs:token">
              <xs:enumeration value="license" />
              <xs:enumeration value="formalTransfer" />
              <xs:enumeration value="accessType" />
            </xs:restriction>
          </xs:simpleType>
        </xs:restriction>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="rightsGrantedNote" type="xs:string" />
  <xs:element name="rightsStatementIdentifierType" type="xs:string" />
  <xs:element name="rightsStatementIdentifierValue" type="xs:string" />
  <xs:element name="signatureEncoding" type="xs:string" />
  <xs:element name="signatureMethod" type="xs:string" />
  <xs:element name="signatureProperties" type="xs:string" />
  <xs:element name="signatureValue" type="xs:string" />
  <xs:element name="signatureValidationRules" type="xs:string" />
  <xs:element name="signer" type="xs:string" />
  <xs:element name="significantPropertiesType" type="xs:string" />
  <xs:element name="significantPropertiesValue" type="xs:string" />
  <xs:element name="storageMedium" type="xs:string" />
  <xs:element name="statuteCitation" type="xs:string" />
  <xs:element name="statuteJurisdiction" type="xs:string" />
  <xs:element name="statuteNote" type="xs:string" />
  <xs:element name="swName" type="xs:string" />
  <xs:element name="swVersion" type="xs:string" />
  <xs:element name="swType" type="xs:string" />
  <xs:element name="swDependency" type="xs:string" />
  <xs:element name="swOtherInformation" type="xs:string" />
- <!--
     ****** complex  type element declarations
  -->
  <xs:element name="agentIdentifier" type="agentIdentifierComplexType" />
  <xs:element name="contentLocation" type="contentLocationComplexType" />
  <xs:element name="copyrightInformation" type="copyrightInformationComplexType" />
  <xs:element name="creatingApplication" type="creatingApplicationComplexType" />
  <xs:element name="dependencyIdentifier" type="dependencyIdentifierComplexType" />
  <xs:element name="dependency" type="dependencyComplexType" />
  <xs:element name="environment" type="environmentComplexType" />
  <xs:element name="eventIdentifier" type="eventIdentifierComplexType" />
  <xs:element name="eventOutcomeDetail" type="eventOutcomeDetailComplexType" />
  <xs:element name="eventOutcomeInformation" type="eventOutcomeInformationComplexType" />
- <xs:element name="eventType">
  - <xs:annotation>
      <xs:documentation xml:lang="en">Added valuelist. TPD 2011-06-10</xs:documentation>
    </xs:annotation>
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:restriction base="xs:anyType">
          <!--  Valid only for AIP & AIU  -->
        - <xs:simpleType>
          - <xs:restriction base="xs:token">
              <xs:enumeration value="Adjustment" />
              <xs:enumeration value="Disposal" />
              <xs:enumeration value="Migration" />
            - <!--
                                                        </xs:restriction>
                                                </xs:simpleType>
                                        </xs:restriction>
                                        <xs:restriction base="xs:contentLocationType">  Valid only for AIC:
                                                <xs:simpleType>
                                                        <xs:restriction base="xs:token">
              -->
              <xs:enumeration value="Creation" />
              <xs:enumeration value="Ingestion" />
              <xs:enumeration value="Deletion" />
            </xs:restriction>
```

```
        </xs:simpleType>
      </xs:restriction>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="fixity" type="fixityComplexType" />
<xs:element name="format" type="formatComplexType" />
<xs:element name="formatDesignation" type="formatDesignationComplexType" />
<xs:element name="formatRegistry" type="formatRegistryComplexType" />
<xs:element name="hardware" type="hardwareComplexType" />
<xs:element name="inhibitors" type="inhibitorsComplexType" />
<xs:element name="licenseIdentifier" type="licenseIdentifierComplexType" />
<xs:element name="licenseInformation" type="licenseInformationComplexType" />
<xs:element name="linkingAgentIdentifier" type="linkingAgentIdentifierComplexType" />
<xs:element name="linkingEventIdentifier" type="linkingEventIdentifierComplexType" />
<xs:element name="linkingIntellectualEntityIdentifier" type="linkingIntellectualEntityIdentifierComplexType" />
<xs:element name="linkingObjectIdentifier" type="linkingObjectIdentifierComplexType" />
<xs:element name="linkingRightsStatementIdentifier" type="linkingRightsStatementIdentifierComplexType" />
<xs:element name="objectCharacteristics" type="objectCharacteristicsComplexType" />
<xs:element name="objectIdentifier" type="objectIdentifierComplexType" />
<xs:element name="originalName" type="originalNameComplexType" />
<xs:element name="preservationLevel" type="preservationLevelComplexType" />
<xs:element name="relatedEventIdentification" type="relatedEventIdentificationComplexType" />
<xs:element name="relatedObjectIdentification" type="relatedObjectIdentificationComplexType" />
<xs:element name="relationship" type="relationshipComplexType" />
<xs:element name="rightsGranted" type="rightsGrantedComplexType" />
<xs:element name="rightsStatement" type="rightsStatementComplexType" />
<xs:element name="rightsStatementIdentifier" type="rightsStatementIdentifierComplexType" />
<xs:element name="signature" type="signatureComplexType" />
<xs:element name="signatureInformation" type="signatureInformationComplexType" />
<xs:element name="significantProperties" type="significantPropertiesComplexType" />
<xs:element name="statuteInformation" type="statuteInformationComplexType" />
<xs:element name="software" type="softwareComplexType" />
<xs:element name="storage" type="storageComplexType" />
<xs:element name="termOfGrant" type="termOfGrantComplexType" />
<!--

    ****** other xs  type  element declarations

-->
<xs:element name="compositionLevel" type="xs:nonNegativeInteger" />
<xs:element name="relatedEventSequence" type="xs:nonNegativeInteger" />
<xs:element name="relatedObjectSequence" type="xs:nonNegativeInteger" />
<xs:element name="size" type="xs:long" />
<!--

    ****** date type  element declarations

-->
<xs:element name="dateCreatedByApplication" type="edtfSimpleType" />
<xs:element name="endDate" type="edtfSimpleType" />
<xs:element name="copyrightStatusDeterminationDate" type="edtfSimpleType" />
<xs:element name="eventDateTime" type="edtfSimpleType" />
<xs:element name="preservationLevelDateAssigned" type="edtfSimpleType" />
<xs:element name="startDate" type="edtfSimpleType" />
<xs:element name="statuteInformationDeterminationDate" type="edtfSimpleType" />
<!--

    ****** extension  type  element declarations

-->
<xs:element name="creatingApplicationExtension" type="extensionComplexType" />
<xs:element name="environmentExtension" type="extensionComplexType" />
<xs:element name="eventOutcomeDetailExtension" type="extensionComplexType" />
<xs:element name="keyInformation" type="extensionComplexType" />
<xs:element name="objectCharacteristicsExtension" type="extensionComplexType" />
<xs:element name="rightsExtension" type="extensionComplexType" />
<xs:element name="signatureInformationExtension" type="extensionComplexType" />
<xs:element name="significantPropertiesExtension" type="extensionComplexType" />
<!--
    ************************************************************************************************
    ********************************** Global Definitions  *****************************************
    ************************************************************************************************

    ************** version definition
-->
<xs:simpleType name="versionSimpleType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="2.0" />
  </xs:restriction>
</xs:simpleType>
<!--

    ***************extensionComplexType
-->
<xs:complexType name="extensionComplexType">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<!--

    ************************************************************************************************

                                date/time Definition:  edtfSimpleType
                                       Extended Date/Time Format
```

```
    ********************************************************************************************************

    edtfSimpleType  is the type used throughout the schema for "date"  and "dateTime" type elements: dateCreatedByApplication, copyri

    iIt s the union of three simple types: xsDate, xs:dateTime -  and edtfRegularExpressions, as folloiws:

  -->
- <xs:simpleType name="edtfSimpleType">
    <xs:union memberTypes="xs:date xs:dateTime edtfRegularExpressions" />
  </xs:simpleType>
- <!--

    "xs:union" (above) means that any string conforming to any one of the types in the union will validate.  xs:date and xs:dateTime a


    ******** edftRegularExpressions
  -->
- <xs:simpleType name="edtfRegularExpressions">
  - <xs:restriction base="xs:string">
      <xs:pattern value="\d{2}(\d{2}|\?\?|\d(\d|\?))(-(\d{2}|\?\?))?~?\??" />
      <xs:pattern value="\d{6}(\d{2}|\?\?)~?\??" />
      <xs:pattern value="\d{8}T\d{6}" />
      <xs:pattern value="((\d{4}(-\d{2})?)|UNKNOWN)/((\d{4}(-\d{2})?)|UNKNOWN|OPEN)" />
    - <!--


        The first  pattern:
        <xs:pattern value="\d{2}(\d{2}|\?\?|\d(\d|\?))(-(\d{2}|\?\?))?~?\??"/>,

        is for year (yyyy) or year-month (yyyy-mm).   The last or last two digits of year may be '?'  meaning "one year in that range
        Hyphen must separate year and month.

        The second pattern:
                                <xs:pattern value="\d{6}(\d{2}|\?\?)~?\??"/>,

        is for  yearMonthDay - yyyymmdd,  where 'dd' may be '??'  so '200412??' means "some day during the month of 12/2004".
        The whole  string may be followed by '?' or '~'  to mean "questionable" or "approximate".     hyphens are  not allowed for thi

        The Third patten:
                                <xs:pattern value="\d{8}T\d{6}"/>,

        is for  date and time with  T separator:'yyyymmddThhmmss'.  Hyphens in date and colons in time not allowed for this pattern.

        And finally:
                                <xs:pattern value="((\d{4}(-\d{2})?)|UNKNOWN)/((\d{4}(-\d{2})?)|UNKNOWN|OPEN)"/>

        is for  a date range.  in years:  'yyyy/yyyy'; or year/month: yyyy-mm/yyyy-mm.    Beginning or end of range value may be 'UNKN
        hyphens mandatory when month is present.

      -->
    </xs:restriction>
  </xs:simpleType>
  <!--    -->
</xs:schema>
```

Vedlegg 5:

EAD i DIAS

EAD *(Encoded Archival Description)* er en standard for å beskrive objekter (logisk) som arkivmateriale. EAD kan betraktes som et utvekslingsformat for ISAD(G), men omfatter en rekke elementer i tillegg. EAD er dokumentert på nettstedet *http://www.loc.gov/ead/*.

DIAS bruker EAD uten egne norske tilpasninger. Men ikke alle elementene i EAD brukes i DIAS. Vedlegg 5 lister opp EAD-elementene med attributter som skal brukes i DIAS. Vedlegg 5 er også en ”mapping”-oversikt som viser hvor disse elementene skal hentes i Asta.

EAD i DIAS

DIAS-prosjektet bruker det offisielle skjemaet fra Library of Congress (LOV) uforandret. Her listes opp de elementene med attributter som skal brukes i DIAS. Det redegjøres også for hvor i Asta man henter de forskjellige dataene. EAD-filen i arkivpakken genereres fra Asta.

<ead> - Encoded Archival Description

Ytre omslag-element for hele filen.

Kan inneholde

- <eadheader> (obligatorisk)
- <archdesc> (obligatorisk)

Attributter

Ingen brukes i DIAS

Eksempel

```
<ead>
    <eadheader>
        <eadid>(...)</eadid>
        <filedesc>
            <titlestmt>
                <titleproper>(...)</titleproper>
            </titlestmt>
        </filedesc>
    </eadheader>
    <archdesc level="series">
        <did>(...)</did>
        <dsc type="combined">(...)</dsc>
    </archdesc>
</ead>
```

<eadheader> - EAD Header

Omslag-element for bibliografisk og deskriptiv informasjon om EAD-dokumentet.

Kan inneholde

- <eadid> (obligatorisk)
- <filedesc> (obligatorisk)

Attributter

Ingen brukes i DIAS.

Eksempel

Se eksempel over, til <ead>

<eadid> - EAD Identifier

Obligatorisk underelement til <eadheader> som tilordner en unik id for et EAD-dokument

Kan inneholde

- #PCDATA

  Her brukes sti og id fra Asta.

Attributter

- countrycode (obligatorisk i DIAS).

  Hentes fra ISO 3166–1
- mainagencycode (obligatorisk i DIAS).

  Samsvarer med ISO/DIS 15511. Ifølge veiledningen på LOCs sider, skal land-koden fjernes og legges i countrycode. Men dette er ikke lov ifølge XML-skjemaet, så land-koden må være med også her. I Norge er det Nasjonalbiblioteket som tildeler koder i samsvar med IS0 15511 (ISIL-koder).

Eksempel

```
<eadid countrycode="NO" mainagencycode="NO-1032207">A-3324/D/Da</eadid>
```

<filedesc> - File Description

Obligatorisk subelement til <eadheader> som samler mye av den bibliografiske informasjonen om EAD-filen.

Kan inneholde

- <titlestmt> (obligatorisk)

Attributter

Ingen brukes i DIAS

Eksempel

```
<filedesc>
    <titlestmt>
        <titleproper>
            Skatteetaten, Sakarkiv <date normal="1998/2005" certainty="ca">1998-2005</date>
        </titleproper>
    </titlestmt>
```

`</filedesc>`

`<titlestmt>` - Title Statement

Obligatorisk omslagselement som grupperer informasjon om navnet til arkivbeskrivelsen og de ansvarlig for å framstille den.

Kan inneholde

- `<titleproper>` (obligatorisk)
- `<author>` (ikke obligatorisk)

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, til `<filedesc>`

`<titleproper>` - Title Proper of the Finding Aid

Navnet til arkivbeskrivelsen.

Kan inneholde

- `#PCDATA`

  Tar med navnet til alle overordnede arkivenheter, atskilt med komma.
- `<date>`

  Hentes fra periode i Asta, dvs. fra- og tildato til arkivenheten som er beskrevet

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, til `<filedesc>`

`<date>` - Date

Generisk element som inneholder år, måned eller dag i hvilket som helst format

Kan inneholde

- `#PCDATA`.

  Da Asta kan angi kun måned og år, altså uten dag, skrives månedene fullt ut, på formatet “2. juni 2010”. Det blir da forståelig med “juni 2010”.

Attributter

- normal (obligatorisk i DIAS).
  Legger inn normalform på formatet yyyy-mm-dd/yyyy-mm-dd evt. kun årstall, som yyyy/yyyy
- certainty (ikke obligatorisk).
  Bruker feltet “datokvalitet” i Asta.

Eksempel

Se eksempel over, til `<filedesc>`

Archival Description

`<archdesc>` - Archival Description

Omslags-element som beskriver innhold, kontekst og omfang til den arkivenheten som skal beskrives.

Kan inneholde

- `<did>` (obligatorisk)
- `<scopecontent>` (ikke obligatorisk)
- `<accessrestrict>` (ikke obligatorisk)
- `<appraisal>` (ikke obligatorisk)
- `<phystech>` (ikke obligatorisk)
- `<processinfo>` (ikke obligatorisk)
- `<dsc>` (ikke obligatorisk)

Attributter

- level (obligatorisk)
  Oversetter Astas arkivenhetstype til level
  Har følgende mapping for de viktigste typene:

```
Asta enhetstype           level
---------------------------------
Arkiv                     fonds
Arkivdel                  subfonds
Serie                     series
Serie under annen serie   subseries
Stykke                    file
Mappe                     file
```

Eksempel

Se eksemplet over, til `<ead>`

<did> - Descriptive Identification

Obligatorisk omslags-element som samler kjerneinformasjon om materialet

Kan inneholde

- <unittitle> (obligatorisk i DIAS)
- <unitdate> (ikke obligatorisk)
- <unitid> (obligatorisk i DIAS)
- <container> (ikke obligatorisk)
- <physdesc> (ikke obligatorisk)
- <note> (ikke obligatorisk)

  Merknad til arkivenhet (under Annet-fanen i Asta)

Attributter

Ingen brukes i DIAS

Eksempel

```
<did>
    <unittitle>Sakarkiv</unittitle>
    <unitdate normal="2000-01-01/2008-12-31" certainty="eksakt">01.01.2000 - 31.12.2008</unitdate
    <unitid countrycode="no" repositorycode="ra">A-0001/D/Da</unitid>
    <physdesc>
        <extent>50 Hyllemeter</extent>
    </physdesc>
</did>
```

<unittitle> Title of the Unit

Navnet til arkivenheten.

Kan inneholde

- #PCDATA

  Bruker “Navn” eller “Alternativt navn” fra Asta

Attributes

- type (ikke obligatorisk)

  Bruker “alternate” for alternative navn i Asta

Eksempel

```
<unittitle>Sakarkiv</unittitle>
```

<unitdate> - Date of the Unit

Dag, måned eller år for når arkivenheten ble dannet.

Kan inneholde

- `#PCDATA`

  Periode fra Asta

Attributter

- normal (obligatorisk i DIAS)

  Bruker formatet 'yy-mm-dd/yy-mm-dd"
- certainty (ikke obligatorisk)

  Verdier i Asta er: eksakt, ca, innen tiår

Eksempel

```
<unitdate normal="2000-01-01/2004-12-31" certainty="eksakt">01.01.2000 - 31.12.2004</unitdate>
```

`<unitid>` - ID of the Unit

Alfanumerisk tekst som tjener som unik referanse til arkivenheten.

Kan inneholde

- `#PCDATA`

  Sti og id fra Asta

Attributter

- countrycode (Skal alltid brukes på archdesc-nivå)
- repositorycode (Skal alltid brukes på archdesc-nivå).

  Skal samsvare med ISO 15511.

Eksempel

```
<unitid countrycode="no" repositorycode="ra">A-3324/D/Da</unitid>
```

`<container>` - Container

Viser typen lagringsenhet og id til lagringsenheten.

Kan inneholde

- `#PCDATA`

  Id til lagringsenheten

Attributter

- type

  Lagringsenhetstype i Asta: Boks, Pakke, Protokoll, Mikrofilmrull

Eksempel

```
<container type="Boks">L0001</container>
```

<physdesc> - Physical Description

Omslags-element for å samle info om fysiske trekk ved arkivenheten.

Kan inneholde

- <extent>
- <genreform>

Attributes

Ingen brukes i DIAS

Eksempel

Se eksempel over, til <did>

<extent> - Extent

Antall eller omfang av arkivenheten

Kan inneholde

- PCDATA.

  Omfang og måleenhet fra Asta.

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, til <did>

<genreform> - Genre / Physical Characteristic

Identifiserer typen materiale. Brukes til å fininndele arkivenhetstyper fra Asta.

Kan inneholde

- #PCDATA

Bruker arkivenhetstype på alle andre nivåer enn arkiv, arkivdel og serie.

Attributter

- source (obligatorisk i DIAS)
  source=“ASTA” ved generering av EAD fra Asta

Eksempel

```
<genreform source="ASTA">Foto</genreform>
```

<note> - Note

Kan inneholde

- <p>
  Merknad til arkivenheten i Asta legges til -element under , jf. eksempel.

Attributter

Ingen brukes i DIAS

Eksempel

```
<archdesc level="fond">
  <did>
    <unittitle>Overprovideringskommisjonen</unittitle>
    <note>Denne kommisjonens forretninger gikk over til 5.
    departement pr. 23/12 1814</note>
  </did>
</archdesc>
```

<scopecontent> - Scope and Content

Tematisk omfang og innhold.

Kan inneholde

- <p>

Attributter

Ingen brukes i DIAS

Eksempel

```
<scopecontent>
    <p>Arkivet inneholder styreprotokoller, møtereferater,
```

```
    reisekassebok, avisutklipp, fotografialbum m.m.</p>
</scopecontent>
```

<p> - Paragraph

En eller flere setninger som danner et logisk tekstavsnitt.

Kan inneholde

- #PCDATA

Attributter

Brukes ikke i DIAS

Eksempel

Se eksempel ovenfor, til <scopecontent>

<accessrestrict> - Conditions Governing Access

Informasjon om betingelser som påvirker tilgjengeligheten til materialet.

Kan inneholde

- <head>
- <p>

Her legges klausulering fra Asta. Det er viktig å registrere elektronisk materiale som er deponert uten at depotinstitusjonen har overtatt ansvaret, også som en klausul, og ikke kun som depotstatus lik “Deponert”. Ellers kommer ikke denne informasjonen inn i EAD-filen.

Attribtutter

Ingen brukes i DIAS

Eksempel

```
<accessrestrict>
  <head>Restriksjon</head>
  <p>Lovhjemlet skjerming av innhold</p>
  <head>Kategori</head>
  <p>Deponert elektronisk materiale, først tilgjengelig etter 25 år</p>
  <head>Hjemmel/avtale</head>
  <p>Riksarkivarens forskrift §8-4</p>
  <head>Antall år</head>
  <p>25</p>
  <head>Rekkevidde</head>
  <p>Hele arkivenheten er skjermet</p>
```

```
</accessrestrict>

<accessrestrict>
  <head>Restriksjon</head>
  <p>Lovhjemlet skjerming av innhold</p>
  <head>Kategori</head>
  <p>Taushetsbelagt / unntatt offentlighet</p>
  <head>Hjemmel/avtale</head>
  <p>Forvaltningsloven §13</p>
  <head>Antall år</head>
  <p>60</p>
  <head>Merknad</head>
  <p>Opplysninger om tilhørighet til politiske partier</p>
  <head>Rekkevidde</head>
  <p>Hele arkivenheten er skjermet</p>
</accessrestrict>
```

`<head>` - Heading

Kan inneholde

- `#PCDATA`

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, til `<accessrestrict>`

`<appraisal>` - Appraisal Information

Bevarings- og kassasjonsbestemmelser for materialet.

Kan inneholde

- `<p>`
- `<head>` Bruker

  og til å angi verdi til alle feltene for bevaring og kassasjon i Asta.

Attributter

Ingen brukes i DIAS

Eksempel

```
<appraisal>
```

```
  <head>Kassasjonsvurdering</head>
  <p>Kasseres</p>
  <head>Bevaringstid</head>
  <p>10 år</p>
  <head>Kassasjonsår</head>
  <p>2020</p>
</appraisal>
```

<processinfo> - Processing Information

Kan inneholde

- <head>
- <p>

Attributter

Ingen brukes i DIAS

Eksempel

```
<processinfo>
    <head>Ordningsgrad</head>
    <p>Ferdig ordnet</p>
```

Vedlegg 6:

EAC-CPF i DIAS

EAC-CPF *(Encoded Archival Context – Corporate bodies, Persons, and Families)* er en standard for å beskrive arkivskapere og andre aktører med tilknytning til arkivobjekter, og supplerer EAD. EAC-CPF er dokumentert på nettstedet *http://eac.staatsbibliothek-berlin.de/schema/cpf.xsd.*

DIAS bruker EAC-CPF uten egne norske tilpasninger. Men ikke alle elementene i EAC-CPF brukes i DIAS. Vedlegg 6 lister opp EAC-CPF-elementene med attributter som skal brukes i DIAS. Vedlegg 6 er også en ”mapping”-oversikt som viser hvor disse elementene skal hentes i Asta.

EAC-CPF i DIAS

DIAS-prosjektet bruker det offisielle skjemaet uforandret. Dette er en veiledning i hvordan et EAC-CPF-dokument genereres basert på informasjon i arkivsystemet Asta.

`<eac-cpf>` - Encoded Archival Context - Corporate Bodies, Persons, and Families

Rot-elementet som inneholder hele beskrivelsen av aktøren.

Kan inneholde

- `<control>` (obligatorisk)
- `<cpfDescription>` (obligatorisk i DIAS)

Attributter

Bruker ingen i konvertering fra Asta

Eksempel

```
<eac-cpf>
    <control>
        <recordId>A-0001</recordId>
        <maintenanceStatus>derived</maintenanceStatus>
        <maintenanceAgency>
            <agencyCode>NO-1032207</agencyCode>
            <agencyName>Riksarkivet<agencyName>
        </maintenanceAgency>
        <maintenanceHistory>(...)</maintenanceHistory>
    </control>
    <cpfDescription>
        <identity>
            <entityId>S-3470</entityId>
            <entityType>corporateBody</entityType>
            <nameEntry>(...)</nameEntry>
        </identity>
        <description>
            <existDates>(...)</existDates>
            <place>(...)</place>
            <occupation>(...)</occupation>
            <generalContext>(...)</generalContext>
            <biogHist>(...)</biogHist>
        </description>
    </cpfDescription>
    <relations>...</relations>
```

```
</eac-cpf>
```

<control> - Control

Inneholder kontroll-informasjon om aktørens identitet, opprettelse, type osv.

Kan inneholde

- <maintenanceAgency> (obligatorisk)
- <maintenanceHistory> (obligatorisk)
- <maintenanceStatus> (obligatorisk)
- <recordId> (obligatorisk)

Attributter

Bruker ingen i konvertering fra Asta

Eksempel

Se eksempel over, under

<recordId> - Record Identifier

Unik identifikator for EAC-CPF-dokumentet

Kan inneholde

- NMTOKEN
  Bruker identifikator fra Asta

Attributter

Bruker ingen

Eksempel

Se eksempel over, under <eac-cpf>

<maintenanceStatus> - Maintenance Status

Den aktuelle statusen til EAC-CPF-dokumentet.

Kan inneholde

- “derived”
  Indikerer at beskrivelsen stammer fra et annet system (Asta).

Attributter

Bruker ingen

Eksempel

```
<maintenanceStatus>derived</maintenanceStatus>
```

<maintenanceAgency> - Maintenance Agency

Institusjonen som er ansvarlig for vedlikehold av EAC-CPF-dokumentet.

Kan inneholde

- <agencyCode> (obligatorisk i DIAS)
- <agencyName> (obligatorisk)

Attributter

Bruker ingen

Eksempel

Se eksempel over, under <eac-cpf>

<agencyCode> - Agency Code

Koden som representerer institusjonen ansvarlig for vedlikehold av EAC-CPF-dokumentet.

Kan inneholde

- [text]

  Koden følger ISO 15511. Dvs. samme kode som mainagencycode i EAD. I Norge er det Nasjonalbiblioteket som tildeler koder.

Attributter

Bruker ingen

Eksempel

Se eksempel over, under

<agencyName> - Agency Name

Nvnet til institusjonen ansvarlig for vedlikehold av EAC-CPF-dokumentet.

Kan inneholde

- [text]

Navn fra depotinstitsjonsnavn i Asta

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under

`<maintenanceHistory>` - Maintenance History

Historien til dannelsen og vedlikehold av EAC-CPF-dokumentet

Kan inneholde

- (obligatorisk)

Attributter

Ingen brukes i DIAS

Eksempel

```
<maintenanceHistory>
    <maintenanceEvent>
        <eventType>derived</eventType>
        <eventDateTime standardDateTime="2011-04-14">14.04.2011</eventDateTime>
        <agentType>machine</agentType>
        <agent>Betty</agent>
        <eventDescription>Konvertert fra Asta</eventDescription>
    </maintenanceEvent>
</maintenanceHistory>
```

`<maintenanceEvent>` - Maintenance Event

En vedlikeholds-hendelse for EAC-CPF-dokumentet. Omslags-element.

Kan inneholde

- `<agent>` (obligatorisk)
- `<agentType>` (obligatorisk)
- `<eventDateTime>` (obligatorisk)
- `<eventDescription>` (ikke obligatorisk)
- `<eventType>` (obligatorisk)

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under `<maintenanceHistory>`

`<eventType>` - Maintenance Event Type

Typen vedlikeholds-hendelse for EAC-CPF-dokumentet.

Kan inneholde

- “derived”

  Brukes for å angi at EAC-CPF-dokumentet er produsert fra Asta-info

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under `<maintenanceHistory>`

`<eventDateTime>` - Maintenance Event Date and Time

Dato og tidspunkt for en vedlikeholds-hendelse for EAC-CPF-dokumentet.

Kan inneholde

- [text]

  Dato for når EAC-CPF-dokumentet ble generert

Attributter

- standardDateTime (ikke obligatorisk)

Eksempel

```
<eventDateTime standardDateTime="2012-04-14">14.04.2012<eventDateTime>
```

`<agentType>`

Typen agent ansvarlig for en vedlikeholds-hendelse for EAC-CPF-dokumentet.

Kan inneholde

- “human”
- “machine”

  Bruker “machine” for å angi konvertering fra Asta til EAD-CPF-dokument

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under `<maintenanceHistory>`

`<agent>` - Agent

Agenten (menneske el. maskin) ansvarlig for en hendelse i vedlikeholdet av EAC-CPF-dokumentet.

Kan inneholde

- [text]
  Ved konvertering fra Asta brukes navn på programmet som står for konverteringen.

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under `<maintenanceHistory>`

`<cpfDescription>` - Corporate Body, Person, or Family Description

Inneholder beskrivelsen av en agent.

Kan inneholde

- `<description>` (obligatorisk i DIAS)
- `<identity>` (obligatorisk)

`<relations>` tas ikke med, da vi ikke kan være sikre på at EAC-CPF-dokumentet det lenkes til, eksisterer.

Attributter

Ingen brukes i DIAS

Eksempel

```
<cpfDescription>
    <identity>
        <entityId>S-3470</entityId>
        <entityType>corporateBody</entityType>
        <nameEntry localType="offisielt">
            <part>Skattedirektoratet</part>
```

```
        </nameEntry>
    </identity>
    <description>
        <existDates>(...)</existDates>
        <generalContext localType="samfunnssektor">(...)</generalContext>
        <generalContext localType="aktortype">(...)</generalContext>
        <biogHist>(...)</biogHist>
    </description>
</cpfDescription>
```

`<identity>` - Identity

Et omslags-element for navnedelen av et EAC-CPF-dokument.

Kan inneholde

- `<entityId>` (ikke obligatorisk)
- `<entityType>` (obligatorisk)
- `<nameEntry>` (obligatorisk i DIAS)

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under `<cpfDescription>`

`<entityId>` - Entity Identifier

En formell identifikator brukt for betegne aktøren som blir beskrevet.

Kan inneholde

- [text]
  Bruker identifikator fra Asta. Merk at samme identifikator brukes for

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under `<cpfDescription>`

`<entityType>` - Entity Type

Type aktør.

Kan inneholde

- “person”
- “corporateBody”

En person i Asta mappes til “person” og en virksomhet til “corporateBody”

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under

`<nameEntry>` - Name Entry

Et element som inneholder navne-innførsel for aktøren.

Kan inneholde

- `<part>` (obligatorisk)
- `<useDates>` (ikke obligatorisk)

Attributter

- localType
  “offisielt” eller “uoffisielt”.

Eksempel

Se eksempel over, under `<cpfDescription>`

`<part>` - Part

Brukes for å skille ut komponenter av navnet til en aktør.

Kan inneholde

- [text]
  Navn fra Asta

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under

<useDates> - Date of Use

Datoene når navnet eller navnene var brukt av den beskrevne aktøren.

Kan inneholde

- <date> (ikke obligatorisk)
- <dateRange> (ikke obligatorisk)
- <dateSet> (ikke obligatorisk)

Ved eksport fra Asta, brukes kun <dateRange>

Attributter

Ingen brukes i DIAS

Eksempel

```
<nameEntry localType="offisielt">
    <part>Statens bunadsnemnd</part>
    <useDates>
        <dateRange>
            <fromDate standardDate="1947">1947</fromDate>
            <toDate standardDate="1955-06-01>1. juni 1955</toDate>
        </dateRange>
    </useDates>
</nameEntry>
```

<dateRange> - Date Range

Angir en periode.

Kan inneholde

- <fromDate> (ikke obligatorisk)
- <toDate> (ikke obligatorisk)

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under <useDates>

<fromDate> - From Date

Startdatoen i en periode

Kan inneholde

- [text]

Skriver måneder fullt ut ved generering av fil fra Asta. Dette fordi det i Asta er lov å angi dato som måned og år, uten dag. Og det er mer forståelig med formatet “juni 2006” enn “06.2006”.

Attributter

- standardDate (ikke obligatorisk)

  Angis på formatet ‘yyyy-mm-dd’. Bruker kun årstall hvis kun årstall er angitt i Asta. Hvis kun årstall og måned er angitt i Asta, benyttes formatet ‘yyyy-mm–00’, dvs. dag settes til ‘00’.

Eksempel

Se eksempel over, under `<useDates>`

`<toDate>` - To Date

Sluttdatoen i en periode

Kan inneholde

- [text]

jf. `<fromDate>`

Attributter

- standardDate (ikke obligatorisk)

jf. `<fromDate>`

Eksempel

Se eksempel over, under `<useDates>`

`<description>` - Description

Et omslags-element for alle elementer som beskriver aktøren.

Kan inneholde

- `<existDates>` (ikke obligatorisk)
- `<occupation>` (ikke obligatorisk)
- `<generalContext>` (ikke obligatorisk)

Attributter

Ingen brukes i DIAS

Eksempel

```
<description>
    <existDates>
        <dateRange>
            <fromDate standardDate="1992-01-01">01.01.1992</fromDate>
            <toDate standardDate="1996-12-31">31.12.1996</toDate>
        </dateRange>
    </existDates>
    <generalContext localType="samfunnssektor">
        <p>Kommune</p>
    </generalContext>
    <generalContext localType="aktørtype">
        <p>Fagetat</p>
    </generalContext>
    <generalContext localType="forvaltningsnivå">
        <p>Fagetater</p>
    </generalContext>
    <generalContext localType="forvaltningsområde">
        <p>Helse og sosiale saker</p>
    </generalContext>
    <biogHist>
        <p>Opprettet ved kongelig resolusjon i 1992</p>
    </biogHist>
</description>
```

`<existDates>` - Dates of Existence

Eksistensdatoer for aktøren, f.eks. datoer for etablering og oppløsning av institusjoner, og fødsels- og dødsdato for personer.

Kan inneholde

- `<date>` (ikke obligatorisk)
- `<dateRange>` (ikke obligatorisk)
- `<dateSet>` (ikke obligatorisk)

Bruker `<dateRange>` ved eksport fra Asta til EAC-CPF-dokument.

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under `<description>`

`<occupation>` - Occupation

Beskriver yrket til aktøren.

Kan inneholde

- `<term>`

Attributter

Ingen brukes i DIAS.

Eksempel

```
<occupation>
    <term>Fotograf</term>
</occupation>
```

`<term>` - Term

Generisk element brukt til å legge inn en deskriptiv betegnelse i samsvar med lokale regler.

Kan inneholde

- [text]

Attributter

Ingen brukes i DIAS.

Eksempel

Se eksempel over, under `<occupation>`

`<generalContext>` - General Context

Holder informasjon om den generelle samfunnsmessige og kulturelle konteksten til aktøren.

Kan inneholde

- `<list>`
- `<p>`

Attributter

- localType

Utdyper hvilken kontekst der er snakk om. Bruker ledetekst fra Asta som verdi, men kun små bokstaver.

Eksempel

Se eksempel over, under `<description>`

`<list>` - List

Omslags-element for en enkel liste med elementer

Kan inneholde

- `<item>`

Attributter

Ingen brukes i DIAS

Eksempel

```
<generalContext localType="næringskategori">
    <list>
        <item>Trykking av aviser</item>
        <item>Annen grafisk produksjon</item>
    </list>
</generalContext>
```

`<item>` - Item

Brukes for hver av de enkelte postene i en liste.

Kan inneholde

- [text]
  Hver av postene i Næringskategorier og Organisasjonskategorier i Asta.

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, til `<list>`

`<biogHist>` - Biographical or Historical Note

Essay som gir biografisk eller historisk informasjon om aktøren.

Kan inneholde

- <p>

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, under <description>

<relations> - Relations

Omslags-element for å gruppere relasjoner

Kan inneholde

- <cpfRelation>

Attributter

Ingen brukes i DIAS

Eksempel

```
<relations>
    <cpfRelation cpfRelationType="hierarchical-parent">
        <objectXMLWrap>
            <cpfDescription>
                <identity>
                    <entityId>S-3F-11079</entityId>
                    <entityType>corporateBody</entityType>
                    <nameEntry>
                        <part>Justisdepartementet</part>
                    </nameEntry>
                </identity>
                <description>...</description>
            </cpfDescription>
        </objectXMLWrap>
        <dateRange>
            <fromDate standardDate="1982-01-01">1. januar 1982</fromDate>
            <toDate standardDate="2005-12-31">31. desember 2005</fromDate>
        </dateRange>
    </cpfRelation>
</relations>
```

<cpfRelation> - Corporate Body, Person, or Family Relation

Virksomhet eller person knyttet til den beskrevne aktøren. Vi tar kun med relasjonene “er del av” og “er underlagt”

Kan inneholde

- `<dateRange>`
- `<objectXMLWrap>`

Attributter

- cpfRelationType

  “er del av” i Asta mappes til “hierarchical-parent”. “er underlagt” i Asta mappes til “associative”

Eksempel

Se eksempel over, under `<relations>`

`<objectXMLWrap>` - Object XML Wrap

Brukes for å inkorporere xml-elementer fra andre namespaces

Kan inneholde

- `<cpfDescription>` til tilknyttet aktør

Attributter

Ingen brukes i DIAS

Eksempel

Se eksempel over, til `<relations>`

Vedlegg 7:

Kravspesifikasjon til forvaltningssystem for DIAS-arkivpakker

DIAS-prosjektet utarbeidet 23.11.2010 en kravspesifikasjon til et fullstendig forvaltnings-system for DIAS-baserte arkivpakker i digitalt depot. Denne kravspesifikasjonen har dannet basis for tilpasningen av ESSArch som DIAS-prosjektets forvaltningssystem.

Her vises kravspesifikasjonen i justert versjon av 26.08.2011, som også lå til grunn for ESSArch-utviklingen.

Oppdatert 26.08.2011

Kravspesifikasjon til forvaltningssystem for DIAS-arkivpakker

Dette dokumentet har to deler: en del med generelle krav til et forvaltningssystem (del 1) og en systematisk del (del 2), hvor krav er knyttet direkte til hver enkelt av aktivitetsfasene ved håndteringen av DIAS-arkivpakker i et digitalt depot.

Del 1 representerer DIAS-prosjektets generelle kravspesifikasjon til et forvaltningssystem for arkivpakker i et digitalt depot. Det er da tale om et depot som bygger på OAIS-modellen. Formålet med *del 2* er å konkretisere og anskueliggjøre kravene. Av denne grunn skilles det i del 2 også mellom hva som skal skje i hver av aktivitetsfasene mht. funksjoner og rutiner, og hva som kreves av forvaltningssystemet i vedkommende fase.

Også de generelle kravene i del 1 har to deler. Den primære og innledende delen omfatter krav til et system begrenset til håndteringen av *arkivobjekter*. Administrasjon av lagring – av medier og av "bits" og filer som *digitale objekter* – kreves i tillegg[1]. Et separat lagringssystem kan forutsettes å bli brukt for dette formålet, men ikke nødvendigvis. Et forvaltningssystem for arkivobjekter kan også være kombinert med grunnfunksjoner for lagringsadministrasjon. Med tanke på en slik mulighet er del 1 supplert med basale tilleggskrav til lagrings-administrasjon (punkt 18 - 26) – som en opsjon.

Del 1: Generelle krav til et forvaltningssystem

Forvaltningssystemet skal kunne utføre følgende funksjoner:

1. Kontrollere definerte arbeids- og lagringsområder utenfor forvaltningssystemets egen database
   - Autorisere brukertilgang til disse områdene
   - Logge opprettelse, oppdatering og sletting av filer på områdene
   - Kontrollere (sammenligne) filer fra arbeidsområder med originalversjoner på et dedikert "kontrollområde" som utelukkende systemet har tilgang til (forvaltningssystemets eget arbeidsområde).
2. Lagre nøkkelinformasjon om arkivpakker og om tilknyttede utførte operasjoner på arbeidsområder og systemets kontrollområde i en tilhørende database.
3. Hente en avleveringspakke – en DIAS-strukturert SIP eller en annen vilkårlig sammen-setning av filer/objekter – inn til et definert arbeidsområde, foreta utpakking av filer og lagre filer på arbeidsområdet.
4. Generere sjekksummer for filer/objekter og for samlede pakker.
5. Verifisere sjekksummer for filer/objekter og for samlede pakker.

[1] DIAS-modellen skal også kunne brukes ved off-line lagring på CD/DVD, og i dette tilfellet vil det bare kreves et enkelt system for lagringsadministrasjon.

6. Generere tar-filer for vilkårlige grupperinger av objekter.
7. Generere en arkivpakke basert på basert på DIAS-METS, DIAS-PREMIS, EAD og EAC-CPF **ï** alternativt som en bevaringspakke (AIP), tillegg til bevaringspakke (AIU), brukspakke (DIP) og samlepakke (AIC). Prinsipielt skal det også kunne genereres en avleveringspakke (SIP).
8. Eksportere informasjon fra og om en arkivpakke fra systemets sentrale kontrollområde til:
   - et eksternt system (typisk: et sentralt arkivinformasjonssystem)
   - et eventuelt eget lagringssystem for digitalt depot.
9. Importere nøkkelinformasjon om en arkivpakke (herunder pakkens samlede sjekksum) til forvaltningssystemets egen database.
10. Lagre/plassere arkivpakke (AIP, AIU, AIC, DIP) i depot ved å skrive til vilkårlig valgt lagringsmedium– også i flere eksemplarer på ulike lagringsteknologier.
11. Hente ut (kopiere) arkivpakke fra depot til systemets sentrale kontrollområde for oppdatering.
12. Logge innlegging av arkivpakker i depot, uthenting (kopiering) av pakker til sentralt kontrollområde, oppdatering av pakker og kopiering av objekter ut av sentralt kontroll-område (ved bruk av terminalaksess).
13. Gi samlet oversikt over lagrede arkivpakker i depot – organiseringen av pakker i AIC-er, den enkelte pakkes ID (URN) og type, opprettelsestidspunkt, "flagg" som viser om pakken er en aktiv (gjeldende) versjon, tilgangsbestemmelser, klausul og eventuell sikkerhetsmerking.
14. Gi mulighet for fremfinning av arkivpakker i depot.
15. Gi mulighet for å styre tilgang til lagrede arkivpakker i depot (adgang til å kopiere til arbeidsområdet). Tilgang skal kunne styres individuelt for de enkelte arkivpakker på grunnlag av definert tilgangskategori.
16. Gi mulighet for å sperre for uthenting/kopiering av definerte arkivpakker fra depot på grunnlag av definert tilgangskategori.
17. Gi mulighet for å produsere rapporter, eksempelvis: oversikt over lagrede pakker i depot, pakker med sensitive personopplysninger, utført verifisering av sjekksummer og utførte operasjoner ved generering og oppdatering av pakker.

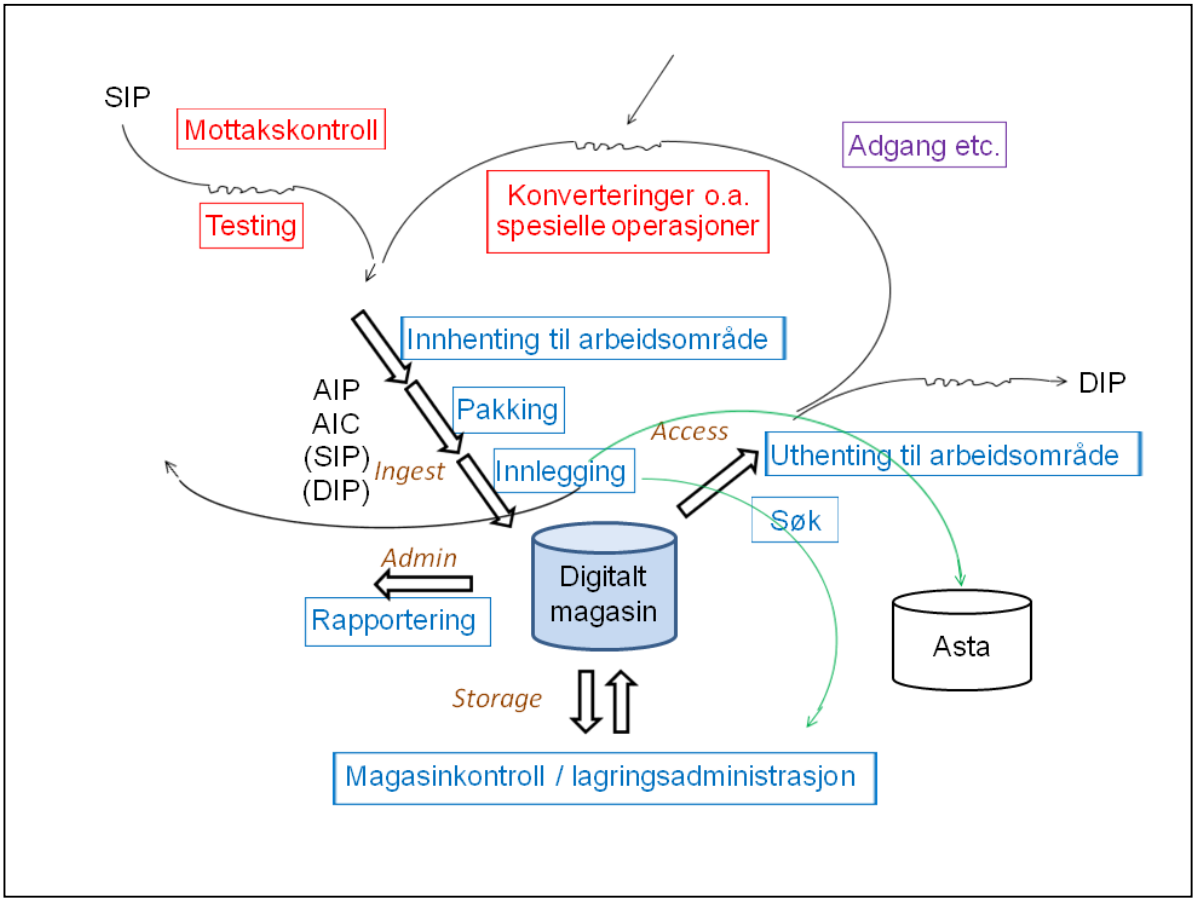
<u>Tilleggskrav til lagringssystem</u> (opsjon i de tilfeller lagringsadministrasjon inngår)

18. Filsystemets konsistens må kunne verifiseres.
19. Det må finnes mekanismer som sikrer synkronisert lagring av versjoner (kopier) av samme objekt på ulike medier (typisk disk og tape-robot).
20. Alle (identiske) kopier av de lagrede objektene må kunne lokaliseres.
21. Objekter må kunne migreres til nye medier, og medie-migrering må kunne verifiseres.
22. Alle former for korrumpering eller tap av data må kunne detekteres umiddelbart, også feil ved migrering eller synkronisering av kopier.
23. Det må finnes mekanismer for å monitorere bestanden av lagret informasjon, og bekrefte at den faktisk omfatter de objekter som skal finnes der – og bare disse.

24. De lagrede objektenes integritet må kunne monitoreres ved kontroll av sjekksummer.
25. Mekanismer for validering og logging av tilgang bør kunne implementeres på objektnivå slik at de omfatter individuelle arkivpakker, jf. også punkt 15 og 16.
26. All kopiering av objekter ut av digitalt depot må kunne logges.

Del 2: Krav til forvaltningssystemet i de enkelte håndteringsfaser

De ulike aktivitetsfasene ved håndtering av DIAS-arkivpakker i et digitalt depot er illustrert i figuren nedenfor:



Spesielt om aktiviteter i røde bokser (ðMottakskontrollò, ðTestingòog ðSpesielle operasjonerð): Det forventes ikke at forvaltningssystemet skal støtte disse funksjonene, men det skal kunne lagre resultater av operasjonene.

A. Mottakskontroll

Funksjoner og rutiner

En avleveringspakke (SIP) skal håndteres i et kontrollert miljø. Den skal integritetssikres ved mottak for å hindre en senere uautorisert endring av innhold, og for å kunne verifisere at innholdet er bevart uendret fra og med mottak. Det skal derfor genereres en samlet sjekksum ved mottak av en SIP. Sjekksum(mer) som følger med en SIP, skal verifiseres, enten som ledd i mottakskontrollen eller i tilknytning til etterfølgende testing, jf. punkt B. I mottakskontrollen inngår også følgende øvrige oppgaver: Materialet skal sjekkes for virus ï etter en definert periode med karantene. Det skal kontrolleres at tilfredsstillende dokumentasjon er vedlagt, og foretas en initiell kontroll av at informasjonsinnholdet er i samsvar med bestemmelser og inngåtte avtaler. Mottakskontrollen må ivareta krav til konfidensialitetssikring. Utførte aktiviteter ved kontrollen skal dokumenteres og autentiseres.

Innledende virussjekk forutsettes å bli foretatt på frittstående utstyr.

Krav til forvaltningssystemet

A.1 Mottatte objekter (SIP) skal kunne hentes inn til kontrollert arbeids- og lagringsområde. Alternativt skal det kunne opprettes et separat område med dedikerte tilgangsrettigheter for å innhente og lagre en SIP med dokumentasjon fra eksternt utført mottakskontroll.

A.2 Det skal kunne genereres en samlet sjekksum for en mottatt SIP.

A.3 Det skal være mulig å verifisere sjekksummer som følger med en SIP (knyttet til samlet innhold og/eller til selve transporten).

A.4 Mottatte objekter skal kunne pakkes ut.

A.5 Utførte aktiviteter ved mottakskontrollen skal kunne logges (registreres) i forvaltnings-systemet og/eller lagres som en egen dokumentasjonsfil med tilknyttet sjekksum.

A.6 Henting/kopiering av objekter til og fra kontrollert arbeidsområde skal logges automatisk av systemet. Loggen skal også vise hvem som har utført operasjonene.

B. Testing

Funksjoner og rutiner

Prosedyrer skal utføres for å verifisere om innholdet i en SIP er teknisk konsistent, korrekt og komplett, om det oppfyller definerte struktur- og formatkrav, og om det er tilknyttet de tekniske og logiske metadata som kreves for å bevare informasjonen med opprettholdt lesbarhet og autentisitet. Sjekksummer for enkeltfiler i SIP-en skal verifiseres (eventuelt også samlet SIP-sjekksum, jf. punkt A). Testingen skal utføres innenfor et kontrollert og beskyttet område, og ivareta krav til konfidensialitetssikring. De utførte operasjonene og resultatene av testingen skal dokumenteres og autentiseres.

Krav til forvaltningssystemet

B.1 En SIP med tilhørende objekter og sjekksummer skal kunne hentes inn til et eget (dedikert) kontrollområde for testing. Både eksport av objekter til testområdet og tilbakehenting av objekter til sentralt kontrollområde skal logges automatisk av systemet.

B.2 Sjekksummer for enkeltfiler skal kunne verifiseres ved å generere nye sjekksummer for sammenligning med de mottatte.

B.3 Utførte aktiviteter ved testingen skal kunne registreres i en logg.

B.4 Operasjonslogger og dokumentasjon av testresultater skal kunne lagres integritetssikret med sjekksummer.

Ingest

C. Innhenting

Funksjoner og rutiner

Genereringen av en bevaringspakke (AIP) må tilrettelegges. Første versjon (generasjon) av en AIP skal være den opprinnelige SIP-en slik den ble mottatt. Dokumentasjon av operasjonene ved mottak og testing skal lagres i AIP versjon 2 dersom det også blir generert og lagret en slik versjon ved lagringen av SIP-en. Dersom det ikke blir generert en AIP nr. 2 ved lagringen av SIP-en, skal operasjonene ved mottak og testing av vedkommende SIP lagres i en separat AIU som er tilknyttet SIP-en via overordnet AIC.

Objektene som skal inngå i AIP-er, AIC-er og AIU-er, skal hentes fra dedikerte områder til et sentralt, beskyttet kontrollområde (arbeids- og lagringsområde) som første trinn i arbeidet med å generere en AIP. En eksisterende AIP og den overordnede AIC (arkivsamlepakke) som den er tilknyttet, skal også hentes ut fra digitalt depot til det sentrale kontrollområdet for å kunne oppdateres, jf. punkt D og E. Fra kontrollområdet skal det tilrettelegges for senere eksport av opplysninger i eller om arkivpakker, herunder til et sentralt arkiv-informasjonssystem (vanligvis Asta), jf. punkt E.

Krav til forvaltningssystemet

C.1 En SIP med sjekksummer og tilhørende operasjonslogger og dokumentasjon fra mottaks-kontroll og testing skal kunne hentes inn til sentralt kontrollområde for å tilrettelegges som en AIP.

C.2 En innhentet SIP skal kunne verifiseres mot SIP-innholdet på testområdet med logg-hendelser.

C.3 Innhentet dokumentasjon av operasjoner ved mottak og testing skal kunne verifiseres.

C.4 Metadata som skal finnes som egne objekter i en AIP og en AIC, skal kunne hentes fra SIP og fra logger/dokumentasjon av operasjonene ved mottak og testing, og lagres som egne filer. Nødvendige tilleggsmetadata skal kunne registreres og lagres som egne filer.

C.5 Nygenererte filer skal kunne integritetssikres med sjekksummer.

C.6 Data skal kunne hentes ut for å tilrettelegges for senere eksport til arkivinformasjons-system (ASTA) samt til forvaltningssystemets egen database.

C.7 Utførte aktiviteter i fasen med innhenting skal kunne registreres i en logg.

C.8 Trafikk (henting/kopiering av objekter) inn/ut av dedikerte arbeidsområder - også mellom dem - skal styres av forvaltningssystemet, jf. punkt A.6.

D. Pakking

Funksjoner og rutiner

En ny arkivpakke skal genereres som en AIP eller DIP (bruksversjon) i henhold til reglene for organisering og pakking i DIAS XML-skjema. For en ny AIP skal det også genereres en ny, overordnet AIC med peker til vedkommende AIP. For hver arkivpakke skal det genereres en samlet sjekksum. En AIP skal også pakkes som en tar-fil, og pakkens sjekksum skal i dette tilfellet være knyttet til tar-filen. Samlet sjekksum for en AIP skal ligge i tilhørende AIC (AIC-ens samlede sjekksum skal lagres i forvaltningssystemet, eventuelt også eksporteres til SAN-systemet og til arkivinformasjonssystemet, jf. punkt E). Oppdatering av en eksisterende AIP eller AIU skal skje ved at det pakkes og genereres en ny, fullstendig AIP- eller AIU-versjon med utgangspunkt i den foregående. Dette krever i seg selv en oppdatering av tilhørende AIC ***i*** *med ny pakking og generering av ny samlet sjekksum.*

Krav til forvaltningssystemet

D.1 En SIP, AIP, AIC og DIP skal kunne genereres etter reglene i DIAS XML-skjema, tildeles en unik ID (URN), og organiseres som en METS-fil med PREMIS innbygd.

D.2 En SIP, AIP og DIP skal i tillegg kunne pakkes som en tar-fil.

D.3 Samlet sjekksum for en SIP, AIP og DIP skal kunne genereres etter tar-pakking.

D.4 En SIP, AIP, AIC og DIP skal kunne valideres i forhold til DIAS XML-skjema.

D.5 Utførte aktiviteter i fasen med pakking skal kunne registreres i en logg og lagres i forvaltningssystemet.

E. Innlegging

Funksjoner og rutiner

Nye og oppdaterte arkivpakker skal overføres til digitalt depot og innlemmes. Informasjon om arkivpakkene skal dessuten kommuniseres til arkivinformasjonssystem (Asta), forvaltningssystemets database og et eventuelt eget system for lagringsadministrasjon (SAN-system). En oppdatert versjon av en AIP skal innlemmes i digitalt depot som en ny versjon i tillegg til den foregående. Den krever at det også innlemmes en tilhørende overordnet AIC i oppdatert versjon. En oppdatert AIC skal imidlertid erstatte foregående AIC-versjon. Istedenfor å oppdateres kan en eksisterende AIP tilknyttes en enkel, frittstående AIU, bestående f.eks. av en oppdatert ADDML-fil. Teknisk skal en slik AIU alltid tilknyttes en AIP via dens overordnede AIC. Lagring av en AIU i digitalt depot krever følgelig også oppdatering og ny innlegging av vedkommende AIC.

Krav til forvaltningssystemet

E.1 Nygenererte og oppdaterte arkivpakker skal kunne lagres i digitalt depot. Systemet skal bare tillate lagring av en ny eller oppdatert AIP, AIU eller DIP når den lagres sammen med en tilknyttet ny eller oppdatert AIC.

E.2 En oppdatert versjon av en AIP skal lagres i tillegg til den foregående, og ikke kunne overskrive den. En oppdatert versjon av en AIC skal erstatte den foregående. En oppdatert DIP skal også erstatte den foregående. En lagret DIP skal dessuten kunne slettes.

E.3 En AIU (med tilknytning til overordnet AIC) skal kunne lagres i digitalt depot som et eget objekt. En AIU skal miste flagg for "aktiv" i forvaltningssystemets database eller alternativt kunne slettes automatisert ved innlegging av en ny, oppdatert AIP-versjon under vedkommende AIC.

E.4 Tilleggsinformasjon om en innlemmet arkivpakke skal kunne overføres til et eventuelt separat system for lagringsadministrasjon (SAN-system), herunder sjekksum for overordnet AIC.

E.5 Lagringsbekreftelse og Magasin-ID for arkivpakke skal kunne mottas fra eget lagringsadministrasjonssystem – om dette finnes.

E.6 Informasjon om den enkelte arkivpakke – herunder pakke-ID, (evt.) magasin-ID, status for AIP og for tilknyttede AIU-er ("flagg" for status = aktiv/ikke aktiv), tilgangsbestemmelser, forekommende formattyper, sikkerhetsmerke/markør, type klausul (restriksjon) og varighet for klausul – skal kunne overføres til forvaltningssystemet og lagres i dets database.

E.7 Tilrettelagt arkiv- og aktørbeskrivelse skal kunne eksporteres til bruk for sentralt arkivinformasjonssystem (ASTA), jf. punkt C.6.

E.8 Innlegging av objekter i digitalt depot skal logges automatisk av systemet.

E.9 Utførte aktiviteter i fasen med pakking skal kunne registreres i en logg og lagres i forvaltningssystemet.

Aksess

F. Uthenting

Funksjoner og rutiner

En arkivpakke (AIP + AIC) skal kunne hentes fra digitalt depot til sentralt arbeidsområde for å oppdateres eller for å tilpasses som en DIP. Teknisk skal en slik uthenting alltid skje som en kopiering av objekter fra digitalt depot. En AIP som hentes ut for oppdatering, skal kopieres

(og senere oppdateres) sammen med dens tilhørende AIC. Det samme gjelder for en eksisterende DIP som hentes ut for oppdatering - med mindre en oppdatert DIP-versjon ikke skal lagres i digitalt depot. Forvaltningssystemet skal gi oversikt over objekter som er kopiert fra digitalt depot til sentralt arbeidsområde. Systemet skal også gi oversikt over uthentede objekter fra digitalt depot som er kopiert (eksportert) ut av sentralt arbeidsområde

Krav til forvaltningssystemet

F.1 Objekter (AIP, AIU, AIC og DIP) skal kunne hentes fra digitalt depot ved å kopieres til kontrollert og beskyttet lagringsområde.

F.2 Systemet skal kunne sperre for uthenting/kopiering av objekter med en bestemt sikkerhetsmerking eller klausultype.

F.3 Systemet skal kunne låse for (samtidig) uthenting av mer enn én kopi av en AIC.

F.4 Uthenting (kopiering) av objekter fra digitalt depot skal kunne logges automatisk av systemet, likeledes eksport (kopiering) av uthentede objekter til områder utenfor sentralt arbeidsområde.

F.5 Det skal kunne produseres en samlet oversikt over uthentede objekter fra digitalt depot til sentralt arbeidsområde som viser tidspunktet for kopiering, tidspunktet for sletting og hvem som har foretatt operasjonene.

F.6 Det skal kunne produseres en tilsvarende oversikt over uthentede objekter som er eksportert (kopiert) til områder utenfor sentralt arbeidsområde.

G. Søk

Funksjoner og rutiner

På grunnlag av informasjonen som er registrert i forvaltningssystemet, skal det være mulig å fremsøke en oversikt over lagrede arkivpakker i depot, og fremfinne arkivpakker for uthenting (kopiering) fra depot.

Krav til forvaltningssystemet

G.1 Det skal kunne fremsøkes en samlet oversikt over arkivpakker og avgrensede oversikter over pakker etter valgte kriterier på grunnlag av opplysningene i forvaltningssystemet.

G.2 Arkivpakker skal enkeltvis kunne fremsøkes for uthenting (kopiering) fra depot.

H. Spesielle operasjoner

Funksjoner og rutiner

*En sentral funksjon ved testing av arkivpakker vil være formatkontroll. En viktig funksjon i tilknytning til vedlikehold **ï** eventuelt også ved mottak **ï** vil være formatkonverteringer. Forvaltningssystemet forutsettes ikke å kunne støtte utførelsen av slike funksjoner, men det må kunne dokumentere operasjonene, resultatene og verifiseringen av resultatene .*

I. Rapportering

Funksjoner og rutiner

Det skal være mulig å produsere ulike rapporter om arkivobjektene i digitalt depot på grunnlag av informasjonen som forvaltningssystemet har registrert om dem.

Krav til forvaltningssystemet

I.1 Det skal kunne produseres rapporter med oversikt over:
- alle lagrede pakker i depot,
- pakker med gradert materiale,
- pakker med sensitive personopplysninger,
- forekommende dokumentformater i arkivpakker,
- utførte operasjoner ved generering og oppdatering av pakker
- utført verifisering av sjekksummer
- statistikk

J. Magasinkontroll

Funksjoner og rutiner

Det må finnes systemer for å identifisere alle lagrede arkivobjekter i digitalt depot. Objektene må kunne aksesseres for å utføre monitoreringsfunksjoner. Det må finnes funksjoner for å overvåke den samlede arkivbestanden, og bekrefte at den faktisk omfatter de arkivpakker som skal finnes der ï og bare disse. Arkivpakkenes opprettholdte integritet må være gjenstand for en aktiv monitorering. Det må eksistere en logg for kontrollen av sjekksummer. Det må også finnes tilgjengelige mekanismer for monitorering som varsler om fare når dokumentformater og tekniske metadata er i ferd med å bli teknologisk forgjengelige.

Krav til forvaltningssystemet

J.1 Det må finnes mekanismer for å identifisere og aksessere alle lagrede arkivobjekter.

J.2 Det må finnes mekanismer for å sikre at lagrede arkivobjekter bare kan endres eller slettes ved spesielle prosedyrer og med spesiell autorisasjon.

J.3 Det må finnes mekanismer for å styre hvilke operasjoner som er lovlige og mulige i forhold til de enkelte arkivpakker - basert på forvaltningssystemets opplysninger om klausuler, sikkerhetsmerker og andre tilgangsbestemmelser.

J.4 Basert på forvaltningssystemets opplysninger om forekommende dokumentformater i arkivpakker skal det kunne implementeres opplegg for en monitorering av formater.

J.5 Jf. for øvrig Del 1, punkt 18-26 om krav til lagringssystem.

K. Adgangskontroll

Funksjoner og rutiner

Det må finnes særskilte mekanismer for å styre tilgang til arkivobjekter i depot. For å ivareta behovet for konfidensialitetssikring må tilgangsstyringen også omfatte adgangen til å kopiere materiale ut av kontrollerte områder.

Krav til forvaltningssystemet

K.1 Systemet skal gi mulighet for å differensiere bruker- og tilgangsrettigheter

K.2 Mekanismer for validering og logging av tilgang skal kunne implementeres på objekt-nivå, og være knyttet til den enkelte arkivpakke.

K.3 Kopiering av informasjonsinnhold ut av kontrollerte områder skal kunne logges særskilt.