

2. Aufbau einer EAD(DDB)-Datei

EAD(DDB) ist ein XML-Format

Die archivische Erschließung ist in der Regel hierarchisch aufgebaut. In der einfachen Variante bedeutet das: Ein Archiv enthält eine große Zahl an Beständen, jeder einzelne Bestand enthält wiederum eine noch größere Anzahl an einzelnen Archivalien.

Dabei lassen sich für alle diese Ebenen Erschließungsinformationen festhalten, z.B. Name und Adresse des Archivs; Titel, Laufzeit und Einleitung der einzelnen Bestände; Titel und Laufzeit der einzelnen Archivalien.

Um diese in einander verschachtelten Informationen strukturiert wiederzugeben eignen sich XML-Formate (Extensible Markup Language). Mittels XML-Dokumenten lassen sich beliebige Hierarchien darstellen und ineinander verschachteln.

Wie funktioniert eine XML-Datei?

- XML Dateien Bestehen aus Elementen: `<element>`
- Ein Element muss immer geschlossen werden: `</element>`

- Zwischen das öffnende und das schließende Element kann Text eingetragen werden:

```
<element>Elementinhalt als Text zwischen öffnendem und schließendem Element</element>
```

- Elemente können ineinander beliebig tief verschachtelt werden:

```
<element>
  <unterelement>Elementinhalt als Text zwischen öffnendem und schließendem Element</unterelement>
</element>
```

- Alle Elemente und Unterelemente müssen in der *korrekten* Reihenfolge wieder geschlossen werden (vergleichbar mit Klammern in der Mathematik). Folgende Reihenfolge würde *nicht* funktionieren:

```
<element>
  <unterelement>Elementinhalt als Text zwischen öffnendem und schließendem Element</element>
</unterelement>
```

- Elemente, die nur als Klammer für ein oder mehrere Unterelemente (oder auch mehrere ineinander verschachtelte Unterelemente) dienen und keinen eigentlichen Elementinhalt haben, werden auch als "Hüllenelemente" bezeichnet. So auch das `<element>` im obigen Beispiel, hier zur Veranschaulichung mit anderem Namen:

```
<huellenelement>
  <unterelement>Elementinhalt als Text zwischen öffnendem und schließendem Element</unterelement>
</huellenelement>
```

- Elementen können durch Attribute zusätzliche Informationen hinzugefügt werden (Beispiele aus EAD(DDB)):

```
<unitdate normal=2019-12-31>Silvester 2019</unitdate>  Angabe der Laufzeit als Freitext im Elementinhalt sowie als maschinenlesbares Datum im Attribut normal (normalisierte Datumsangabe)
```

```
<origination label="pre">Beispielbehörde Musterhausen</origination>  Angabe einer Provenienz im Elementinhalt, Kennzeichnung, dass es sich lediglich um die Vorprovenienz handelt durch die Angabe "pre" im Attribut label.
```

Was ist das Besondere an EAD(DDB)?

Die in EAD(DDB) definierten Elemente spiegeln die in Archiven üblichen Erschließungsangaben wieder. Im Gegensatz zu vielen anderen Metadatenformaten ist EAD(DDB) dabei recht gut menschenlesbar und verständlich. Eine allgemeine Einführung zum Verständnis des Formats mit einer Vorstellung einiger wichtiger Elemente, die sich überwiegend unmittelbar aus der archivarischen Praxis ableiten, findet sich unter [2.1 Abbildung der archivischen Erschließung in Findbuch und Tektonik](#). Eine ausführliche Dokumentation aller Elemente und Attribute in EAD(DDB) findet sich unter [4. EAD \(DDB\) Tag Library](#).