

XML-Praxis

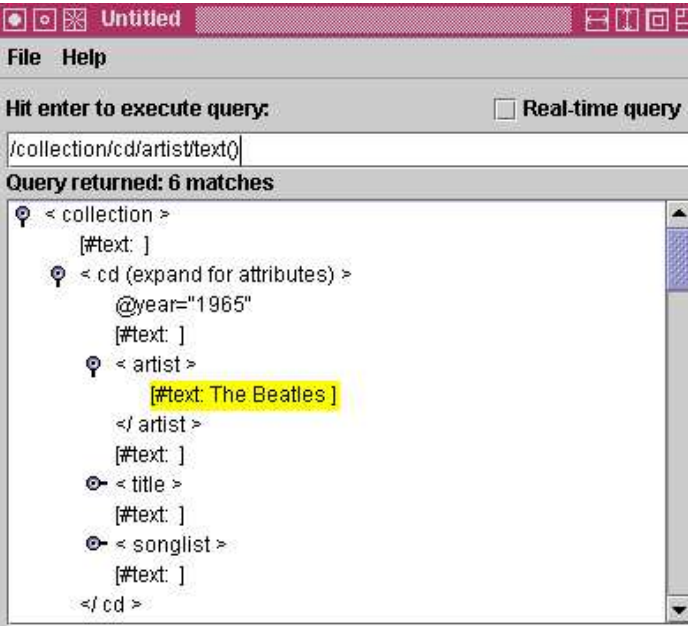
XSLT, Teil 1 – Übungen

Jörn Clausen

joern@TechFak.Uni-Bielefeld.DE

unsere CD-Sammlung ...

- Ausgangspunkte: [cd-collection.xml](#), [cd-collection.dtd](#)
- `xpathtester` in Verzeichnis mit obigen Dateien starten
- `cd-collection.xml` laden



The screenshot shows the XPath Tester application window. The title bar reads "Untitled". The menu bar contains "File" and "Help". Below the menu bar, there is a text input field with the query `/collection/cd/artist/text()` and a checkbox labeled "Real-time query" which is currently unchecked. Below the input field, it says "Query returned: 6 matches". The main area displays the XML tree structure with the following content:

```
< collection >
  [#text: ]
  < cd (expand for attributes) >
    @year="1965"
    [#text: ]
    < artist >
      [#text: The Beatles ]
    </ artist >
    [#text: ]
    < title >
      [#text: ]
    < songlist >
      [#text: ]
    </ cd >
```

XPath-Ausdrücke ausprobieren

- erst überlegen, dann ausprobieren, dann erklären

```
/collection
```

```
/collection/cd
```

```
/collection/cd/artist/text()
```

```
/collection/cd/@year
```

```
/collection/cd[@year="1973"]
```

```
/collection/cd[@year="1973"]/artist/text()
```

```
//song[position()=3]
```

```
//song[3]
```

```
//song[position()=last()]
```

```
count(/collection/cd)
```

```
/collection/cd[contains(artist,"Beatles")]/title
```

die CD-Sammlung in HTML

- rudimentäres Stylesheet in `cd2html1.xsl`

```
<xsl:template match="/collection">
  <html>
    <head>
      <title>CD-Collection</title>
    </head>
    <body>
      <xsl:apply-templates/>
    </body>
  </html>
</xsl:template>
```

- XSLT-Prozessor aufrufen und Ausgabe in Datei umlenken

```
$ xsltproc cd2html1.xsl cd-collection.xml > cd-collection.html
```

Song-Liste

- erzeuge *unordered list* () aus Song-Liste
- templates für songlist und song

```
<xsl:template match="songlist">  
  <ul>  
    <xsl:apply-templates/>  
  </ul>  
</xsl:template>
```

```
<xsl:template match="song">  
  <li>  
    <xsl:apply-templates/>  
  </li>  
</xsl:template>
```

- [cd2html2.xsl](#)

Überschrift

- erzeuge Überschrift mit Künstler und Titel

```
<h1>The Beatles / Help!</h1>
```

- In welchem template?

```
<xsl:template match="cd">  
  <h1>  
    <xsl:value-of select="artist"/>  
    <xsl:text> / </xsl:text>  
    <xsl:value-of select="title"/>  
  </h1>  
  <xsl:apply-templates/>  
</xsl:template>
```

- [cd2html3.xsl](#)

Überschrift, cont.

- Problem?
- `<artist>` und `<title>` werden doppelt verarbeitet
- Lösungsmöglichkeiten?
- „leere“ templates

```
<xsl:template match="artist|title"/>
```
- [cd2html4.xsl](#)
- andere Möglichkeit: template-Aufrufe steuern