

## Linkcontainer – <ptr>-links – <rs>-links – emptylinks – Legends, Digilin Images

Letztes Update: 10/2011, W. Zieger

### <ptr>-link – „Einsprung“-Verbindung vom Census zum KHI-Text

- Bei Aktivierung des links [Fulltext] in der Census-Datenbank wird man zu unserer Literatur-Datenbank an die „entsprechende Stelle“ geleitet.
- Von der Cenus Datenbank springt man auf das KHI-Transkript durch Angabe der URL, die um die spezifischen Variablen für Seitenzahl und PTR ID erweitert wird
- Der PTR Link wird mithilfe eines Icons oder einer Hintergrundmarkierung der jeweiligen Einsprungzeile verdeutlicht (nur, wenn auf diese Weise vom Census aus hinein gesprungen wurde – ansonsten erfolgt keine Visualisierung!).
- laufende TN001\_...Nummern werden vom Census vergeben und wurden bereits in die CENSUS-Datenbank eingetragen / könnten aber auch automatisiert erzeugt werden (letzteres wurde inzwischen verworfen)
- Einsprunglink Syntax:
  - Einsprunglinks müssen prinzipbedingt immer auf die Transkripte zeigen
  - Geliefert werden müssen zwei Parameter: die OpacID (bzw. AlephSysNummer) und der <ptr> Link, der in der Regel so aussieht „TN001\_00118“. Ich schlage vor, diesen <ptr> Parameter dann auch tatsächlich als „ptr“ zu bezeichnen, so dass ein typischer Einsprunglink wie folgt aussieht:
    - [http://tn.khi.fi.it/index.php?id=20&data=b304254f&type=transcript&ptr=TN001\\_00118](http://tn.khi.fi.it/index.php?id=20&data=b304254f&type=transcript&ptr=TN001_00118)
      - Was dabei auffällt, ist, dass der „pos“ Parameter fehlt. Dieser pos Parameter gibt normaler Weise die physische Seitenzahl (bzw. Scannummer) an. Leider können unsere Partner workflowbedingt diese Seitenzahl nicht ohne größeren Aufwand angeben. Daher müsste praktisch ein Miniindex erstellt werden, aus dem abgelesen werden kann, welche <ptr>-ID hinter welche physischer Seitenzahl fällt (das wäre so etwas wie <br pphys="pphys10">) und somit die „pos“ erst aus dem ptr Parameter ermittelt werden kann. Das wäre möglicherweise auch in Realtime (also ohne vorher erstellten Index) machbar, aber das kann ich nicht einschätzen.
- Der <ptr>-link wird nicht von einem Linkcontainer umschlossen. Er kann aber innerhalb eines <img><legend>... Tags auftauchen.
- Früher wurde <ptr> zusätzlich mit dem Parameter „cRef“ versehen, der auf die „monumentum.xml“ referenzierte. Das ist inzwischen obsolet, da es für das Prinzip des Einsprungsverhaltens nicht relevant ist.

- Beispiel für den <ptr> Tag:
- `<ptr xml:id="TN001_00118"/>`

### ~~<rs>-links – Verbindung vom KHI-Text zum Census~~

→ **Obsolet seit 10/2010! Ersetzt durch <cm> Links**

- ~~Die Links dienen der genaueren Bestimmung von einzelnen Münzen (Monuments). Sie führen aus unserer Datenbank hinaus zu den Monument-Einträgen im Census.~~
- ~~Sie werden in unseren Texten von Linkcontainern umschlossen.~~

### <cm>-Links – Verbindung vom KHI-Text zum Census

- Die Links dienen der genaueren Bestimmung von einzelnen Münzen (Monuments). Sie führen aus unserer Datenbank hinaus zu den Monument-Einträgen im Census.
- Sie werden in unseren Texten von Linkcontainern umschlossen bzw. tauchen sie unterhalb von <img><legend> auf
- Technische Funktionsweise:
  - Es wird anhand der Angabe des Attributs „CID“ ein Eintrag innerhalb der Datei Monumentum.xml anhand des dort vorhandenen Attributs „censusID“ gesucht und dann darauf verlinkt. Bei der Darstellung des Transkripts wird zudem der „Key“ Eintrag der monumentum.xml als „Link-Erläuterungstext“ innerhalb des Linkcontainers eingeblendet. Die hierfür entscheidende Datei Monumentum XML wird zunächst per Hand vom Census gepflegt, soll aber in einer späteren Projektphase durch eine automatisierte Abfrage der Census „easydb“ Datenbank regelmäßig generiert/aktualisiert werden.
- Syntax:
  - `<cm CID="[censusID]" type="monument"/>`
  - → Das „type“-Attribut lautet aktuell immer „monument“ – könnte also für die Website vernachlässigt werden.
- Beispiel
  - `<cm CID="10026009" type="monument"/>`
- Hinweis:
  - „cm“ soll für „Census Monumentum“ stehen

### Emptylinks

- Emptylinks werden vergeben, wenn Münzen bisher nicht mit dem Census oder anderen DB verbunden werden können.
- Es ist möglich, unterschiedliche Arten von emptylinks zu unterscheiden: bspw. sind erwähnte Münzen nicht zu identifizieren, es handelt sich um Fälschungen oder sie stammen aus einer anderen Epoche, die für den Census keine Rolle spielt.
- In einer DTD würde das dann so aussehen:

```
-----
<!ATTLIST emptylink reason (Census_non_definable |
    Census_other_epoch) >
-----
```

- Im tatsächlichen XML-Syntax in den Transkripten würde das dann wie folgt aussehen:

```
-----
<emptylink reason="Census_non_definable"/>
Oder
<emptylink reason="Census_other_epoch"/>
Oder nach wie vor
<emptylink/>
-----
```

– **Achtung! Das Attribut “reason” wird aktuell *NICHT* verwendet!**

- Emptylinks tauchen i.d.R. innerhalb von Linkcontainern oder innerhalb von Bildlegenden auf

### Linkcontainer

- Linkcontainer dienen der Bündelung von Links, die von einer im Text genannten oder visualisierten Münze auf andere Datenbanken verweisen.
- Im Münzcontainer haben deshalb <cm>-links, (MKM-links) und emptylinks Platz, die bei Öffnung des Münzcontainer-Fensters entsprechend visualisiert werden.
- **MKM Links sind aktuell nicht darzustellen.**

### Bildlegenden (<img><legend>[...]</legend></img>)

- Gekennzeichnet durch <legend> Tag
- Tauchen ausschließlich unterhalb von <img> auf
- Ziel ist die Beschreibung des Bildes in Form einer „Untertafel“ → wenn das Bild ein Digilib Image ist, können bei den Links entsprechende Digilibparameter auftauchen

- Verhalten ähnlich wie Linkcontainer (zusätzlich können <ptr> Einsprungpunkte sowie Digilib Parameter bei den <cm> Links verwendet werden)
- Gestattete Untertags:
  - <emptylink>
  - <ptr>
  - <cm>
- Die Tags <cm> und <emptylink> können mit dem Parameter „digilibno“ erweitert sein – dieser sollte mit ausgegeben werden – er ist eine Referenz auf die Digilib Bilder, die eben diese Nummern als „Marker“ beinhalten. Der im Beispiel weiter unten verwendete Parameter „digilibpara“ kommt aktuell nicht zum Einsatz. *Wäre er gesetzt, könnte man den ausgegebenen „digilibno“ Parameter mit einem Link auf das durch Digilib geöffnete <img num> mit genau den URL-Parametern setzen, die unter „digilibpara“ angegeben sind, um eine gezoomte Ansicht des Bildausschnitts zu erhalten.*

## Digilib Bilder

Digilib ist ein Imageviewer, der serverseitig das Zoomen und Drehen von Bildern erlaubt – und entsprechend per Parameter vorgegebene Zoom- und Drehstufen direkt umsetzt. Darüber hinaus kann man Bildbereiche durch kleine nummerierte „Marker“ kennzeichnen. Für Digilib Bilder wird der <img num=###> Parameter erweitert durch den Parameter „digilibpara“ – der genau diese Angaben zu Zoomstufe und etwaige Markerkennzeichnungen an Digilib weitergibt.

Der DigilibViewer MUSS dafür als iframe eingebunden werden, da leider zwar das von ihm erzeugte Bild direkt eingeblendet werden könnte, dann aber ohne die für Translatio Nummorum essentiellen Markierungen von Bildbereichen. Zur genaueren Technik informiert das Beispiel am Ende dieser Dokumentation.

## Beispiel für Digilib Auszeichnung in Zusammenhang mit Legend

Dieses Beispiel bezieht sich auf das Bild unter dieser URL:

[http://biblioweb1.khi.fi.it:8080/greyskin/diginew.jsp?fn=/translatio/books/b258350f/obj\\_i\\_mgg1-1&mk=0.2864/0.2039,0.5898/0.1668,0.9119/0.1878,0.2958/0.377,0.9138/0.3782,0.2977/0.8751,0.6011/0.8751,0.9176/0.8813](http://biblioweb1.khi.fi.it:8080/greyskin/diginew.jsp?fn=/translatio/books/b258350f/obj_i_mgg1-1&mk=0.2864/0.2039,0.5898/0.1668,0.9119/0.1878,0.2958/0.377,0.9138/0.3782,0.2977/0.8751,0.6011/0.8751,0.9176/0.8813)

Er setzt sich also zusammen aus den <img> Parametern „num“ und „digilibpara“ sowie der Base-URL

<http://biblioweb1.khi.fi.it:8080/greyskin/diginew.jsp?fn=/translatio/books/> und der OpacID (hier b258350f), die im XML File ganz oben steht als <opacid> und bekanntlich den Namen des Digital-Objects-Verzeichnisses bildet.

Also [BaseURL] + <opacid> + “/“ + <img num> + “&“ + <img digilibpara> .

Und richtig, eine FileEndung des Images muss nicht angegeben werden, Digilib sucht die sich selbst.

```
<img num="imgg1-1"
digilibpara="mk=0.2864/0.2039,0.5898/0.1668,0.9119/0.1878,0.2958/0.377,0.9138/0.3782,0.2977/0.8751,0.6011/0.8751,0.9176/0.8813&ws=1">
```

```
<!-- 1. Teilbild -->
<ptr xml:id="TN001_00118"/>
<cm
  digilibno="1"
  CID="10018806"
  type="monument"
  digilibpara="obj_imgg1-1&wx=0.1594&wy=0.1199&ww=0.142&wh=0.0921&mk=0.2864/
0.2039,0.5898/0.1668,0.9119/0.1878,0.2958/0.377,0.9138/0.3782,0.2977/0.8751,0.6011/0.8751,0.9176/0.8813&ws=1"
/>
```

```
<!-- 2. Teilbild -->
<ptr xml:id="TN001_00119"/>
<cm
  digilibno="2"
  CID="10018806"
  type="monument"
  digilibpara="obj_imgg1-1&wx=0.1594&wy=0.1199&ww=0.142&wh=0.0921&mk=0.2864/
0.2039,0.5898/0.1668,0.9119/0.1878,0.2958/0.377,0.9138/0.3782,0.2977/0.8751,0.6011/0.8751,0.9176/0.8813&ws=1"
/>
```

```
<!-- 3. Teilbild -->
<ptr xml:id="TN001_00120"/>
<cm
  digilibno="3"
  CID="10018806"
  type="monument"
  digilibpara="obj_imgg1-1&wx=0.1594&wy=0.1199&ww=0.142&wh=0.0921&mk=0.2864/
0.2039,0.5898/0.1668,0.9119/0.1878,0.2958/0.377,0.9138/0.3782,0.2977/0.8751,0.6011/0.8751,0.9176/0.8813&ws=1"
/>
```

```
<!-- 4. Teilbild mit unbekanntem Link aber bereits definiertem Ausschnitt-->
<emptylink
  digilibno="4"
  digilibpara="obj_imgg1-1&wx=0.1594&wy=0.1199&ww=0.142&wh=0.0921&mk=0.2864/
0.2039,0.5898/0.1668,0.9119/0.1878,0.2958/0.377,0.9138/0.3782,0.2977/0.8751,0.6011/0.8751,0.9176/0.8813&ws=1"
/>
```

```
<!-- 5. Teilbild mit unbekanntem Link ohne festgelegten Ausschnitt -->  
<emptylink digilibno="5" />
```

```
</legend>
```

Dialoghi DIDONANTONIOAGOSTINI... [Weiterer Bildtext]

```
</img>
```

**Bitte beachten Sie, dass der Parameter „digilibpara“, der als Erweiterung für <emptylink> und <cm> in diesem Beispiel auftaucht, aktuell NICHT eingesetzt wird.**