

## zvdd/DFG-Viewer METS-Profil – Version 2.0

---

XML Version des METS-Profiles: [http://dfg-viewer.de/fileadmin/groups/dfgviewer/METS\\_Anwendungsprofil\\_2.0.xml](http://dfg-viewer.de/fileadmin/groups/dfgviewer/METS_Anwendungsprofil_2.0.xml)

---

### Zusammenfassung

Dieses METS Profil beschreibt das Datenformat für den DFG-Viewer und definiert darüber hinausgehende Erweiterungen für das zvdd-Portal. Dokumente, die diesem Profil entsprechen, können sowohl durch den DFG-Viewer angezeigt als auch durch das zvdd-Portal verarbeitet und indexiert werden. Zu beachten ist, dass sämtliche enthaltene Beispiele das jeweilige METS-Dokument nur ausschnittsweise wiedergeben. Einige komplette Beispiele finden sich in der XML-Version des Profils am Ende des Abschnitts <Appendix>.

### URI

[http://dfg-viewer.de/fileadmin/groups/dfgviewer/METS\\_Anwendungsprofil\\_2.0.pdf](http://dfg-viewer.de/fileadmin/groups/dfgviewer/METS_Anwendungsprofil_2.0.pdf)

### Datum

2009-04-16T12:00:00

### Kontakt

**Stefan E. Funk**

[funk@sub.uni-goettingen.de](mailto:funk@sub.uni-goettingen.de)

Niedersächsische Staats- und Universitätsbibliothek

Papendiek 14, 37073 Göttingen, Germany

### Weitere verknüpfte/benutzte Profile

Dieses Profil ist eigenständig; keine anderen METS-Profile sind mit diesem Profil verknüpft.

### Benutzte Extension Schema

Metadata Object and Description Schema (MODS)

<http://www.loc.gov/standards/mods/>

## Description Rules

### Kontrollierte Vokabularien

#### ISO 639-2

International Standard Organization

<http://www.iso.ch/>

Sämtliche Sprachangaben innerhalb des MODS-Extension Schemas erfolgen gemäß ISO 639-2 Sprachcodes. Dies betrifft das Element `<mods:languageTerm>`.

#### zvdd Strukturdatentypologie

zvdd - Zentrales Verzeichnis Digitalisierter Drucke

[http://zvdd.gdz-cms.de/dokumentation/datenformate\\_im\\_ueberblick/strukturdaten/](http://zvdd.gdz-cms.de/dokumentation/datenformate_im_ueberblick/strukturdaten/)

Für zvdd verfügen sämtliche `<div>` über ein TYPE Attribut, dessen Wert der Strukturdatentypologie entsprechen muss.

#### zvdd MODS Anwendungsprofil

zvdd - Zentrales Verzeichnis Digitalisierter Drucke

[http://dfg-viewer.de/fileadmin/groups/dfgviewer/MODS\\_Anwendungsprofil\\_1.0.pdf](http://dfg-viewer.de/fileadmin/groups/dfgviewer/MODS_Anwendungsprofil_1.0.pdf)

Das zvdd MODS Anwendungsprofil muss mit dem zvdd/DFG-Viewer METS Profil verwendet werden.

## Strukturelle Anforderungen

### Deskriptive Metadaten-Sektion

#### dmdSec Anforderung 1: Anzahl und Art der deskriptiven Metadaten

Jedes Strukturelement (das heißt jedes `<div>`) kann eine oder mehrere deskriptive Metadatensektionen `<dmdSec>` besitzen. Der Typ einer Metadatensektion muss im MDTYPE Attribut eines jeden `<mdRef>` oder `<mdWrap>` Elements angegeben sein. Sowohl der DFG-Viewer als auch das zvdd-Portal unterstützen lediglich deskriptive Metadatensektionen vom Typ MODS. Diese müssen in das METS Dokument eingebunden sein und sich innerhalb von `<mdWrap>` befinden. Ferner wird lediglich die erste Metadatensektion des Typs berücksichtigt. Sollten also mehrere MODS Sektionen existieren, wird lediglich die Sektion berücksichtigt, dessen Identifier an erster Stelle im entsprechenden DMDID Attribut des `<div>` Elements steht.

Das oberste `<div>` Element aus der logischen `<structMap>` muss einen entsprechenden MODS Metadatenatz besitzen, da dieser unter anderem Informationen zur eindeutigen Identifizierung und Indexierung der Resource sowie zur Anzeige in entsprechenden Page-Turnern – wie dem DFG-Viewer – enthält. Handelt es sich beim obersten Element um ein `<div>` eines übergeordneten Werkes (beispielsweise eine Zeitschrift), das keinen MODS Metadatenatz referenziert, muss das Kind-Element einen solchen besitzen.

Weitere Metadatensektionen können vorhanden sein, um bspw. weitere Metadatenformate wie DublinCore aufzunehmen oder aber mittels `<mdRef>` auf den entsprechenden Metadatenatz im lokalen OPAC zu verweisen. Sowohl der DFG-Viewer als auch das zvdd-Portal

ignorieren entsprechende Sektionen.

### dmdSec Anforderung 2: MODS Metadaten für logische Objekte

MODS Metadaten können aufgrund ihres Detaillierungsgrades für die Indexierung eines Dokuments – konkret eines jeden <div> Elements – genutzt werden. Dies entspricht im wesentlichen der Verwendung durch das zvdd-Portal.

Die MODS Metadaten des obersten <div> Elements (bzw. die des ersten Kind-Elements) werden durch den DFG-Viewer zur Anzeige der bibliographischen Daten genutzt. Informationen über die MODS Felder, deren Nutzung und deren Vorhandensein (Pflichtfelder, optionale Felder) finden sich im separaten MODS Profil.

### dmdSec Anforderung 3: Eindeutige Identifizierung des Dokuments mit <mods:identifizier>

Jedes Dokument bzw. jede Dokumentstruktur wird mittels des Wertes, welches im <mods:identifizier> Element gespeichert wird, eindeutig identifiziert. Es können beliebige, unterschiedliche Identifier für ein <div> Element vorhanden sein. So können neben lokalen bibliotheksinternen Identifiern auch allgemein gebräuchliche Identifier wie bspw. die ISBN in diesem Feld gespeichert werden. Für das oberste <div>-Element (bzw. für dessen erstes Kind-Element) in der logischen <structMap> ist das Vorhandensein eines entsprechenden Identifiers Pflicht.

zvdd fordert an dieser Stelle einen weltweit eindeutigen Identifier wie z.B. URN, PURL, DOI, Handle oder ARK. Für den Kontext der digitalen Bibliothek hat die URN in den vergangenen Jahren an Bedeutung gewonnen. Daher wird die URN als persistenter Identifier vom zvdd-Portal empfohlen.

Der DFG-Viewer benötigt für die Anzeige eines Werkes keine solche URN. Es wird trotzdem empfohlen eine solche URN in den MODS Metadaten zu speichern, da anhand eines solchen Identifier weitere Services denkbar sind, die mittels der URN angesprochen werden können. Sie könnte bspw. zur Verknüpfung zwischen DFG-Viewer und dem zvdd-Portal dienen.

Aufgrund der Eindeutigkeit kann die URN auch als Identifier für OAI-PMH genutzt werden (Protokoll für Metadata-Harvesting). Das heißt der am zvdd-Portal aufgesetzte OAI-Harvester bedient sich der URN, um die entsprechenden OAI-Records eindeutig zu identifizieren. Dabei wird jedes <div> Element, welche eine URN besitzt als eigenständiger OAI-Record betrachtet.

#### Beispiel 1: Identifizierung eines <div>

```
<mets:mets>
  <mets:dmdSec ID="DMD_00">
    <mets:mdWrap MDTYPE="MODS">
      <mets:xmlData>
        <!-- Der MODS-spezifische Metadatenblock beginnt hier -->
        <mods:mods>
          <!-- Eindeutige Identifizierung des entsprechenden <div> Elements -->
```

```

        <mods:identifier type="urn">urn:nbn:gbv-7-PPN481399712</mods:identifier>
      </mods:mods>
    </mets:xmlData>
  </mets:mdWrap>
</mets:dmdSec>

<!-- Logische Struct-Map -->
<mets:structMap TYPE="LOGICAL">
  <mets:div DMDID="DMD_00" TYPE="Monograph" />
</mets:structMap>
</mets:mets>

```

#### dmdSec Anforderung 4: Hierarchische Verknüpfung von Dokumenten mittels MODS

Neben der gängigen METS-internen Möglichkeit, Teile von Dokumenten hierarchisch miteinander zu verknüpfen (siehe Erläuterungen zur optionalen logischen Dokumentstruktur), existieren auch hierarchische Verknüpfungen zwischen Dokumenten. Ein Beispiel für eine solche Verknüpfung ist die Beziehung zwischen einem Mehrbändigen Werk und seinen einzelnen ihm untergeordneten Bänden. Üblicherweise werden sämtliche Bände eigenständig digitalisiert, so dass für jeden der Bände eine eigene METS-Datei generiert wird. Um die Bände zu gruppieren und deren Zusammengehörigkeit abzubilden, wird für das Mehrbändige Werk eine separate METS-Datei erzeugt. Diese enthält keine Verweise auf Image-Dateien, sondern nur auf die untergeordneten Bände. Diese hierarchische Verknüpfung gilt analog für alle anderen <div> Elemente, die auf ein übergeordnetes Element verweisen – also bspw. auch für eine Verknüpfung zwischen Zeitschriftenband und Zeitschrift.

In zvdd wird eine solche Verknüpfung in MODS abgelegt. Das <relatedItem> Element vom Typ "host" speichert einen Verweis auf das übergeordnete Werk. Dazu enthält es ein Unterelement vom Typ <mods:recordInfo><mods:recordIdentifier>, das einen zur Laufzeit innerhalb des liefernden Repositories gültigen Identifier enthält. Persistenz wird nicht gefordert. Hierbei ist zu beachten, dass eine solche Verknüpfung an dieser Stelle grundsätzlich immer nur auf die übergeordnete Einheit erfolgt.

Der DFG-Viewer nutzt für die Navigation durch hierarchische Strukturen jedoch METS-Pointer <mptr>, die jeweils direkt auf über- und auch auf untergeordnete METS-Dateien verweisen, so kann in beide Richtungen navigiert werden. Beispielsweise kann von einem Zeitschriften-Element auf alle Bände verwiesen werden und von den einzelnen Bänden zum übergeordneten Zeitschriften-Element, siehe auch Kapitel "Behandlung von Zeitschriften und Mehrbändigen Werken im DFG-Viewer".

#### Beispiel 2: Hierarchische Verknüpfung zweier Dokumente mittels MODS - nachfolgendes und übergeordnetes Element

```

<!-- Nachfolgendes Element -->
<mets:mets>
  <mets:dmdSec ID="DMD_01">
    <mets:mdWrap MDTYPE="MODS">

```

```

<mets:xmlData>
  <mods:mods>
    <!-- Eindeutige Identifizierung des <div> Elements -->
    <mods:identifier type="urn">urn:nbn:gbv-7-PPN481399712</mods:identifier>
    <!-- Identifizierung des Elements -->
    <mods:recordInfo>
      <mods:recordIdentifier source="gbv-ppn">PPN481399712</mods:recordIdentifier>
    </mods:recordInfo>
    <!-- Verknüpfung zum übergeordneten Element -->
    <mods:relatedItem type="host">
      <mods:recordInfo>
        <mods:recordIdentifier source="gbv-ppn">PPN515678759</mods:recordIdentifier>
      </mods:recordInfo>
    </mods:relatedItem>
  </mods:mods>
</mets:xmlData>
</mets:mdWrap>
</mets:dmdSec>

<!-- Logische Struct-Map -->
<mets:structMap TYPE="LOGICAL">
  <mets:div DMDID="DMD_01" />
</mets:structMap>
</mets:mets>

<!-- Übergeordnetes Element -->
<mets:mets>
  <mets:dmdSec ID="DMD_02">
    <mets:mdWrap MDTYPE="MODS">
      <mets:xmlData>
        <mods:mods>
          <!-- Eindeutige Identifizierung des <div> Elements -->
          <mods:identifier type="urn">urn:nbn:gbv-7-PPN515678759</mods:identifier>
          <!-- Identifizierung des Elements -->
          <mods:recordInfo>
            <mods:recordIdentifier source="gbv-ppn">PPN515678759</mods:recordIdentifier>
          </mods:recordInfo>
        </mods:mods>
      </mets:xmlData>
    </mets:mdWrap>
  </mets:dmdSec>

  <!-- Logische Struct-Map -->
  <mets:structMap TYPE="LOGICAL">
    <mets:div DMDID="DMD_02" />
  </mets:structMap>
</mets:mets>

```

## dmdSec Anforderung 5: Reihenfolge der Strukturelemente in der logischen Struktur

Innerhalb eines METS Dokuments wird die Reihenfolge der <div> Elemente durch deren Anordnung in der logischen <structMap> bestimmt. Im Fall einer Verknüpfung mittels <relatedItem> muss die entsprechende Information jedoch an anderer Stelle im MODS Datensatz abgespeichert werden, da für diese Ebene keine verschachtelten <div> Elemente existieren. Vielmehr sind die einzelnen <div> Elemente in unterschiedlichen METS-Dokumenten abgelegt.

Unter MODS dient das <mods:part> Element dazu, konkrete Informationen über einen einzelnen Teil und dessen Zugehörigkeit zu einer größeren Gesamtheit zu speichern. In einer METS-Datei für einen Band sollte an dieser Stelle die Bandnummer sowohl für die Anzeige als auch für die Sortierung des Inhaltsverzeichnisses gespeichert werden. Diese Informationen werden sowohl durch den DFG-Viewer (Anzeige von Bandnummer als bibliographische Information) als auch durch das zvdd-Portal (Sortierung des Inhaltsverzeichnis des Mehrbändigen Werks) genutzt.

Das <mods:part> Element muss über ein ORDER Attribut verfügen, das die Reihenfolge des jeweiligen <div>s innerhalb der übergeordneten Einheit angibt. Der Wert dieses Elements muss ganzzahlig sein. Informationen zur Anzeige der Bandnummer werden innerhalb des <part> Elements unter <detail><number> gespeichert. Das <number> Element enthält sämtliche Informationen als Text, das heißt <number> darf bspw. Zahlenwörter oder sonstige beliebige Angaben zur Nummerierung enthalten.

Das <mods:detail> Element muss über ein TYPE Attribut verfügen, das für zvdd einem Typ der zvdd Strukturdatentypologie entsprechen muss. Für den DFG-Viewer sollte es einem der in der MODS Dokumentation der Library of Congress vorgeschlagenen Typen entsprechen: volume, part, issue, chapter, section, paragraph oder track.

Innerhalb des <part> Elements müssen immer beide Felder (ORDER Attribut und <number> Element) vorhanden sein. Sollte die Information zur Anzeige identisch mit denen zur Sortierung sein, müssen entsprechende Informationen dupliziert und in beiden Feldern angegeben werden.

### Beispiel 3: Verweis auf die Bandnummer des übergeordneten Werkes

```
<mets:mets>
  <mets:dmdSec ID="DMD_03">
    <mets:mdWrap MDTYPE="MODS">
      <mets:xmlData>
        <mods:mods>
          <!-- Eindeutige Identifizierung des entsprechenden <div> Elements -->
          <mods:identifier type="urn">urn:nbn:gbv-7-481399712</mods:identifier>
          <!-- Verknüpfung zum übergeordneten Element -->
          <mods:relatedItem type="host">
            <mods:recordInfo>
              <mods:recordIdentifier source="gbv-ppn">PPN515678759</mods:recordIdentifier>
            </mods:recordInfo>
          </mods:relatedItem>
          <!-- Einordnung innerhalb des übergeordneten Elements -->
```

```

        <mods:part order="80">
          <mods:detail type="volume">
            <mods:number>Achte Theil</mods:number>
          </mods:detail>
        </mods:part>
      </mods:mods>
    </mets:xmlData>
  </mets:mdWrap>
</mets:dmdSec>

<!-- Logische Struct-Map -->
<mets:structMap TYPE="LOGICAL">
  <mets:div DMDID="DMD_03" />
</mets:structMap>
</mets:mets>

```

## Administrative Metadaten-Sektion

### amdSec Anforderung 1: Metadaten zu Rechteinhaber und Urheber

Um im zvdd-Portal sowie bei der Anzeige durch den DFG-Viewer den Rechteinhaber bzw. den Urheber eines Digitalisats zu erwähnen, muss diese Information innerhalb eines <rightsMD> Element gespeichert werden. Innerhalb dieses Elements kommt ein eigenes Schema zur Anwendung, welches weiter unten erläutert wird.

Da entsprechende Urheber-/Rechteinformationen in aller Regel für eine komplette Resource gelten, kann die entsprechende <amdSec> nur von dem obersten logischen <div> Element (oder von dessen ersten Kind-Element) einer METS-Datei verknüpft werden. Das <amdSec> Element muss daher also über ein Attribut ID verfügen. Das <rightsMD> muss gemäß METS-Spezifikation zwar vorhanden sein, wird jedoch nicht aus dem ADMID Attribut des <div> Elements heraus verlinkt.

Die entsprechenden Rechte-Metadaten müssen innerhalb des <mdWrap> Elements gespeichert sein. Eine Referenzierung der Metadaten ist nicht zulässig. Das MDTYPE Attribut des <mdWrap> Elements muss den Wert "OTHER" besitzen. Dieser Typ wird im OTHERMDTYPE Attribut konkret spezifiziert. Dieses muss den Wert "DVRIGHTS" besitzen.

Innerhalb des <mdWrap> Elements kommt ein eigenes Rechte-Schema zur Anwendung, welches seinen Inhalt in ein umschließendes <dv:rights> Element schreibt. Innerhalb dieses Elements gibt es folgende drei Unterelemente, die jeweils nur genau einmal vorkommen dürfen und müssen.

owner — Der Urheber des Digitalisats

logo — Eine URL des Logos des Urhebers; dieses Logo wird durch den DFG-Viewer sowie das zvdd-Portal entsprechend angezeigt.

homepage — Eine URL der Homepage des Urhebers.

Weitere Rechteinformationen können in separaten <rightsMD> Elementen innerhalb derselben Administrativen Metadatensektion abgelegt

werden. Diese müssen jedoch andere Rights-Schemata verwenden und diese in den Attributen MDTYPE und OTHERMDTYPE des <mdWrap> Elements dokumentieren.

#### Beispiel 4: Metadaten zu Rechteinhaber und Urheber

```
<mets:mets>
  <mets:amdSec ID="ex04__AMD_00">
    <!-- rightsMD Sektion -->
    <mets:rightsMD ID="ex04__RIGHTSMD_00">
      <mets:mdWrap MDTYPE="OTHER" OTHERMDTYPE="DFGRIGHTS">
        <mets:xmlData>
          <!-- Hier wird der DFG-Namespace definiert;
               momentan existiert noch kein Schema dafür -->
          <dv:rights xmlns:dv="http://dfg-viewer.de/">
            <dv:owner>Digitalisierungszentrum der Niedersächsischen Staats- und
              Universitätsbibliothek Göttingen</dv:owner>
            <dv:ownerLogo>http://gdz.sub.uni-goettingen.de/logo_gdz_dfgv.png</dv:ownerLogo>
            <dv:ownerSiteURL>http://gdz.sub.uni-goettingen.de</dv:ownerSiteURL>
          </dv:rights>
        </mets:xmlData>
      </mets:mdWrap>
    </mets:rightsMD>
    <!-- digiprovMD Sektion siehe unten -->
  </mets:amdSec>

  <!-- Logische Struct-Map -->
  <mets:structMap TYPE="LOGICAL">
    <mets:div ADMID="ex04__AMD_00" TYPE="Monograph" />
  </mets:structMap>
</mets:mets>
```

#### amdSec Anforderung 2: Metadaten zu Herkunft und Online-Präsentation

Informationen zu Herkunft und Präsentation eines Digitalisats werden innerhalb eines <digiprovMD> Elements in der <amdSec> gespeichert.

Analog zu den Metadaten zu Rechteinhaber und Urheber müssen die entsprechenden Herkunfts-Metadaten innerhalb des <mdWrap> Elements gespeichert sein. Eine Referenzierung der Metadaten ist ebenfalls nicht zulässig. Das MDTYPE Attribut des <mdWrap> Elements muss den Wert "OTHER" besitzen. Dieser Typ wird im OTHERMDTYPE Attribut konkret spezifiziert. Dieses muss hier den Wert "DVLINKS" besitzen.

Innerhalb des <mdWrap> Elements kommt ein eigenes Herkunfts-Schema zur Anwendung, welches seinen Inhalt in ein umschließendes



<dv:links> Element schreibt. Innerhalb dieses Elements gibt es folgende zwei Unterelemente, die jeweils nur genau einmal vorkommen dürfen und müssen.

reference — Eine URL auf den Katalogeintrag des Digitalisats

presentation — Eine URL auf die Online-Präsentation des Digitalisats.

Weitere Herkunftsinformationen können in separaten <digiprovMD> Elementen innerhalb derselben Administrativen Metadatensektion abgelegt werden. Diese müssen jedoch andere Links-Schemata verwenden und diese in den Attributen MDTYPE und OTHERMDTYPE des <mdWrap> Elements dokumentieren.

#### Beispiel 5: Metadaten zu Herkunft und Online-Präsentation

```
<mets:mets>
  <mets:amdSec ID="ex05__AMD_00">
    <!-- rightsMD Sektion siehe oben -->
    <!-- digiprovMD Sektion -->
    <mets:digiprovMD ID="ex05__DIGIPROVMD_00">
      <mets:mdWrap MDTYPE="OTHER" MIMETYPE="text/xml" OTHERMDTYPE="DVLINKS">
        <mets:xmlData>
          <!-- Hier wird der DFG-Namespace definiert;
               momentan existiert noch kein Schema dafür -->
          <dv:links xmlns:dv="http://dfg-viewer.de/">
            <dv:reference>http://opac.sub.uni-goettingen.de/DB=1/PPN?PPN=394930762</dv:reference>
            <dv:presentation>http://resolver.sub.uni-goettingen.de/purl?PPN394930762</dv:presentation>
          </dv:links>
        </mets:xmlData>
      </mets:mdWrap>
    </mets:digiprovMD>
  </mets:amdSec>

  <!-- Logische Struct-Map -->
  <mets:structMap TYPE="LOGICAL">
    <mets:div AMDID="ex05__AMD_00" TYPE="Periodical" />
  </mets:structMap>
</mets:mets>
```

## File-Sektion

### fileSec Anforderung 1: File-Section

In der File-Section <fileSec> sind alle Inhaltsdateien aufgeführt, aus dem das Dokument besteht. Inhaltsdateien sind Dateien, die den

semantischen Inhalt der Resource enthalten. Dateien, die bspw. Metadaten zu diesem Dokument enthalten, werden nicht aus der File-Sektion heraus verlinkt.

### fileSec Anforderung 2: File-Groups

Die Dateien selber können in unterschiedliche Gruppen gegliedert sein. Jede Gruppe beinhaltet Dateien ähnlichen Typs zu ähnlichen Verwendungszwecken. In einem METS-Dokument muss mindestens eine Gruppe existieren. Die Anzahl der Gruppen ist nicht beschränkt. Für die Nutzung der METS-Datei im DFG-Viewer sind jedoch mindestens zwei der unten stehenden Dateigruppen zu implementieren.

Jede File-Group wird durch ein `<fileGrp>` Element deklariert und ist direkt dem `<fileSec>` Element untergeordnet. Untergruppen sind nicht möglich, das heißt ein `<fileGrp>` Element darf keine weiteren `<fileGrp>` Elemente enthalten.

Existieren mehrere File-Groups, so ist jedes `<fileGrp>` Element mit einem Attribut `USE` zu versehen, das Informationen über die Verwendung enthält.

### fileSec Anforderung 3: Files

Jede Datei wird durch ein `<file>` Element deklariert. Dieses `<file>` Element ist Kind-Element eines `<fileGrp>` Elements, das heißt jede Datei muss zu genau einer Gruppe gehören. Die Zugehörigkeit einer Datei zu mehreren Gruppen ist nicht möglich.

Der eigentliche Inhalt der Datei (der Bytestream) wird außerhalb des METS-Dokuments gespeichert, jedoch so persistent mit der METS-Datei mittels `<FLocat>` verknüpft, dass der DFG-Viewer die Datei bei Bedarf aus dem ursprünglichen Repository laden und anzeigen kann. Dazu muss das `<file>` Element als einziges Kind-Element ein `<FLocat>` Element enthalten, welches mittels einer URL, die in dem Attribut `xlink:href` gespeichert ist, referenziert wird. Das Attribut `LOCTYPE` muss daher den Wert "URL" besitzen. Für die Gültigkeit der in der METS-Datei enthaltenen URL ist das Repository verantwortlich, welche die METS-Datei produziert und in Umlauf gebracht hat. In die METS-Datei eingebetteter Content unter Verwendung des `<FContent>` Elements wird nicht unterstützt.

Jedes `<file>` Element enthält ein Attribut `ID`, welches als eindeutiger Verweis innerhalb des METS-Dokuments dient. Diese IDs werden von den `<fptr>` Elementen in `<structMap>` referenziert und weisen den `<div>` Elementen das entsprechende `<file>` Element zu.

Jedes `<file>` muss ein `MIMETYPE` Attribut beinhalten, welches den Mime-Type der Inhaltsdatei enthält. Als weitere technische Metadaten sollten `CHECKSUM`, `CHECKSUMTYPE` und `SIZE` vorhanden sein. Sie beinhalten einen Hashwert des Content-Files bzw. entsprechende Informationen zu dem verwendeten Algorithmus sowie die Länge der Datei in Bytes. Diese Informationen sind vor allem für den DFG-Viewer hilfreich, um die Authentizität der gelieferten Contentdateien vor der Anzeige zu beurteilen.

Derzeit werden keine technischen Metadaten für einzelne Dateien unterstützt. Sollten dies gewünscht sein, können sie in einer separaten administrativen Metadatensektion (`admSec`) abgespeichert werden und mittels des `ADMID` Attributs des `<file>` Elements verknüpft werden. Die Verwendung von technischen Metadaten ist also optional.

### Beispiel 6: Definition von Inhaltsdateien in verschiedenen File-Groups

```

<mets:mets xmlns:xlink="http://www.w3.org/1999/xlink">
  <mets:fileSec>

    <!-- Enthält alle Images in der beim Start des Viewers angezeigten Version -->
    <mets:fileGrp USE="DEFAULT">
      <mets:file ID="ex06__FILE00_DEF" MIMETYPE="image/tiff" SIZE="43630">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/default/image/00.tif" />
      </mets:file>
      <mets:file ID="ex06__FILE01_DEF" MIMETYPE="image/tiff" SIZE="63235">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/default/image/01.tif" />
      </mets:file>
      <mets:file ID="ex06__FILE02_DEF" MIMETYPE="image/tiff" SIZE="225434">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/default/image/02.tif" />
      </mets:file>
    </mets:fileGrp>

    <!-- Enthält alle Images in einer kleineren Version. Diese fileGrp muss genauso
    viele Images enthalten wie alle anderen fileGrps -->
    <mets:fileGrp USE="MIN">
      <mets:file ID="ex06__FILE00_MIN" MIMETYPE="image/png" SIZE="2356">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/smaller/image/00.png" />
      </mets:file>
      <mets:file ID="ex06__FILE01_MIN" MIMETYPE="image/png" SIZE="3976">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/smaller/image/01.png" />
      </mets:file>
      <mets:file ID="ex06__FILE02_MIN" MIMETYPE="image/png" SIZE="6472">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/smaller/image/02.png" />
      </mets:file>
    </mets:fileGrp>

    <!-- Die weiteren (optionalen) fileGrps folgen hier -->
  </mets:fileSec>

  <!-- Physische Struct-Map -->
  <mets:structMap TYPE="PHYSICAL">
    <mets:div TYPE="physSequence" />
  </mets:structMap>
</mets:mets>

```

#### fileSec Anforderung 4: Dateigruppen

Die Verwendung unterschiedlicher Dateigruppen hat den Zweck, Dateien mit ähnlichen Merkmalen unter dem Aspekt identischer Nutzung zusammenzufassen. Auf den DFG-Viewer übertragen bedeutet dies, dass Imagedaten mit identischer Imagebreite zu einer Gruppe zusammengefasst werden. Hierbei ist zu beachten, dass alle Dateien dieser Gruppen Derivate der Originaldateien sind, die zur Anzeige in verschiedenen Größen berechnet wurden. Daher muss eine Dateigruppe auch immer ein komplettes Set an Imagedaten enthalten – für

jede Seite muss in jeder Dateigruppe genau ein Image existieren.

Der DFG-Viewer kann diese unterschiedlichen Derivate entsprechend ihrer Imagebreite ordnen. Entsprechend der vorhandenen Imagebreiten stellt der DFG-Viewer die Images in unterschiedlichen Zoomstufen dar. Zu diesem Zweck gibt das Attribut USE des <fileGrp> Elements die Pixelbreite der enthaltenen Images an. Die möglichen Werte des USE Attributs sind für den DFG-Viewer standardisiert. Dateigruppen mit abweichenden Werten werden vom DFG-Viewer ignoriert und können so bspw. Verweise auf Volltextdaten oder lokale Dateien enthalten. Die möglichen Werte für den DFG-Viewer sind:

DEFAULT — enthält Imagedateien mit einer Breite zwischen 1000 und 1500 Pixel. Dies ist die Bildgröße, die beim ersten Aufruf des Dokuments im DFG-Viewer angezeigt wird.

MIN — enthält Imagedateien mit einer Breite zwischen 600 und 1000 Pixel. Diese Bildgröße wird angezeigt, wenn im DFG-Viewer aus dem Dokument herausgezoomt wird.

MAX — enthält Imagedateien der größtmöglichen online verfügbaren Auflösung. Diese Bildgröße wird angezeigt, wenn im DFG-Viewer in das Dokument hineingezoomt wird.

THUMBS — enthält alle Seiten als kleines Übersichtsbild mit genau 150 Pixel Breite oder 150 Pixel Höhe. Bei Hochkant-Bildern ist also die Höhe auf 150 Pixel beschränkt, bei Querformat-Bildern dagegen die Breite. Die jeweils andere Größe sollte so gewählt werden, dass die Proportionen erhalten bleiben. Das Bildformat für die Thumbnails muss entweder PNG oder JPG sein. Daraus wird der DFG-Viewer zukünftig eine Übersicht aller Seiten erstellen. Daher ist es wichtig, dass diese <fileGrp> ein entsprechendes verkleinertes Derivat eines jeden Seitenimages enthält.

DOWNLOAD — enthält Dateien, die für den Download vorgesehen sind. Zu beachten ist hier die korrekte Angabe des entsprechenden MIMETYPE Attributs der <file> Elemente, bei einer PDF-Datei muss beispielsweise der Wert "application/pdf" angegeben sein. Die einzelnen Dateien können sowohl physischen wie auch logischen Strukturelementen zugeordnet werden.

Es müssen mindestens die entsprechenden File-Groups für die Auflösungen "MIN" und "DEFAULT" in der METS-Datei enthalten sein, alle weiteren File-Groups sind optional.

#### Beispiel 7: Die vom DFG-Viewer unterstützten <FileGrp> Elemente

```
<mets:mets xmlns:xlink="http://www.w3.org/1999/xlink">
  <mets:fileSec>

    <!-- Enthält alle Images in der beim Start des DFG-Viewers angezeigten Version -->
    <mets:fileGrp USE="DEFAULT">
      <mets:file ID="ex07__FILE00_DEF" MIMETYPE="image/jpeg" SIZE="51654">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/default/image/00.jpg" />
      </mets:file>
      <mets:file ID="ex07__FILE01_DEF" MIMETYPE="image/jpeg" SIZE="46566">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/default/image/01.jpg" />
      </mets:file>
    </mets:fileGrp>
  </mets:fileSec>
</mets:mets>
```

```

</mets:fileGrp>

<!-- Enthält alle Images in einer kleineren Version -->
<mets:fileGrp USE="MIN">
  <mets:file ID="ex07__FILE00_MIN" MIMETYPE="image/jpeg" SIZE="23630">
    <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/smaller/image/00.jpg" />
  </mets:file>
  <mets:file ID="ex07__FILE01_MIN" MIMETYPE="image/jpeg" SIZE="19233">
    <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/smaller/image/01.jpg" />
  </mets:file>
</mets:fileGrp>

<!-- Enthält alle Images in der größt möglichen Version -->
<mets:fileGrp USE="MAX">
  <mets:file ID="ex07__FILE00_MAX" MIMETYPE="image/jpeg" SIZE="643630">
    <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/bigger/image/00.jpg" />
  </mets:file>
  <mets:file ID="ex07__FILE01_MAX" MIMETYPE="image/jpeg" SIZE="591244">
    <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/bigger/image/01.jpg" />
  </mets:file>
</mets:fileGrp>

<!-- Enthält alle Images in einer kleinen Vorschau-Version -->
<mets:fileGrp USE="THUMBS">
  <mets:file ID="ex07__FILE00_THB" MIMETYPE="image/png" SIZE="8234">
    <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/thumb/image/00.png" />
  </mets:file>
  <mets:file ID="ex07__FILE01_THB" MIMETYPE="image/png" SIZE="8775">
    <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/thumb/image/01.png" />
  </mets:file>
</mets:fileGrp>

<!-- Enthält zum Download angebotenen Dateien, beispielsweise PDF- oder
TIFF-Dateien -->
<mets:fileGrp USE="DOWNLOAD">
  <mets:file ID="ex07__FILE00_DWL" MIMETYPE="application/pdf" SIZE="12057">
    <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/pdf/00.pdf" />
  </mets:file>
  <mets:file ID="ex07__FILE01_DWL" MIMETYPE="application/pdf" SIZE="13001">
    <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/pdf/01.pdf" />
  </mets:file>
</mets:fileGrp>
</mets:fileSec>

<!-- Physische Struct-Map -->
<mets:structMap TYPE="PHYSICAL">
  <mets:div TYPE="physSequence" />
</mets:structMap>
</mets:mets>

```

## Structural Map

### structMap Anforderung 1: Bibliographisches Dokumentenmodell

Wird ein Dokument gemäß des Bibliographischen Dokumentenmodells erfasst, existiert lediglich eine logische <structMap>. Daher muss das TYPE Attribut des einzigen <structMap> Elements den Wert "LOGICAL" besitzen. Die logische <structMap> enthält bei Monographien lediglich ein einziges <div> Element, welches das jeweilige Werk repräsentiert. In diesem Modell existieren keinerlei <div> Elemente als Kind-Elemente.

Demzufolge ist dem <div> mindestens ein MODS Metadatensatz zugeordnet. Entsprechend hat das DMDID Attribut des <div> Elements einen Eintrag. Das TYPE Attribut muss einen Wert aus der zvdd-Strukturdatentypologie enthalten.

Da keine entsprechenden Seiten definiert sind, ist dieses Modell für den DFG-Viewer NICHT geeignet. Es existiert lediglich als notdürftige Lösung für das zvdd-Portal, um Dokumente, die lediglich als PDF vorliegen und zu denen rudimentäre Metadaten existieren, in das Portal aufnehmen und dort indexieren zu können.

#### Beispiel 8: <structMap> beim Bibliographischen Dokumentenmodell

```
<mets:mets>
  <mets:structMap TYPE="LOGICAL">
    <mets:div DMDID="DMD_00" TYPE="Monograph" />
  </mets:structMap>
</mets:mets>
```

### structMap Anforderung 2: Seitenbasiertes Dokumentenmodell

Das Seitenbasierte Dokumentmodell ist das Minimalmodell, welches vom DFG-Viewer unterstützt wird. Es definiert einzelne Seiten, die durch den DFG-Viewer angezeigt werden können. Zu jeder dieser Seiten müssen mindestens zwei entsprechende Dateiverweise existieren (einer in die File-Group "DEFAULT" und einer in die File-Group "MIN").

Das Seitenbasierte Dokumentenmodell zeichnet sich durch zwei <structMap> Elemente aus. Ein <structMap> Element enthält die logische Struktur, dessen TYPE Attribut besitzt den Wert "LOGICAL". Das andere <structMap> Element enthält die physische Struktur, das TYPE Attribut hat den Wert "PHYSICAL". Weitere <structMap> Elemente dürfen nicht existieren. Die <structLink> Sektion muss in diesem Modell vorhanden sein und entsprechende logische und physische Strukturen verknüpfen. Dabei muss jedes <div> Element aus der physischen Struktur mindestens einem <div> Element aus der logischen Struktur direkt oder indirekt zugeordnet sein. Als Ausnahme gilt lediglich das erste logische <div> Element eines Mehrbändigen Werkes oder einer Zeitschrift, da dieses das übergeordnete Werk bezeichnet und kein physisches Strukturelement dafür existiert. Weitere Informationen zur Verknüpfung der logischen und physischen Struktur sind dem

<structLink> Abschnitt zu entnehmen.

Innerhalb des physischen <structMap> Elements wird die Seitenstruktur durch <div> Elemente wiedergegeben, die einem obersten <div> Element untergeordnet sind. Dieses oberste <div> Element umfasst die Seiten, die die bibliographische Einheit repräsentieren. Daher muss dessen TYPE Attribut immer den Wert "physSequence" besitzen.

Die darunterliegenden Strukturelemente repräsentieren die Seiten bzw. den Einband. Einband und Seiten werden auf derselben hierarchischen Ebene angesiedelt. Im Fall einer Seiten-Repräsentation enthält das jeweilige <div> Element den Wert "page" im TYPE Attribut. Eine weitere Verschachtelung bspw. zum Abbilden von Seitenbereichen wie Spalten etc. ist ebenfalls denkbar und können als den Seiten untergeordnete <div> Elemente implementiert werden. Der DFG-Viewer berücksichtigt entsprechende <div> Elemente unterhalb der Seitenebene jedoch nicht.

Jedes <div> Element innerhalb der physischen Struktur muss über ein ID Attribut verfügen, welches einen eindeutigen Wert besitzt. Die Eindeutigkeit dieses Wertes gilt für das komplette METS-Dokument.

Obwohl es auf Seitenebene sinnvoll erscheint, die <div> Elemente in derselben Reihenfolge anzuordnen, wie die Seiten innerhalb der gebundenen Einheit angeordnet sind, muss diese Reihenfolge im ORDER Attribut eines jeden <div> Elements auf Seitenebene explizit angegeben werden. Das ORDER Attribut darf lediglich einen ganzzahligen Wert (Integer) enthalten, der auf Ebene der Seiten eindeutig sein muss. Für die Reihenfolge der Seiten sind einzig die Werte der ORDER Attribute ausschlaggebend; die Reihenfolge der <div> Elemente bleibt unberücksichtigt. Für den Seiten untergeordnete Strukturelemente wird die Verwendung des ORDER Attributs ebenfalls empfohlen.

Besitzt eine Seite eine aufgedruckte Seitenzahl, so ist diese in Vorlagenform im ORDERLABEL Attribut des <div> Elements zu speichern. Seiten, die nicht in der Paginierung berücksichtigt sind (ungezählte Seiten) benötigen kein ORDERLABEL Attribut. Das Ausfüllen des LABEL Attributs ist optional möglich. Der DFG-Viewer benutzt den Wert des ORDERLABEL Attributs zur Anzeige und Navigation (gezieltes Anspringen von Seiten), wenn das Attribut vorhanden ist.

Das Attribut CONTENTIDS gibt einen (oder mehrere) persistente Identifier für jede einzelnen Seite des Dokuments an und besteht aus einer URI oder einer Liste von URIs. Diese werden vom DFG-Viewer als persistente IDs für die jeweilige Seite angezeigt und können als dauerhafter Link für Zitationen direkt auf die entsprechende Seite genutzt werden.

Mit Hilfe des File-Pointers <fptr> können Dateien für den Download referenziert werden. In der physischen Struct-Map ist dies nur für einzelne Seiten möglich, siehe auch Kapitel "Verweise auf Dateien – physische Struktur". Eine Datei des kompletten Werkes kann in der logischen Struct-Map referenziert werden, siehe Kapitel "Verweise auf Dateien – logische Struktur".

#### Beispiel 9: <structMap> der physischen Struktur

```
<mets:mets>
  <!-- Logische Struct-Map -->
  <mets:structMap TYPE="LOGICAL">
    <mets:div DMDID="DMD_00" ID="ex09__LOG_00" TYPE="Monograph" />
  </mets:structMap>
```



```

<!-- Physische Struct-Map -->
<mets:structMap TYPE="PHYSICAL">
  <mets:div ID="ex09__PHYS_00" TYPE="physSequence">
    <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-ex10__PHYS_01" ID="ex09__PHYS_01" ORDER="1" ORDERLABEL="I"
TYPE="page" />
    <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-ex10__PHYS_02" ID="ex09__PHYS_02" ORDER="2" ORDERLABEL="II"
TYPE="page" />
    <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-ex10__PHYS_03" ID="ex09__PHYS_03" ORDER="3"
ORDERLABEL="III" TYPE="page" />
    <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-ex10__PHYS_04" ID="ex09__PHYS_04" ORDER="4" ORDERLABEL="1"
TYPE="page" />
    <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-ex10__PHYS_05" ID="ex09__PHYS_05" ORDER="5" ORDERLABEL="2"
TYPE="page" />
  </mets:div>
</mets:structMap>
</mets:mets>

```

### structMap Anforderung 3: Komplexes Dokumentenmodell

Das Komplexe Dokumentenmodell erweitert das Seitenbasierte Modell um weitere Strukturinformationen auf logischer Ebene. Für die physische Struktur (<structMap TYPE="PHYSICAL">) gilt im Komplexen Dokumentenmodell all jenes, was bereits oben für das Seitenbasierte Modell beschrieben wurde.

Der DFG-Viewer nutzt die zusätzlichen Strukturinformationen, um das Inhaltsverzeichnis zu erstellen und ermöglicht eine Navigation in der Struktur. Das zvdd-Portal verarbeitet die logischen Strukturdaten und nutzt sie zur Indizierung und zur Suche.

Die oberste logische Struktureinheit ist das bereits im Bibliographischen Dokumentenmodell erwähnte <div> Element, welches die bibliographische Einheit repräsentiert. Diesem <div> werden weitere <div> Elemente untergeordnet, so dass sich durch die verschachtelten Elemente die logische Struktur des Dokuments abbildet. Die Tiefe der Verschachtelung ist nicht beschränkt. In der logischen Struktur gibt die Reihenfolge der <div> Elemente die tatsächliche Reihenfolge der zu repräsentierenden Strukturen wieder. Es ist nicht notwendig, das ORDER oder ORDERLABEL Attribut zu benutzen; deren Werte werden im Kontext des DFG-Viewers sowie des zvdd-Portals ignoriert.

Jedes <div> Element innerhalb der logischen Struktur muss ebenfalls über ein ID Attribut verfügen, welches einen eindeutigen Wert besitzt. Die Eindeutigkeit dieses Wertes gilt für das komplette METS-Dokument.

Weiterhin muss jedes <div> Element innerhalb der logischen Struktur über ein TYPE Attribut verfügen. Für zvdd muss dessen Inhalt einem Typ der zvdd Strukturdatentypologie entsprechen. Werden die Strukturtypen des DFG-Viewer Strukturdatensets verwendet, werden diese Strukturen – je nach Sprachauswahl des Nutzers – auch übersetzt angezeigt. Ist ein Attribut LABEL vorhanden, wird dieses zur Anzeige des jeweiligen Strukturtyps in der Übersicht genutzt.

Mit dem Attribut CONTENTIDS können auch in diesem Modell persistente Identifier für alle logischen Elemente angegeben werden, beispielsweise für die gesamte Monographie oder für einzelne Kapitel. Diese werden ebenfalls vom DFG-Viewer als persistente IDs für



Zitationen angezeigt.

Zusätzlich zu den Download-Referenzen in der physischen Struct-Map können auch in der logischen Struct-Map Dateien per <fptr> referenziert werden, beispielsweise PDF-Dateien des kompletten Werkes, von Artikeln oder Kapiteln. Der DFG-Viewer wertet zur Zeit nur die Referenz auf das Gesamtwerk aus, siehe auch Kapitel "Verweise auf Dateien – logische Struktur".

#### Beispiel 10: Verschachtelung der <structMap> Elemente der logische Dokumentstruktur

```
<mets:mets>
  <mets:structMap TYPE="LOGICAL">
    <!-- Logisches Strukturelement für die Monographie -->
    <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-" DMDID="DMD_00" ID="ex10__LOG_00" LABEL="Monographie"
TYPE="Monograph">

      <!-- Logisches Strukturelement für das erste Kapitel mit Unterteilung -->
      <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-ex10__LOG_01" DMDID="DMD_01" ID="ex10__LOG_01"
LABEL="Erstes Kapitel" TYPE="Chapter">
        <!-- Logisches Strukturelement für ein Inhaltsverzeichnis -->
        <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-ex10__LOG_02" ID="ex10__LOG_02" LABEL="Illustration"
TYPE="Illustration" />
        <!-- Logisches Strukturelement für eine Textpassage -->
        <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-ex10__LOG_03" ID="ex10__LOG_03" LABEL="Textabschnitt"
TYPE="TextSection" />
      </mets:div>

      <!-- Logische Strukturelemente für zwei weitere Kapitel ohne weitere
        Unterteilung -->
      <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-ex10__LOG_04" DMDID="DMD_02" ID="ex10__LOG_04"
LABEL="Zweites Kapitel" TYPE="Chapter" />
      <mets:div CONTENTIDS="urn:nbn:de:gbv-7-gdz-12345678-ex10__LOG_05" ID="ex10__LOG_05" LABEL="Drittes Kapitel"
TYPE="Chapter" />
    </mets:div>
  </mets:structMap>
</mets:mets>
```

#### structMap Anforderung 4: Behandlung von Zeitschriften und Mehrbändigen Werken im DFG-Viewer

Bandübergreifende Navigation im DFG-Viewer wird ermöglicht, indem die Bände und ihre übergeordnete Struktur mittels METS-Pointern untereinander verknüpft werden. Dies geschieht über eine weitere METS-Datei, die nur die Zeitschrift oder das Mehrbändige Werk beschreibt. Hier sind nur die Metadaten der Zeitschrift sowie eine logische <structMap> enthalten, in der alle <div> Elemente der Bände verzeichnet sind. Die Reihenfolge der <div> Elemente entspricht der Reihenfolge der Bände. Es sind keine Inhaltsdateien von dieser Datei aus verlinkt.

Die Verknüpfung zwischen den METS-Dateien der Zeitschrift und den METS-Dateien der Bände erfolgt mittels METS-Pointer Elementen `<mptr>`, die den einzelnen `<div>` Elementen untergeordnet sind. Aus der METS-Datei des Bandes weist ein METS-Pointer auf die METS-Datei der Zeitschrift und von der METS-Datei der Zeitschrift weist jeweils ein METS-Pointer auf die zugehörige METS-Datei eines jeden Bandes.

**Beispiel 11: Logische Struct-Map für einen Band und eine übergeordnete Zeitschrift (der Übersichtlichkeit halber ohne CONTENTIDS)**

```
<!-- Logische Struct-Map für einen Band -->
<mets:mets xmlns:xlink="http://www.w3.org/1999/xlink">
  <mets:structMap TYPE="LOGICAL">
    <mets:div LABEL="Zeitschrift" TYPE="Periodical">
      <!-- Der METS-Pointer referenziert die METS-Datei der Zeitschrift / des
           Mehrbändigen Werkes -->
      <mets:mptr LOCTYPE="URL" xlink:href="http://link/to/mets/file/periodical" />

      <!-- Hier beginnt die oben bereits besprochene Beschreibung des einzelnen
           Bandes -->
      <mets:div DMDID="DMD_00" ID="ex11__LOG_00" LABEL="Zeitschriftenband" TYPE="Volume">
        <mets:div DMDID="DMD_01" ID="ex11__LOG_01" LABEL="Erstes Zeitschriftenheft" TYPE="Issue">
          <mets:div ID="ex11__LOG_02" LABEL="Erster Artikel" TYPE="Article" />
          <mets:div ID="ex11__LOG_03" LABEL="Zweiter Artikel" TYPE="Article" />
        </mets:div>
        <mets:div DMDID="DMD_01" ID="ex11__LOG_04" LABEL="Zweites Zeitschriftenheft" TYPE="Issue" />
        <mets:div ID="ex11__LOG_05" LABEL="Drittes Zeitschriftenheft" TYPE="Issue" />
      </mets:div>
    </mets:div>
  </mets:structMap>
</mets:mets>

<!-- Logische Struct-Map für eine übergeordnete Zeitschrift -->
<mets:mets xmlns:xlink="http://www.w3.org/1999/xlink">
  <mets:structMap TYPE="LOGICAL">
    <mets:div DMDID="DMD_00" ID="ex11__LOG0_00" LABEL="Zeitschrift" TYPE="Periodical">
      <mets:div LABEL="Erster Zeitschriftenband" TYPE="Volume">
        <!-- Ein METS-Pointer auf den ersten Band der Zeitschrift -->
        <mets:mptr LOCTYPE="URL" xlink:href="http://link/to/mets/file/1st/volume/mets.xml" />
      </mets:div>
      <mets:div LABEL="Zweiter Zeitschriftenband" TYPE="Volume">
        <!-- Ein METS-Pointer auf den zweiten Band der Zeitschrift -->
        <mets:mptr LOCTYPE="URL" xlink:href="http://link/to/mets/file/2nd/volume/mets.xml" />
      </mets:div>
      <mets:div LABEL="Dritter Zeitschriftenband" TYPE="Volume">
        <!-- Ein METS-Pointer auf den dritten Band der Zeitschrift -->
        <mets:mptr LOCTYPE="URL" xlink:href="http://link/to/mets/file/3rd/volume/metx.xml" />
      </mets:div>
    </mets:div>
  </mets:structMap>
</mets:mets>
```

```

</mets:structMap>
</mets:mets>

```

### structMap Anforderung 5: Metadaten für Strukturelemente

Grundsätzlich können jedem <div> Element ein oder mehrere Metadatensektionen zugeordnet werden. Dies ist unabhängig davon, ob sich das Element in einer logischen oder einer physischen Struktur befindet. Sowohl das zvdd-Portal als auch der DFG-Viewer nutzen allerdings lediglich deskriptive Metadaten gemäß der in der Sektion "Deskriptive Metadaten" genannten Bedingungen (siehe MODS-Metadatenschema), die von der logischen <structMap> aus referenziert werden. Andere Metadatensektionen werden ignoriert.

### structMap Anforderung 6: Verweise auf Dateien – physische Struktur

Um von einem <div> Element auf zugehörige Dateien zu verweisen, wird das File-Pointer Element <fptr> benutzt. Dieses Element ist daher immer Kind-Element des <div> Elements, für welches es die Dateien verlinkt. Jedes <div> Element kann einen oder mehrere File-Pointer besitzen.

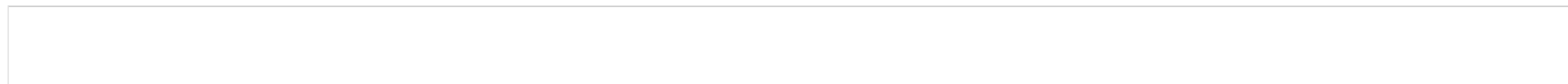
Ein File-Pointer verweist immer auf eine Datei, die in der File-Sektion an beliebiger Stelle aufgeführt ist; das heißt, die Dateien können in verschiedenen File-Groups enthalten sein. Die Verknüpfung erfolgt über das FILEID Attribut. Jedes <fptr> Element muss ein FILEID Attribut besitzen.

Dateien, die lediglich den Inhalt einer Seite wiedergeben (bspw. Seiten), werden nur aus der physischen Dokumentenstruktur heraus verlinkt – konkret: nur aus den <div> Elementen heraus, die die jeweilige Seite repräsentieren. Durch die hierarchische Struktur wird eine Verknüpfung der den Seiten zugeordneten Dateien (Seitenimages) zur gebundenen Einheit implizit ausgedrückt. Den unterliegenden Strukturen (bspw. Seiten) zugeordneten Dateien dürfen den überliegenden Strukturen nicht explizit ein weiteres Mal zugeordnet werden.

Generell dürfen nur <file> Elemente verlinkt werden, Verknüpfungen zu <fileGrp> Elemente sind unzulässig. Damit der DFG-Viewer entsprechende Seiten anzeigen kann, müssen zu jedem <div> vom Typ "page" mindestens zwei Verknüpfungen vorhanden sein. Eine Verknüpfung muss jeweils auf je eine Datei aus den beiden Pflichtdateigruppen "DEFAULT" und "MIN" erfolgen. Wenn weitere File-Groups vorhanden sind, werden auch diese hier verlinkt.

Für den DFG-Viewer können in der physischen Struct-Map Verweise auf Dateien für den Download einzelner Seiten referenziert werden, ein solcher Einzelseiten-Download wird dann entsprechend im DFG-Viewer angeboten. Ein Verweis auf einen Gesamt-Download im obersten <div> wird nur dann angezeigt, wenn kein entsprechender Verweis in der logischen Struct-Map vorhanden ist. Siehe auch Kapitel "Verweise auf Dateien – logische Struktur".

#### Beispiel 12: <structMap> der physischen Struktur



```

<mets:mets xmlns:xlink="http://www.w3.org/1999/xlink">
  <!-- Physische Struct-Map -->
  <mets:structMap TYPE="PHYSICAL">
    <mets:div TYPE="physSequence">
      <!-- Aus jeder File-Group werden hier die verschiedenen
            Auflösungen / Formate eines jeden Files referenziert -->
      <mets:div ID="ex12__PHY_00" ORDER="1" ORDERLABEL="I" TYPE="page">
        <mets:fptr FILEID="ex07__FILE00_DEF" />
        <mets:fptr FILEID="ex07__FILE00_MIN" />
        <mets:fptr FILEID="ex07__FILE00_MAX" />
        <mets:fptr FILEID="ex07__FILE00_THB" />
        <mets:fptr FILEID="ex07__FILE00_DWL" />
      </mets:div>
      <mets:div ID="PHYS09_01" ORDER="2" ORDERLABEL="II" TYPE="page">
        <mets:fptr FILEID="ex07__FILE01_DEF" />
        <mets:fptr FILEID="ex07__FILE01_MIN" />
        <mets:fptr FILEID="ex07__FILE01_MAX" />
        <mets:fptr FILEID="ex07__FILE01_THB" />
        <mets:fptr FILEID="ex07__FILE01_DWL" />
      </mets:div>
    </mets:div>
  </mets:structMap>
</mets:mets>

```

### structMap Anforderung 7: Verweise auf Dateien – logische Struktur

Ebenso wie aus der physischen Struktur können Verknüpfungen auf Dateien aus der logischen Struktur heraus gesetzt werden. Eine aus der logischen Struktur heraus verknüpfte Datei muss jedoch den kompletten Inhalt der entsprechenden logischen Dokumentstruktur enthalten. Beispiel für eine solche Verknüpfung kann bspw. der Link auf eine PDF-Datei sein, die eine komplette Monographie oder ein einzelnes Kapitel derselben enthält. Es ist nicht erlaubt, Verweise auf mehrere Dateien, die erst als Sequenz den kompletten Inhalt der logischen Struktur wiedergeben für dieses <div> Element zu setzen.

Sinnvoll ist ein solcher Verweis immer dann, wenn die verknüpfte Datei nicht granular genug adressiert werden kann, um sie aus der physischen Struktur heraus zu verlinken; bspw. wenn keine entsprechenden Seitenumbrüche in der Datei enthalten sind oder diese den Download kompletter logischer Strukturen ermöglicht.

Der DFG-Viewer nutzt gezielt die Verknüpfungen von logischen Strukturelementen auf Dateien der File-Group vom Typ DOWNLOAD, um den Download einzelner Kapitel und kompletter Werke zu ermöglichen. Ist an dieser Stelle ein Verweis auf das komplette Werk vorhanden, wird dieser vom DFG-Viewer bevorzugt als Link zum Herunterladen des gesamten Dokuments angezeigt. Die entsprechenden Dateien müssen innerhalb der <fileSec> definiert sein und dessen <file> Elemente müssen über entsprechende MIMETYPE Attribute verfügen. Pro logischer Struktureinheit wird nur eine – die erste – Verknüpfung mit einer Datei unterstützt. Eine solche Verknüpfung ist für den DFG-Viewer optional, wenn das Dokument das Seitenbasierte oder das Komplexe Dokumentenmodell implementiert. Wird das Bibliographische Dokumentenmodell implementiert, ist ein solcher Link verpflichtend, da er die einzige Verknüpfung zum Inhalt darstellt.

Inhalte dürfen entweder direkt aus der logischen Struktur heraus oder aber indirekt über eine Verknüpfung der logischen Struktur mit der physischen Struktur verknüpft werden. Eine Redundanz der Verknüpfung auf eine Datei aus beiden Strukturen heraus ist auf alle Fälle zu vermeiden. Eine Verknüpfung auf einzelne Seiten-Images für ein logisches <div> ist nicht erlaubt, wenn bereits von einem physischen Strukturelement (bspw. einer Seite) auf dasselbe Seitenimage verwiesen wird. Stattdessen ist das logische <div> Element mit der Seite über die <structLink> Sektion zu verknüpfen und von der Seite auf das entsprechende Image zu verweisen. Für das Blättern der Seiten im DFG-Viewer ist eine entsprechende Verknüpfung nicht relevant. Jedoch könnte der DFG-Viewer dem Benutzer ein Link auf das PDF im Originalrepository anbieten.

#### Beispiel 13: Struct-Map der logischen Struktur mit Verweis auf eine PDF-Datei eines Artikels

```
<mets:mets xmlns:xlink="http://www.w3.org/1999/xlink">
  <!-- File-Group als Beispiel für den File-Pointer in der logischen Struct-Map -->
  <mets:fileSec>
    <mets:fileGrp USE="DOWNLOAD">
      <!-- Verweis auf die PDF-Datei des Bandes -->
      <mets:file ID="ex13__FILE00_Monograph" MIMETYPE="application/pdf" SIZE="1643630">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/monograph.pdf" />
      </mets:file>
      <!-- Verweis auf die PDF-Datei des Kapitels -->
      <mets:file ID="ex13__FILE01_Chapter" MIMETYPE="application/pdf" SIZE="47676">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/chapter/01.pdf" />
      </mets:file>
    </mets:fileGrp>
  </mets:fileSec>

  <!-- Logische Struct-Map -->
  <mets:structMap TYPE="LOGICAL">
    <mets:div TYPE="Periodical">
      <!-- Der METS-Pointer referenziert die METS-Datei der Zeitschrift / des
      Mehrbändigen Werkes -->
      <mets:mptr LOCTYPE="URL" xlink:href="http://link/to/mets/file/periodical" />

      <!-- Hier beginnt die oben bereits besprochene Beschreibung des einzelnen
      Bandes -->
      <mets:div DMDID="DMD_00" ID="ex13__LOG_00" TYPE="Volume">
        <!-- File-Pointer auf das PDF des Bandes -->
        <mets:fptr FILEID="ex13__FILE00_Monograph" />

        <mets:div DMDID="DMD_01" ID="ex13__LOG_01" TYPE="Issue">
          <mets:div ID="ex13__LOG_02" TYPE="Article">
            <!-- File-Pointer auf das PDF eines Kapitels -->
            <mets:fptr FILEID="ex13__FILE01_Chapter" />
          </mets:div>
          <mets:div ID="ex13__LOG_03" TYPE="Article" />
        </mets:div>
      </mets:div>
    </mets:div>
  </mets:structMap>
</mets:mets>
```

```

    <mets:div DMDID="DMD_01" ID="ex13__LOG_04" TYPE="Issue" />
    <mets:div ID="ex13__LOG_05" TYPE="Issue" />
  </mets:div>
</mets:div>
</mets:structMap>
</mets:mets>

```

### structMap Anforderung 8: Mehrfache Verknüpfungen aus einer <div> Sektion

Werden mehrere Dateien aus einem <div> Element heraus verknüpft, so müssen diese Dateien denselben semantischen Inhalt beinhalten. Die Darstellungsform kann jedoch unterschiedlich sein. Dazu ist es notwendig, dass sich die verknüpften Dateien in unterschiedlichen File-Groups befinden. Somit lassen sich für ein <div> unterschiedliche Auflösungen ein und dergleichen Seite in unterschiedlichen <file> Elemente speichern und dieser Seite zuordnen.

Parallel oder in sequentieller Reihenfolge darstellbare Dateien werden nicht unterstützt, das heißt <par> und <seq> Elemente dürfen in der METS-Datei nicht enthalten sein.

### structMap Anforderung 9: Verknüpfungen in Dateien

Teilweise unterscheidet sich die Granularität des Dokuments von der Granularität der einzelnen Content-Files. So kann es sinnvoll sein, Dokumente in der logischen oder physischen Struktur feiner granuliert zu erfassen, als dies für eine Speicherung der Inhalte sinnvoll ist. Als Beispiel ist hier die Speicherung von Volltexten zu nennen. In aller Regel werden Volltexte gemäß der TEI-Spezifikation immer komplett für ein Dokument in einer XML-Datei gespeichert, das heißt der Volltext umfasst mehrere Seiten, die innerhalb des Textes markiert sind. Daher stellt METS Möglichkeiten bereit, direkt in eine solche Datei hinein zu verweisen, um den Start und das Ende des jeweiligen Inhalts des <div> Elements in der Inhaltsdatei zu markieren.

Mittels einer solchen granularen Verknüpfung kann das zvdd-Portal Volltexte seitenbasiert indexieren und den Volltext entsprechenden Strukturelementen (<div>s) zuordnen. Der DFG-Viewer könnte in späteren Versionen in der Lage sein, neben reinen Images auch Volltexte seitenbasiert dazustellen bzw. mittels solcher Information Suchtreffer in den Images zu markieren. Die Unterstützung von Volltext ist generell sowohl im DFG-Viewer als auch im zvdd-Portal optional.

Eine solche Verknüpfung erfolgt über das <area> Element, welches als Kind-Element des jeweiligen File-Pointes (<fptr> Element) existiert. Ist ein <area> Element vorhanden, so wird von diesem Element mittels des FILEID Attributs auf die Inhaltsdatei verwiesen. In diesem Fall darf das <fptr> Element selber kein FILEID Element besitzen.

Als erste Verweisart wird der Verweis in Imagebereiche unterstützt. Hierbei enthält der Verweis Pixelkoordinaten, um einen Bereich innerhalb des referenzierten Images zu markieren, welcher den Inhalt des jeweiligen <div> Elements wiedergibt. Ein solcher Verweis muss folgende Attribute für das <area> beinhalten:

SHAPE — Die Form des Bereiches: "RECT", "CIRCLE" oder "POLY".

COORDS — Die Koordinateninformation des Bereichs gemäß HTML 4.0.

Als zweite Verweisart wird der Verweis in XML-Dateien hinein unterstützt. Ein solcher Verweis wird mittels ID Attributen durchgeführt. Das heißt in der Ziel-Datei müssen XML-Elemente existieren, die über ID Attribute mit den angegebenen Werten verfügen. In diesem Fall muss das <area> Element die Attribute BETYPE mit dem Wert "IDREF" enthalten und die Attribute BEGIN und END müssen die jeweiligen Werte der ID Attribute der Volltextdatei enthalten. Soll nur auf ein einzelnes Element verwiesen werden, so enthalten die BEGIN und END Attribute denselben Wert.

Andere Verweisarten bspw. über Timecode oder Binär-Offsets dürfen in der METS-Datei nicht vorkommen.

#### Beispiel 14: Verweis in eine Datei mittels <mets:area>

```
<mets:mets xmlns:xlink="http://www.w3.org/1999/xlink">
  <mets:fileSec>
    <!-- Eine beispielhafte fileGrp zur Nutzung von <mets:area> -->
    <mets:fileGrp USE="AREA">
      <mets:file ID="ex14__FILE00_ARE" MIMETYPE="text/xml" SIZE="6523">
        <mets:FLocat LOCTYPE="URL" xlink:href="http://link/to/xml/tei/file" />
      </mets:file>
    </mets:fileGrp>
  </mets:fileSec>

  <!-- Physische Struct-Map -->
  <mets:structMap TYPE="PHYSICAL">
    <mets:div TYPE="physSequence">
      <mets:div ID="ex14__PHY_00" ORDER="3" ORDERLABEL="III" TYPE="page">
        <!-- Verweis in eine Datei hinein. Dabei wird der eigentliche
              Inhalt über die IDRef-Werte in den Attributen BEGIN und
              END definiert -->
        <mets:fptr>
          <mets:area BEGIN="TEIID_24" BETYPE="IDREF" END="TEIID_63" FILEID="ex14__FILE00_ARE" />
        </mets:fptr>
      </mets:div>
    </mets:div>
  </mets:structMap>
</mets:mets>
```

## Strukturverknüpfungen (structLink)

### structLink Anforderung 1: Verknüpfung von logischer und physischer Struktur

Jedes METS-Dokument, welches sowohl eine logische als auch eine physische <structMap> besitzt, muss eine <structLink> Sektion haben. Dies betrifft also alle METS-Dokumente, die entsprechend des Seitenbasierten sowie des Komplexen Dokumentmodells erstellt



wurden.

Die <structLink> Sektion speichert Verknüpfungen zwischen logischer und physischer Struktur. Für jede einzelne Verknüpfung wird ein eigenes <smLink> Element genutzt. Jedes dieser Elemente verfügt über xlink:from and xlink:to Attribute, welche die Werte der ID Attribute der jeweiligen <div> Elemente aus der logischen und physischen Struktur beinhalten.

Es wird immer von der logischen Struktur auf die physische Struktur verwiesen. Dies bedeutet, dass das xlink:from Attribut den ID-Wert eines <div> Elements aus der logischen Struktur enthalten muss; xlink:to beinhaltet demzufolge also den Wert des ID Attributs eines <div>s aus der physischen Struktur.

#### Beispiel 15: Verknüpfung von logischer und physischer Struktur

```
<mets:mets xmlns:xlink="http://www.w3.org/1999/xlink">
  <!-- Logische Struct-Map -->
  <mets:structMap TYPE="LOGICAL">
    <mets:div ID="ex15__LOG_00" TYPE="Monograph">
      <mets:div ID="ex15__LOG_01" TYPE="Chapter">
        <mets:div ID="ex15__LOG_02" TYPE="Chapter" />
        <mets:div ID="ex15__LOG_03" TYPE="Chapter" />
      </mets:div>
    </mets:div>
  </mets:structMap>

  <!-- Physische Struct-Map -->
  <mets:structMap TYPE="PHYSICAL">
    <mets:div ID="ex15__PHY_00" TYPE="physSequence">
      <mets:div ID="ex15__PHY_01" ORDER="1" ORDERLABEL="III" TYPE="page" />
      <mets:div ID="ex15__PHY_02" ORDER="2" ORDERLABEL="1" TYPE="page" />
      <mets:div ID="ex15__PHY_03" ORDER="3" ORDERLABEL="2" TYPE="page" />
      <mets:div ID="ex15__PHY_04" ORDER="4" ORDERLABEL="3" TYPE="page" />
      <mets:div ID="ex15__PHY_05" ORDER="5" ORDERLABEL="4" TYPE="page" />
    </mets:div>
  </mets:structMap>

  <!-- Hier werden die logischen und die physischen Strukturelemente
       verknüpft -->
  <mets:structLink>
    <!-- Verknüpfung zwischen Monograph und Physical Sequence -->
    <mets:smLink xlink:from="ex15__LOG_00" xlink:to="ex15__PHY_00" />
    <!-- Verknüpfung zwischen Kapitel eins und Seiten -->
    <mets:smLink xlink:from="ex15__LOG_01" xlink:to="ex15__PHY_02" />
    <mets:smLink xlink:from="ex15__LOG_01" xlink:to="ex15__PHY_03" />
    <mets:smLink xlink:from="ex15__LOG_01" xlink:to="ex15__PHY_04" />
    <mets:smLink xlink:from="ex15__LOG_01" xlink:to="ex15__PHY_05" />
    <!-- Verknüpfung zwischen Kapitel zwei und Seiten -->
    <mets:smLink xlink:from="ex15__LOG_02" xlink:to="ex15__PHY_03" />
  </mets:structLink>
</mets:mets>
```



```

<mets:smLink xlink:from="ex15__LOG_02" xlink:to="ex15__PHY_04" />
<!-- Verknüpfung zwischen Kapitel drei und Seiten -->
<mets:smLink xlink:from="ex15__LOG_03" xlink:to="ex15__PHY_04" />
<mets:smLink xlink:from="ex15__LOG_03" xlink:to="ex15__PHY_05" />
</mets:structLink>
</mets:mets>

```

### structLink Anforderung 2: Vererbung von Strukturverknüpfungen

Eine Verknüpfung auf eine physische Struktur bezieht sich auch immer auf alle unterliegenden Strukturelemente. Eine Verknüpfung von der obersten logischen Struktureinheit (bspw. Monographie) auf die physische Sequenz (physSequence) impliziert also auch alle Verknüpfungen auf die einzelnen Seiten, die dem physischen Sequenz untergeordnet sind. Eine explizite Verknüpfung zwischen Monographie und den einzelnen Seiten sind nicht notwendig.

Dagegen werden Verknüpfungen auf logischer Struktur NICHT vererbt. Das heißt für jedes logische <div> Element sind alle Verknüpfungen erneut wieder aufzuführen. Die Gesamtheit aller unterliegenden <div> Elemente muss jedoch nicht zwangsläufig alle Verknüpfungen des übergeordneten <div> Elements enthalten.

Dies bedeutet, dass für das Seitenbasierte Dokumentenmodell, welches vom DFG-Viewer implementiert wird, lediglich eine Verknüpfung von der obersten logischen Dokumentstruktur auf die physSequence notwendig ist. Die definierten Seiten müssen nicht aus der logischen Struktur heraus verknüpft werden. Der DFG-Viewer ist in der Lage, den impliziten Verknüpfungen zu folgen.

#### Beispiel 16: Vererbung der Verknüpfungen auf physische Strukturen für das Seitenbasierte Dokumentenmodell

```

<mets:mets xmlns:xlink="http://www.w3.org/1999/xlink">
  <!-- Logische Struct-Map -->
  <mets:structMap TYPE="LOGICAL">
    <mets:div ID="ex16__LOG_00" TYPE="Monograph" />
  </mets:structMap>

  <!-- Physische Struct-Map -->
  <mets:structMap TYPE="PHYSICAL">
    <mets:div ID="ex16__PHYS_00" TYPE="physSequence">
      <mets:div ID="ex16__PHYS_01" ORDER="1" ORDERLABEL="III" TYPE="page" />
      <mets:div ID="ex16__PHYS_02" ORDER="2" ORDERLABEL="1" TYPE="page" />
      <mets:div ID="ex16__PHYS_03" ORDER="3" ORDERLABEL="2" TYPE="page" />
      <mets:div ID="ex16__PHYS_04" ORDER="4" ORDERLABEL="3" TYPE="page" />
      <mets:div ID="ex16__PHYS_05" ORDER="5" ORDERLABEL="4" TYPE="page" />
    </mets:div>
  </mets:structMap>

```

```

<!-- Hier werden die logischen und die physischen Structurelemente
      verknüpft -->
<mets:structLink>
  <!-- Verknüpfung zwischen Monograph und physical Sequence -->
  <mets:smLink xlink:from="ex16__LOG_00" xlink:to="ex18__PHY_00" />
</mets:structLink>
</mets:mets>

```

## Technische Anforderungen

### Inhaltsdateien (Content Files)

#### Images

Alle Images, die in den vom DFG-Viewer genutzten File-Groups als Content-File referenziert werden, müssen in einem Format vorliegen, welches direkt durch einen Web-Browser darstellbar ist. Als solche Formate gelten derzeit JPG, GIF und PNG. Andere Formate werden nicht unterstützt. Eine Ausnahme bilden all jene Dateien, die in der File-Group vom Typ DOWNLOAD enthalten sind. Hierbei handelt es sich überwiegend um Dateien, die dem Nutzer zum Download angeboten werden, beispielsweise PDF.

#### Volltexte

Derzeit existiert noch keine Beschreibung eines standardisierten Formats für den Volltext zur Präsentation im DFG-Viewer oder zur Indexierung im zvdd-Portal. Entsprechend wird Volltext, so er denn als Content-File referenziert wird, durch beide Applikationen nicht berücksichtigt.