

**B-KH**



# Generierung von PDF-Dokumenten mittels XSL-FO

**Bernd Hort**

[bernd.hort@hort-net.de](mailto:bernd.hort@hort-net.de)



# Agenda

- Vorstellung
- Motivation
- XSL-FO
- Zusammenspiel mit Notes / Domino
- Zusammenfassung & Erfahrungen
- Web-Ressourcen und Literatur-Empfehlungen



# Vorstellung

- Bernd Hort
- Diplom-Informatiker
- Lotus Notes Anwendungsentwicklung seit 1995
- Dual Principal CLP R5
- CLI Certified Lotus Instructor SA & AD
- IBM Certified Advanced Application Developer - Lotus Notes and Domino 6/6.5

Certified for

**IBM** **Lotus**

e-business software



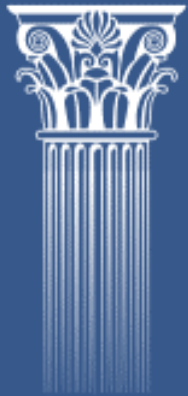
# Ausgangssituation

- Das leidige Thema: „Notes druckt nicht“
- Mehrere Lösungsansätze
  - MS Office Integration
  - Proprietäre Zusatzprodukte wie z.B. Notes2Paper
  - Generierung von PDF-Dokumenten mittels XSL-FO



# Motivation für XSL-FO

- Offener Standard spezifiziert durch das World Wide Web Consortium (W3C)
- Verwendung von XML
- Mit Apache FOP existiert eine Open Source Implementierung
- Bei einem Wechsel vom Notes Client zum Browser kann die Programmierung (zum Teil) übernommen werden



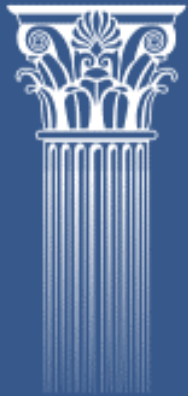
# XSL-FO

- XSL-FO ist Bestandteil der Spezifikation der **eXtensible Stylesheet Language - XSL** des World Wide Web Consortium (W3C)
- Die XSL-Spezifikation beschäftigt sich mit der Verarbeitung und Nutzung von XML-Daten bzw. -Dokumenten
- XSL-Formatting Objects (XSL-FO) beschreibt die Ausgabe in festen Seiten-Dimensionen (Layouter)
- => PDF-Dokumenten



# XSL Transformation - XSLT

- Weiterer Bestandteil der XSL-Specification
- Transformation von einem XML-Dokument in z.B.
  - Ein XML-Dokument mit einem anderem Aufbau
  - Ein (X)HTML-Dokument
  - Ein Text-Dokument
- Funktionale Sprache
  - Ähnliche wie @Functions



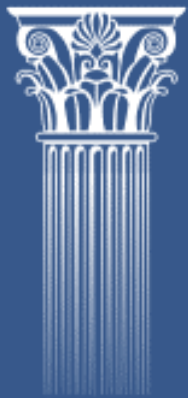
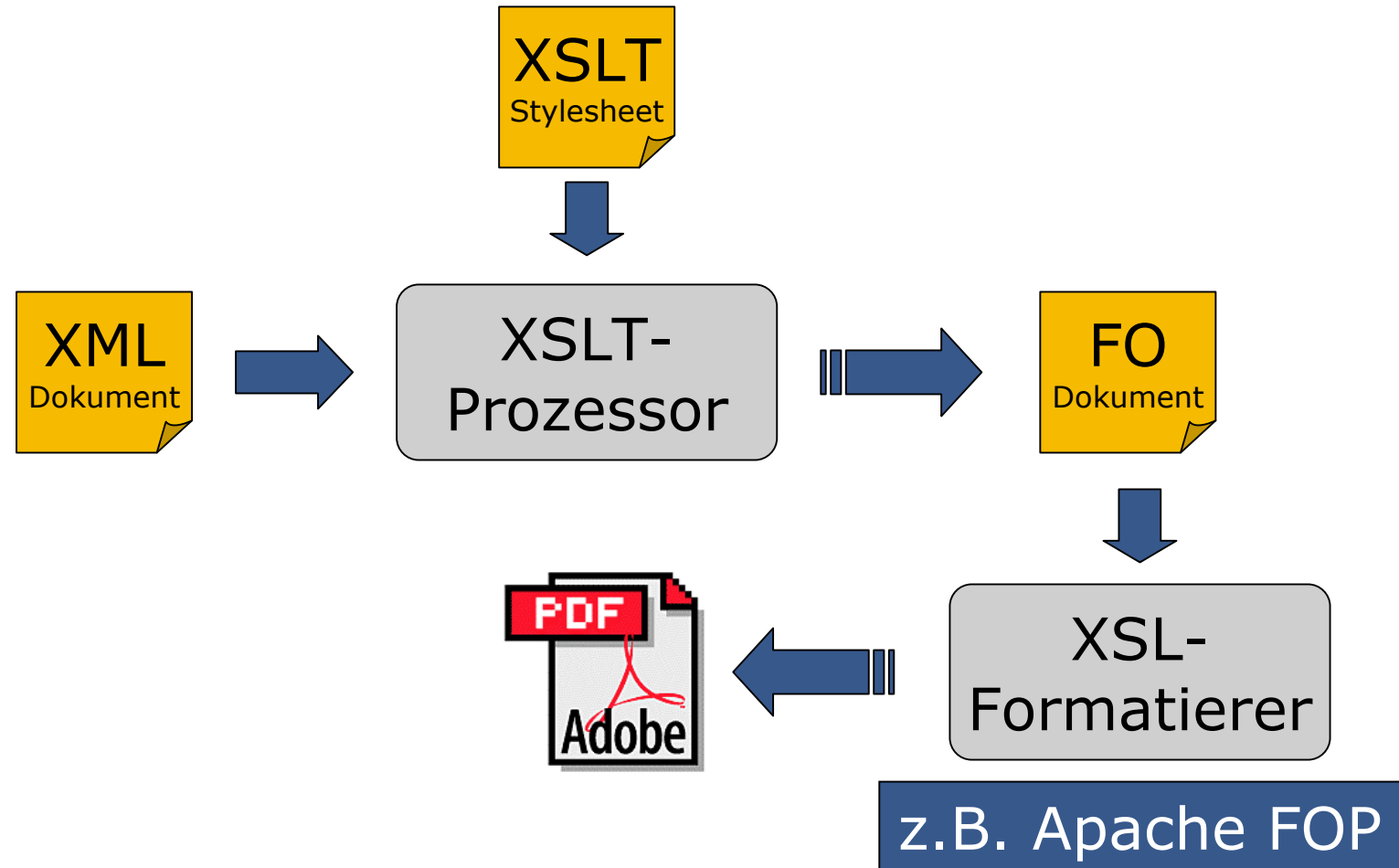
# XML Path Language - XPath

- Navigieren innerhalb eines XML-Dokumentes
- Zugriff auf bestimmte Elemente  
z.B. des Attributes der Personen-ID der siebten Person einer Teilnehmerliste innerhalb der dritten Veranstaltung
- Vorgefertigte Funktionen, z.B.
  - Count() – Anzahl der Knoten
  - Position() – Aktuelle Position





# Der XSL-FO Verarbeitungsprozess



# FO-Dokumente

- In XML geschrieben
- Layout als eine Reihe von geschachtelten Kästen
- XSL-Formatierer entscheidet, wo die Kästen erscheinen
  - Übersteuerbar
- Formatangaben gelten für
  - Block-Bereiche
  - Inline-Bereiche



# „Hello-World“-Beispiel

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

XML-Definition

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
```

XML-Root

```
<fo:layout-master-set>
```

Vorlage

```
<fo:simple-page-master master-name="A4Page"  
  page-height="297mm" page-width="210mm">
```

```
<fo:region-body/>
```

```
</fo:simple-page-master>
```

```
</fo:layout-master-set>
```

```
<fo:page-sequence master-reference="A4Page">
```

Inhalt

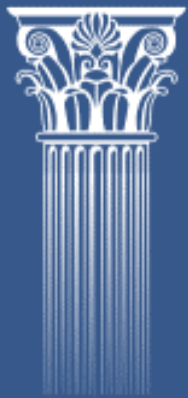
```
<fo:flow flow-name="xsl-region-body">
```

```
<fo:block>Hello World</fo:block>
```

```
</fo:flow>
```

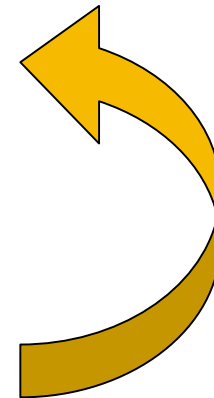
```
</fo:page-sequence>
```

```
</fo:root>
```

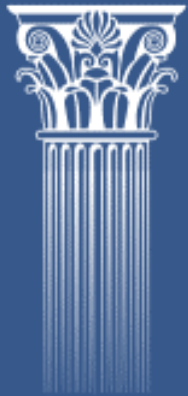
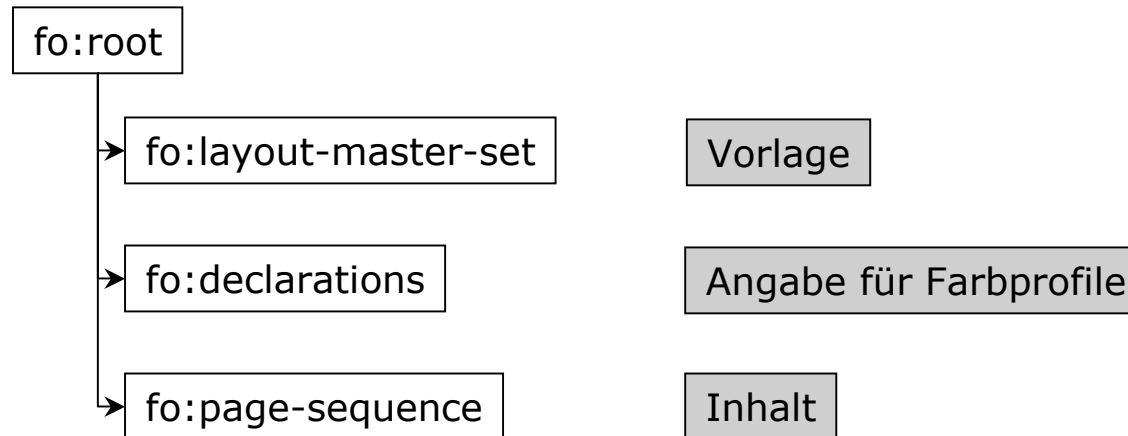


# Referenzierung zwischen Vorlage und Inhalt

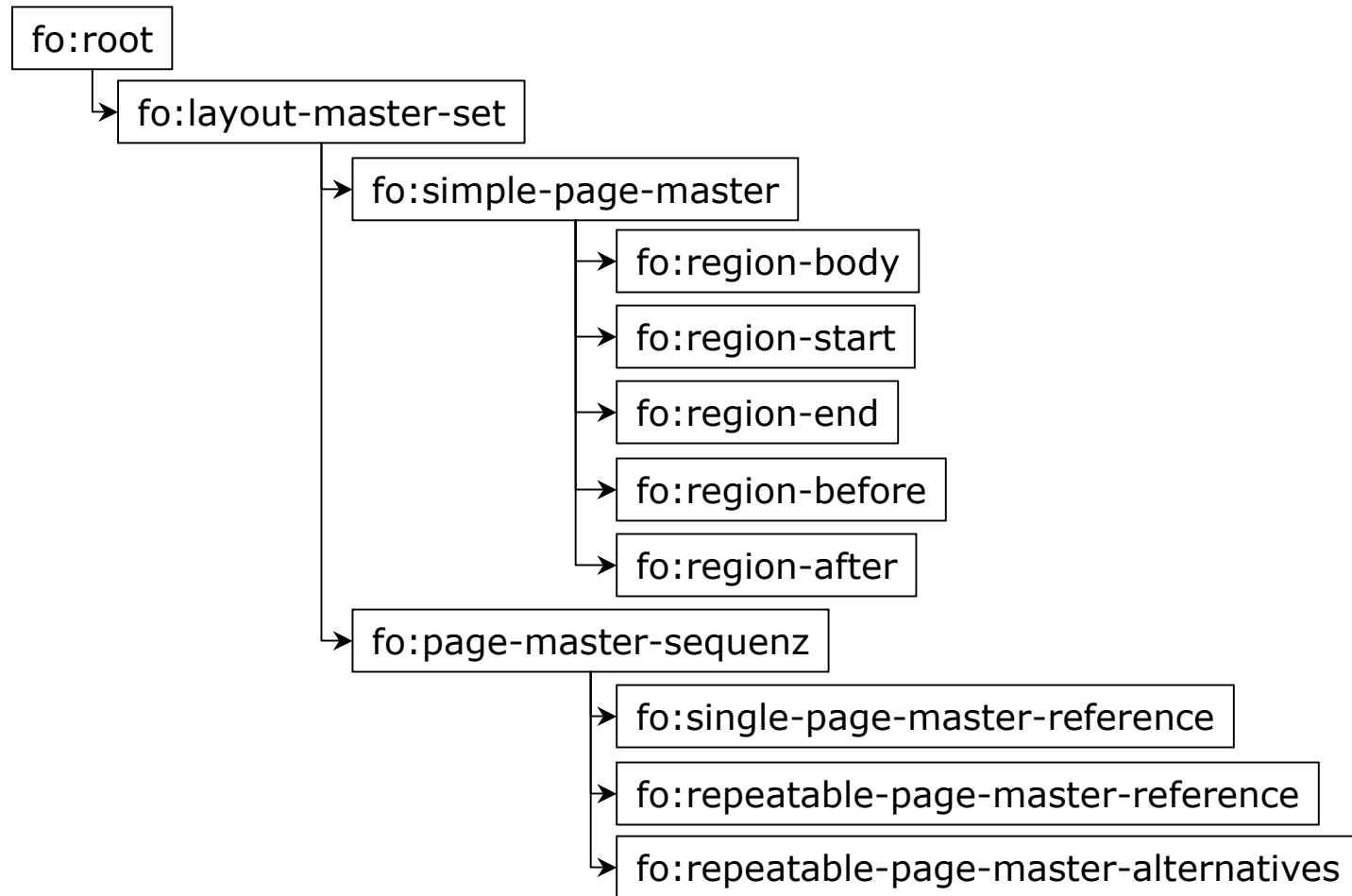
```
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="A4Page"
      page-height="297mm" page-width="210mm">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="A4Page">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>Hello World</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```



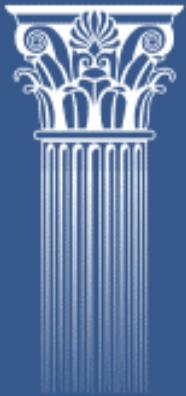
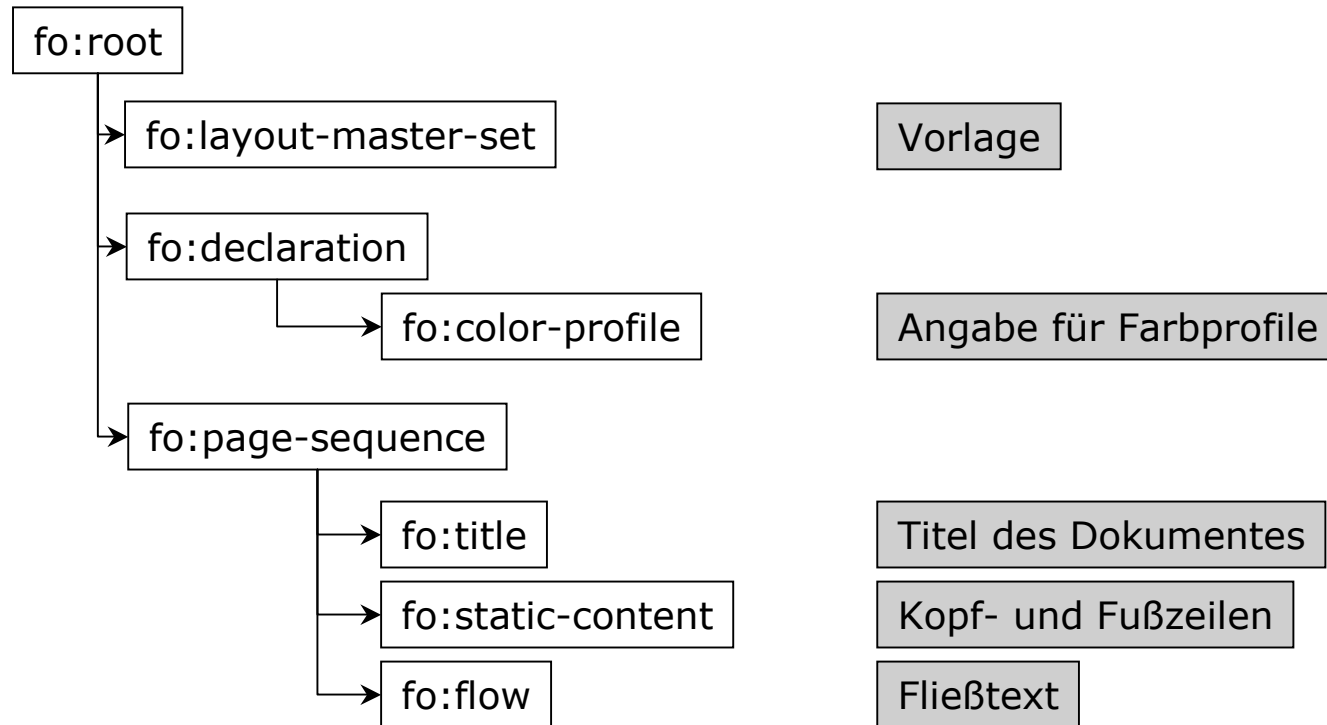
# Aufbau eines FO-Dokumentes



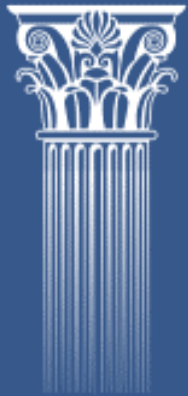
# Aufbau eines FO-Dokumentes



# Aufbau eines FO-Dokumentes



# Seitenaufbau

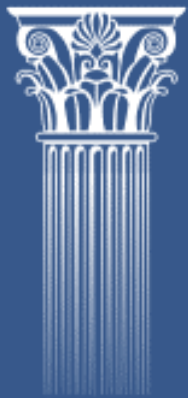


```
<fo:layout-master-set>
  <fo:simple-page-master
    master-name="A4Page"
    page-height="297mm"
    page-width="210mm"
    margin-left="15mm"
    margin-right="15mm"
    margin-top="15mm"
    margin-bottom="15mm">
    <fo:region-before extent="20mm"/>
    <fo:region-after extent="20mm"/>
    <fo:region-start extent="20mm"/>
    <fo:region-end extent="20mm"/>
    <fo:region-body
      margin-top="25mm"
      margin-bottom="25mm"
      margin-left="25mm"
      margin-right="25mm"/>
  </fo:simple-page-master>
</fo:layout-master-set>
```



# Textflüsse und statische Inhalte

- `<fo:flow>` beinhaltet den Fließtext
- Seiten werden gegebenenfalls umgebrochen
- Bestimmt die Anzahl der generierten Seiten
- `<fo:static>` beinhaltet die „statischen“ Textelemente
- Wenn mehr Inhalte als Platz für den statischen Inhalt vorhanden sind, wird der Rest nicht ausgegeben



# Beispiel: Textfluss und statische Inhalte

<pre>&lt;fo:page-sequence master-reference="A4Page"&gt;</pre>	Seitenbereich
<pre>  &lt;fo:static-content flow-name="xsl-region-before"&gt;</pre>	
<pre>    &lt;fo:block&gt;Kopfzeile: EntwicklerCamp '05&lt;/fo:block&gt;</pre>	Kopfzeile
<pre>  &lt;/fo:static-content&gt;</pre>	
<pre>  &lt;fo:static-content flow-name="xsl-region-after"&gt;</pre>	
<pre>    &lt;fo:block&gt;Fußzeile: Seitenaufbau&lt;/fo:block&gt;</pre>	Fußzeile
<pre>  &lt;/fo:static-content&gt;</pre>	
<pre>  &lt;fo:static-content flow-name="xsl-region-start"&gt;</pre>	
<pre>    &lt;fo:block color="#FFFFFF"&gt;Links&lt;/fo:block&gt;</pre>	Linker Bereich
<pre>  &lt;/fo:static-content&gt;</pre>	
<pre>  &lt;fo:static-content flow-name="xsl-region-end"&gt;</pre>	
<pre>    &lt;fo:block color="#FFFFFF"&gt;Rechts&lt;/fo:block&gt;</pre>	Rechter Bereich
<pre>  &lt;/fo:static-content&gt;</pre>	
<pre>  &lt;fo:flow flow-name="xsl-region-body"&gt;</pre>	
<pre>    &lt;fo:block&gt;Lorem ipsum dolor sit amet, consetetur</pre>	
<pre>      sadipscing elitr, sed diam nonumy [...]&lt;/fo:block&gt;</pre>	Fließtext
<pre>  &lt;/fo:flow&gt;</pre>	
<pre>&lt;/fo:page-sequence&gt;</pre>	

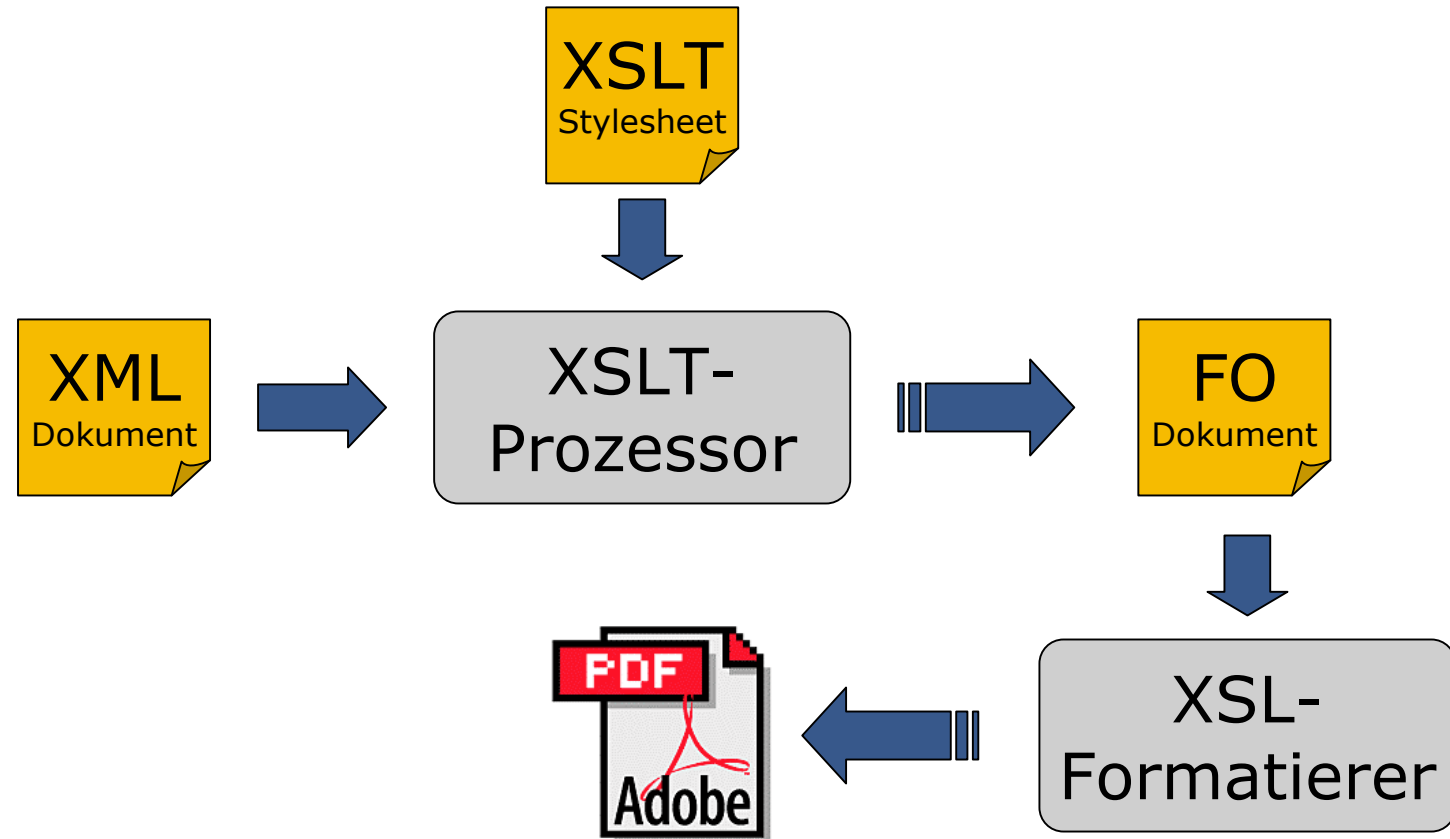


# Textformatierungen

- Angabe als XML-Attribute für
  - Blöcke  
`<fo:block font-family="Helvetica" font-size="14pt" font-weight="bold">Kopfzeile: EntwicklerCamp '05</fo:block>`
  - Inline  
`<fo:inline color="#800000" font-weight="bold">rot und fett</fo:inline>`
- Angaben werden vererbt
- Attribute orientieren sich an Cascading Style Sheets CSS

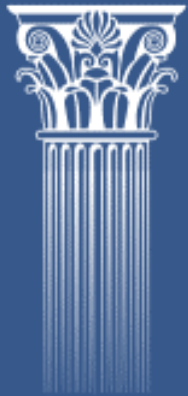


# Der XSL-FO Verarbeitungsprozess



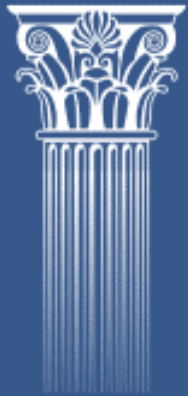
# XML und Domino

- DXL-Exporter
  - XML Toolkit (ab 5.0.3) - rudimentär
  - LotusScript (ab 6.0)
  - Java (ab 6.5)
- Java: Document.generateXML() (seit 5.0.3)
- XML manuell



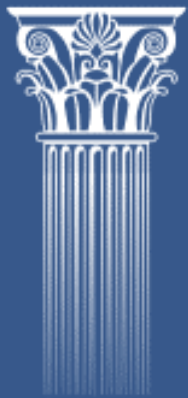
# Domino XML Language - DXL

- DXL ist XML!!!
- Sowohl DXL-Exporter als auch `doc.generateXML()` geben DXL aus
- Sowohl Dokumenten als auch Design Elemente
- Neben dem DXL-Exporter gibt es auch einen DXL-Importer ;-)



# Beispiel DXL-Document

- Generiert aus der Notes-Datenbank PDF.NSF
- Auf der EntwicklerCamp-CD
- Download unter <http://www.hort-net.de/EntwicklerCamp>



# Beispiel: XSL-Transformation DXL in XML

Demo





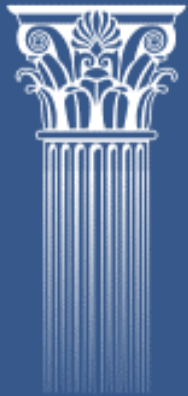
# XSL-Stylesheet um DXL in XML umzuwandeln

## Kopf des XSL-Stylesheets

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  xmlns:dxl="http://www.lotus.com/dxl" exclude-result-prefixes="dxl">  
  <xsl:output method="xml" version="1.0" encoding="UTF-8"  
    indent="yes"/>
```

Namespace xsl => XSL-Transformation

Namespace dxl => Domino XML Language



# XSL-Stylesheet um DXL in XML umzuwandeln

## Verarbeiten des <dxl:database>-Elementes

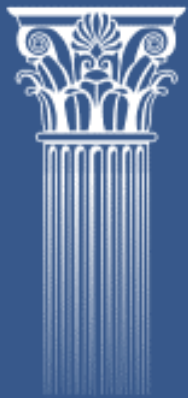
```
<xsl:template match="dxl:database">
  <xsl:element name="database">
    <xsl:apply-templates select="dxl:document"/>
  </xsl:element>
</xsl:template>
```

⇒ Erstellen eines neuen Elementes <database>

## Verarbeiten der <dxl:document>-Elemente

```
<xsl:template match="dxl:document">
  <xsl:element name="{@form}">
    <xsl:for-each select="dxl:item">
      [...]
    </xsl:for-each>
  </xsl:element>
</xsl:template>
```

⇒ Erstellen eines neuen Elementes mit dem Namen der verwendeten Maske

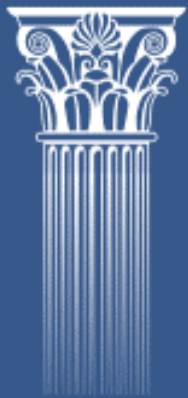


# XSL-Stylesheet um DXL in XML umzuwandeln

## Verarbeiten der <dxl:item>-Elemente

```
<xsl:for-each select="dxl:item">
  <xsl:choose>
    <!-- Items mit Dollarzeichen ignorieren !!! -->
    <!-- Sie verursachen einen Fehler, weil XML-Elemente nicht mit einem
        Dollar anfangen dürfen -->
    <xsl:when test='contains(@name, "$')'/>
    <!-- Items named SaveOptions are skipped -->
    <xsl:when test='@name="SaveOptions"'/>
    <!-- Items starting with 'dsp' are skipped -->
    <xsl:when test='starts-with(@name, "dsp")'/>
    <xsl:otherwise>
      <xsl:element name="{@name}">
        <xsl:value-of select="*" />
      </xsl:element>
    </xsl:otherwise>
  </xsl:choose>
</xsl:for-each>
```

⇒ Erstellen eines neuen Elementes mit dem Namen des Feldes und deren Inhalten



# Beispiel: XSL-Transformation XML in FO

Demo



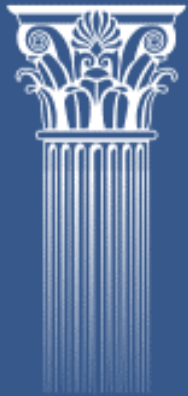
# XSL-Stylesheet um XML in FO umzuwandeln

## Kopf des XSL-Stylesheets

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet version="1.0"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  xmlns:fo="http://www.w3.org/1999/XSL/Format">  
  <xsl:output method="xml" version="1.0" omit-xml-declaration="no"  
    indent="yes"/>
```

Namespace xsl => XSL-Transformation

Namespace fo => XSL-FO



# XSL-Stylesheet XML in FO umzuwandeln

## Verarbeiten des <database>-Elementes

```
<xsl:template match="database">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <fo:layout-master-set>
      <fo:simple-page-master master-name="A4Page"
        page-height="29.7cm" page-width="21cm" margin-top="2cm"
        margin-bottom="2cm" margin-left="2cm" margin-right="2cm">
        <fo:region-body/>
      </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-reference="A4Page">
      <fo:flow flow-name="xsl-region-body">
        <fo:block>
          <xsl:apply-templates select="Sample"/>
        </fo:block>
      </fo:flow>
    </fo:page-sequence>
  </fo:root>
</xsl:template>
```

**XSL:Aufruf für  
<Sample>**

**XSL-FO**



# XSL-Stylesheet XML in FO umzuwandeln

## Verarbeiten der <Sample>-Elemente

```
<xsl:template match="Sample">  
  <fo:block font-size="18pt" font-family="sans-serif" line-height="24pt"  
    space-after.optimum="15pt" background-color="#37588B"  
    color="white" text-align="center" padding-top="3pt">Title:  
    <xsl:value-of select="Subject"/>  
  </fo:block>  
  <fo:block font-size="12pt" font-family="sans-serif" line-height="15pt"  
    space-after.optimum="3pt" text-align="justify">  
    <xsl:value-of select="Richtext"/>  
  </fo:block>  
</xsl:template>
```

**XSL-FO**

**XSL:Aufruf für <Subject>**

**XSL:Aufruf für <Richtext>**



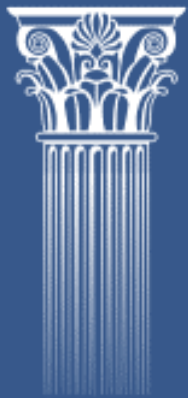
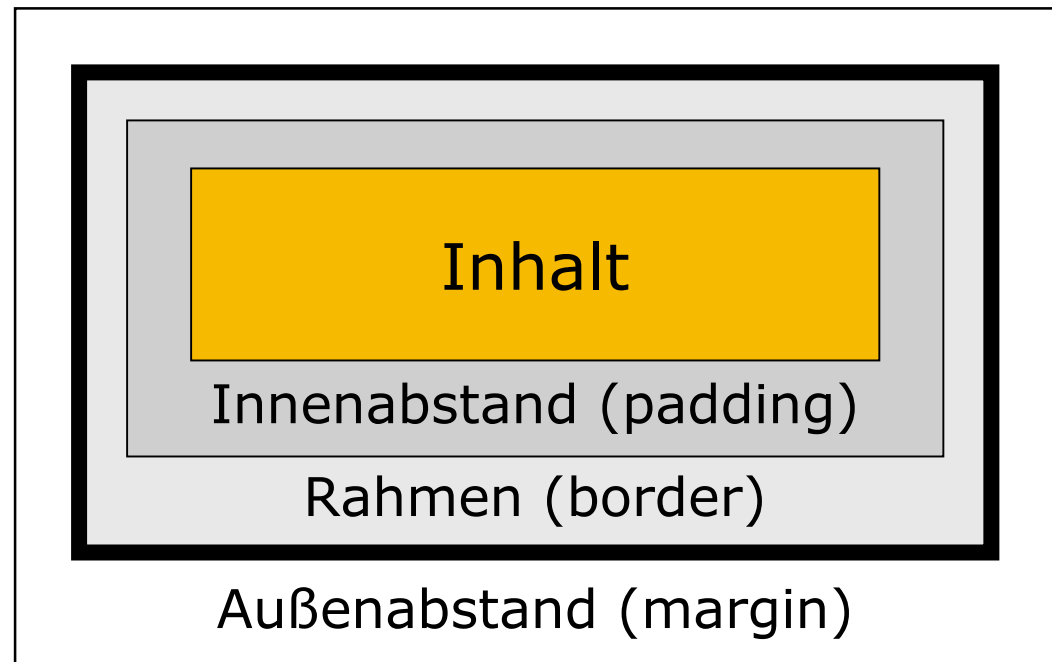
# XSL-FO: Tabellen

```
<fo:table table-layout="fixed",
  inline-progression-dimension="180mm">
<!-- FOP unterstützt nur tab table-layout="fixed" -->
  <!-- Vordefinieren der Spalten -->
  <fo:table-column column-width="proportional-column-width(50)"/>
  <fo:table-column column-width="proportional-column-width(50)"/>
  <!-- Prozentangaben sind in FOP bisher nicht realisiert.
  proportional-column-width(X) funktionieren genau so -->
  <fo:table-body>
    <fo:table-row>
      <fo:table-cell>
        <fo:block>Ein wenig Text</fo:block>
      </fo:table-cell>
      <fo:table-cell>
        <fo:block>Noch mehr Text</fo:block>
      </fo:table-cell>
    </fo:table-row>
  </fo:table-body>
</fo:table>
```





# XSL-FO: Rahmen, Ränder und Abstände



# Zusammenfassung & Erfahrungen

- Durch die Verarbeitungsschritte Domino => XML => XSL-FO => PDF vielseitig einsetzbar
- Einfache Layouts relativ einfach zu realisieren
- Aufwendige Layouts ein wenig zeitaufwändiger
- **XML ist case sensitive !!!**



# Erfahrungen mit Apache FOP

- Hat einige Limitierungen
- Längst nicht alles, was im Standard beschrieben ist, wurde auch umgesetzt
  - Das ist allerdings auch bei einigen kommerziellen Produkten so
- Aufgrund „Open Source“ hohe Verbreitung
  - Einige Treffer bei Google ;-)
- Befindet sich schon recht lange in der Version 0.20.5



# Kommerzielle Alternativen

- Einige Alternativen auf dem Markt
  - XEP
  - Ibex
  - Antenna-House
  - u.a.
- Lösung aus dem Notes Umfeld: Forender
  - Im Beta-Stadium
  - Firma Cappelino
  - Arnd Beißner => EntwicklerCamp 2004
  - <http://www.cappelino.de>



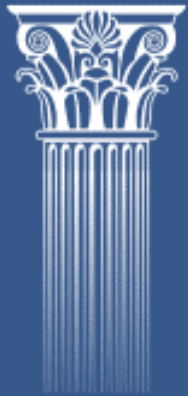
# Erfahrungen mit Notes-Integration

- Starten der JVM verursacht eine Verzögerung beim ersten Start ☹
- FOP-Jar-Files nicht in eine Script Bibliothek integrieren
  - Führt insbesondere über das Netzwerk zu Fehlern, weil die recht großen Jar-Files nicht so schnell zur Verfügung stehen
  - In dem Verzeichnis **Notes\jvm\lib\ext** speichern



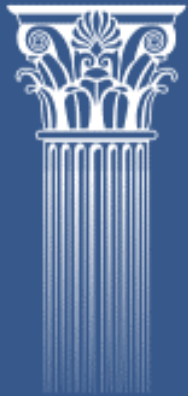
# Erfahrungen mit dem Adobe (Acrobat) Reader

- Bei mehr als 27 geöffneten Dokumenten streikt der Adobe (Acrobat) Reader
- Lösung: Mehrere Seiten in einem Dokument
- Pro `<fo:page-sequence>` Element wird eine Seite erzeugt
- Beim Drucken ist eine größere Anzahl von PDF-Dokumenten die einzeln aufgerufen werden kein Problem



# IBM Ressourcen

- XSL Formatting Objects (XSL-FO) basics  
<http://www-106.ibm.com/developerworks/edu/x-dw-xxslfo-i.html>
- XSL-FO advanced techniques  
<http://www-106.ibm.com/developerworks/edu/x-dw-xxslfo2-i.html>
- IBM Redbook „XML Powered by Domino“  
<http://www.redbooks.ibm.com/abstracts/sg246207.html?Open>
- LDD Artikel „LotusScript: More XML classes Notes/Domino 6“  
<http://www-10.lotus.com/ldd/today.nsf/62f62847467a8f78052568a80055b380/7070013b2de3d53885256cd40060c9da?OpenDocument>




# Web Ressourcen

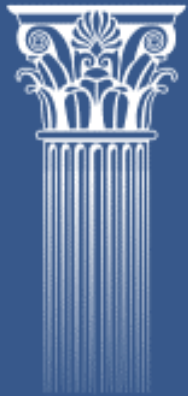
- World Wide Web Consortium (W3C)  
<http://www.w3c.org>
  - Die Standards sind sehr formal gehalten
  - Der Standard zu XSL-FO ist XSL 1.0 ;-)
  - Der Standard zu XSLT ist XSLT 2.0
  - Der Standard zu XPath ist XPath 1.0
- XMLSoftware  
<http://www.xmlsoftware.com>
  - Sehr gute Übersicht über freie und kommerzielle XML-Software
- Apache Software Foundation => FOP  
<http://www.apache.org>





# Literatur

- XSL-FO in der Praxis
  - Manuel Montero Pineda, Manfred Krüger 
  - dpunkt.verlag
  - Sehr gute verständliche Einführung in XSL-FO, XSLT und Xpath
- XSLT Cookbook
  - Sal Mangano
  - O'Reilly & Associates
  - Sehr praxisorientiert
- Die kommerziellen Hersteller von XSL-FO-Tools haben meistens ein Tutorial zum Herunterladen



# Noch Fragen?



# Kontakt / Feedback

Bernd Hort

Dockenhudener Chaussee 21B

D-25469 Halstenbek

Tel. 04101 / 48747

[bernd.hort@hort-net.de](mailto:bernd.hort@hort-net.de)

<http://www.hort-net.de>

Download der Folien und Beispiele

<http://www.hort-net.de/EntwicklerCamp>



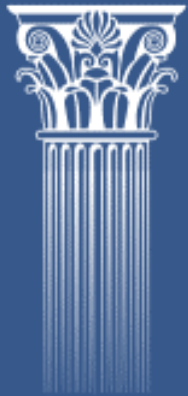
# Anhang: LS2J

- Aufruf von Java-Klassen aus LotusScript heraus
- Ab der Version 6.0
- `Uselsx` `"*javacon"`
- Java-Session initialisieren
- Java-Klasse definieren
- Java-Objekt erzeugen
- Java-Methoden / Java-Eigenschaften verwenden
- Beispiel-DB „LS2J-Examples“ von Julian Robichaux  
<http://www.nsftools.com>



# Beispiel: LS2J

```
UseIsx "*javacon"  
Use "FOPProcess" 'Die Java-Script-Bibliothek  
  
Dim jSession As New JavaSession 'Neue Java-Session  
Dim jclassFOPProcess As JavaClass  
Dim jFOPProcess As JavaObject  
  
'Die Klasse definieren  
Set jclassFOPProcess = jSession.GetClass("FOPProcess")  
'Ein Objekt erzeugen  
Set jFOPProcess = jclassFOPProcess.CreateObject  
  
'Die Methode Process aufrufen  
Call jFOPProcess.process(strFOFilepath, strPDFFilepath)
```



# Anhang: Tipps & Tricks - Layout

- Klareres Layout durch `<xsl:attribute-set>`

```
<xsl:attribute-set name="regular.text">
  <xsl:attribute name="font-family">Helvetica</xsl:attribute>
  <xsl:attribute name="font-size">9pt</xsl:attribute>
</xsl:attribute-set>
<xsl:attribute-set name="header.text" use-attribute-sets="regular.text">
  <xsl:attribute name="font-weight">bold</xsl:attribute>
</xsl:attribute-set>
<xsl:attribute-set name="table.data.head" use-attribute-sets="header.text">
  <xsl:attribute name="padding">2pt</xsl:attribute>
  <xsl:attribute name="border-before-style">solid</xsl:attribute>
  <xsl:attribute name="border-before-width">1pt</xsl:attribute>
  <xsl:attribute name="border-after-style">solid</xsl:attribute>
  <xsl:attribute name="border-after-width">1pt</xsl:attribute>
  <xsl:attribute name="display-align">after</xsl:attribute>
</xsl:attribute-set>
<fo:table-cell xsl:use-attribute-sets="table.data.head" text-align="right">
  <fo:block>Text</fo:block>
</fo:table-cell>
```



# Anhang: Tipps & Tricks - Anzahl Seiten

- Kein vorgefertigtes FO-Element
- Auf der letzten Seite einen leeren Block mit ID-Angabe  
`<fo:block id="last-page"/>`
- Querverweis auf diese ID  
`<fo:page-number-citation ref-id="last-page"/>`
- z.B.  
Seite `<fo:page-number/>` / `<fo:page-number-citation ref-id="last-page"/>`

