

# Hinweise für die Nutzung von XSLTs beim PDF Export der BlueBridge Wiki Extensions

Dieses Dokument beschreibt die Einbindung von XSLTs, um die Format von PDF Exporten zu definieren.

## Wichtiger Hinweis:

Mit dieser Dokumentation beabsichtigen wir, dass Sie einfache Anpassungen für den PDF Export selbst vornehmen können. Es ist allerdings nicht möglich, jede einzelne Funktion zu dokumentieren, und unsere Maintenance & Support Vereinbarung beinhaltet keine Unterstützungsleistungen für das Erstellen und Modifizieren von XSLT-Dateien. Sollten Sie mehr anpassen wollen, als in diesem Dokument beschrieben wird, können Sie uns gerne kontaktieren. Solche Leistungen würden dann im Form eines kleines Projektes getätigt werden.

## Inhaltsverzeichnis

1. Allgemeine Hinweise.....	2
2. Einbinden von XSLTs .....	3
3. Erstellen von XSLTs .....	5
4. Schlussbemerkungen .....	16

## 1. Allgemeine Hinweise

### *Was sind XSLTs?*

XSLT steht für Extensible Stylesheet Language Transformation und definiert also Umwandlungsregeln für XML Dokumenten. Diese Programmiersprache findet häufige Anwendung in Microsoft SharePoint und können benutzt werden, um viele SharePoint Elemente (unter anderem Webparts) zu formatieren.

Auch die Wiki Extensions verwenden diese Technologie und setzen XSLTs ein, um XSL-FO (Formatting Objects) zu erzeugen. Diese legen dann das Format der exportierten PDFs fest. Dies hilft vor Allem, ein CI-konformes Design bei exportierten Dokumenten zu gewährleisten. Im Wesentlich funktioniert die Technologie so, dass der HTML eines Wiki-Artikels eingelesen wird und anhand vordefinierter Regeln umgewandelt wird. Beispielsweise werden alle Texte somit in einer Standard-Schriftart exportiert, und Standard-Elemente wie Seitenränder, Kopf- und Fußzeilen und ein Deckblatt können zentral definiert werden.

In den Wiki Extensions ist bereits eine unsichtbare XSLT eingebaut, die die Formatierung von Exports festlegt. Dies gehört zum Source Code der Wiki Extensions und wird nicht zur Verfügung gestellt. Wenn Sie aber eine eigene Datei erstellen, wird dies zusätzlich zur eingebauten XSLT verwendet – das heißt: Sie müssen nicht Alles definieren, sondern nur die Sachen, die von unserem Standard-XSLT abweichen. Dabei genießt Ihre Datei höhere Priorität, so dass in Konfliktfällen Ihre Eingaben gelten.

Weitere Informationen zu diesen Technologien finden Sie unter diesen hilfreichen Links:

XSLT:

- [http://de.wikipedia.org/wiki/XSL\\_Transformation](http://de.wikipedia.org/wiki/XSL_Transformation)
- <http://de.selfhtml.org/xml/darstellung/index.htm>

XSL FO:

- [http://en.wikipedia.org/wiki/XSL\\_Formatting\\_Objects](http://en.wikipedia.org/wiki/XSL_Formatting_Objects)
- <http://www.xmlpdf.com/builds/ibex.pdf>

*Bitte beachten Sie, dass BlueBridge weder die Richtigkeit, Vollständigkeit noch Aktualität der Inhalte externer Sites gewährleisten kann.*

### **Anpassungen**

XSLTs können teilweise sehr komplex sein und daher sollen Sie diese nur selber erstellen, wenn Sie bereits Erfahrung in der Technologie haben. Da es sich aber um eine Standard-Programmiersprache handelt, bietet BlueBridge keine Unterstützung für das Erstellen von XSLTs im Rahmen der Maintenance & Support-Vereinbarung. Gerne können wir aber ein Angebot unterbreiten, damit wir Sie im Rahmen eines kleinen Projektes unterstützen können.

## 2. Einbinden von XSLTs

### *Eine XSLT Datei einbinden*

Nachdem Sie Ihre XSLT Stylesheet für den Export entwickelt haben, muss dies eingebunden werden. Im einfachsten Fall benennen Sie die Datei einfach in pdf.xslt und kopieren Sie diese in folgendes Verzeichnis:

```
C:\Program Files\Common Files\Microsoft Shared\web server
extensions\12\TEMPLATE\LAYOUTS\BlueBridge.Wiki\export (SharePoint 2007)
```

Oder

```
C:\Program Files\Common Files\Microsoft Shared\web server
extensions\14\TEMPLATE\LAYOUTS\BlueBridge.Wiki\export (SharePoint 2010)
```

### *Mehrere XSLT Dateien für verschiedene Zwecke erlauben*

Um mehrere XSLT-Dateien auf dem Server zuzulassen, muss zunächst eine Änderung in der web.config vorgenommen werden. Zum Beispiel in

```
C:\Inetpub\wwwroot\wss\VirtualDirectories\Default80
```

**Achtung:** Bevor Sie eine Änderung in der web.config vornehmen, sollen Sie in jedem Fall eine Backup-Kopie der web.config machen! Wir empfehlen diesen Schritt ebenfalls nicht, wenn Sie wenig Erfahrung mit web.config haben, da diese Datei sehr wichtig für SharePoint ist.

Folgende Tags sollen im Header definiert werden:

```
<configuration>

  <configSections>

    <sectionGroup name="BlueBridge.Wiki">

      <section name="WikiTemplates"
type="BlueBridge.SharePointExtensions.Galleries.GalleryConfigurationSection,
BlueBridge.Spex.Utils, Version=2.1.0.0, Culture=Neutral,
PublicKeyToken=8ad6826c1afbd920" />

      <section name="Export.PdfStyleSheets"
type="System.Configuration.NameValueSectionHandler, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" />

    </sectionGroup>

  </configSections>
```

Dann muss der entsprechende Eintrag im Body stehen:

```
<BlueBridge.Wiki>
  <Export.PdfStyleSheets>
    <add key=".pdf.xslt" value="./pdf.xslt" />
    <add key=".pdf_test1.xslt" value="./pdf_test1.xslt" />
    <add key=".pdf_category.xslt" value="./pdf_category.xslt" />
  </Export.PdfStyleSheets>
</BlueBridge.Wiki>
```

In diesem Fall stehen drei vorher definierte XSLTs zur Verfügung, die alle im C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\TEMPLATE\LAYOUTS\BlueBridge.Wiki\export abgelegt wurden (pdf, pdf\_test1 und pdf\_category). Unter SharePoint 2010 soll das "12" durch "14" ersetzt werden. Selbstverständlich können Sie die Namen dieser XSLTs ändern.

Vergessen Sie nicht, diesen Abschnitt mit </BlueBridge.Wiki> abzuschließen, falls der Eintrag nicht bereits vorhanden ist.

Danach können Sie diese XSLTs zuweisen, indem Sie die Wiki Konfiguration öffnen und *Konfiguration des PDF-Exports* wählen. Nun stehen die verschiedenen Export-Stile zur Verfügung (bitte beachten Sie, dass Sie ggf. den Application Pool neu starten oder das IIS resetten müssen, bevor diese Einträge erscheinen).

### 3. Erstellen von XSLTs

Unten finden Sie ein paar Beispiele von XSLT-Snippets, die benutzt werden können. Wie oben erwähnt sollen Sie dies nur machen, wenn Sie bereits Erfahrung mit der XSLT-Technologie haben. Denn das Erstellen ist komplex und wir bieten keine Unterstützung für die Anpassung von XSLTs oder Templates (außer dies wird in einem Projekt vereinbart).

Bitte vergessen Sie dabei nicht, die Texte in eine \*.xslt-Datei zu kopieren und im entsprechenden Verzeichnis zu speichern. Dateien sollen in UTF-8 kodiert sein.

#### **Beispiel 1: Table of Contents umbenennen**

In diesem Beispiel wird der Text "TABLE OF CONTENTS" in "**\*\*TABLE OF CONTENTS\*\***" umbenannt. Die entsprechende Stelle wird fett markiert.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata" xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:ibex="http://www.xmlpdf.com/2003/ibex/Format" xmlns:bb-
request="http://schemas.bluebridge.de/xslt/request" exclude-result-
prefixes="xs msdata bb-request">
  <xsl:import href="res://./pdf/wiki2pdf.xslt"/>

  <xsl:template name="layout.toc">
    <!--generate a block of table of contents-->
    <fo:block id="toc" font-family="Arial" font-weight="bold" font-
size="12pt">
      <!--generate the title "Table of Contents"-->
      <fo:block language="DE" keep-with-next.within-page="always" font-
family="Arial" font-weight="bold" font-size="12pt">
        <fo:inline>
          <fo:leader leader-length="0pt"/>
          <xsl:text>**TABLE OF CONTENTS**</xsl:text>
        </fo:inline>
      </fo:block>
      <xsl:apply-templates select="." mode="toc"/>
    </fo:block>
  </xsl:template>

  <xsl:template name="layout.pagefooter.simple">
    <xsl:param name="type"/>
    <xsl:param name="author">
      <xsl:call-template name="webparam">
        <xsl:with-param name="name">spuser_name</xsl:with-param>
      </xsl:call-template>
    </xsl:param>
    <xsl:param name="date">
```

```
<xsl:call-template name="webparam">
  <xsl:with-param name="name">spdate_now</xsl:with-param>
</xsl:call-template>
</xsl:param>
<!-- footer start-->
<fo:static-content flow-name="{ $type }-footer" display-align="after">
  <fo:table width="100%">
    <xsl:attribute name="font-weight">
      <xsl:choose>
        <xsl:when test="$type = 'front'">bold</xsl:when>
        <xsl:otherwise>normal</xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
    <xsl:attribute name="font-size">
      <xsl:choose>
        <xsl:when test="$type = 'front'">11pt</xsl:when>
        <xsl:otherwise>8pt</xsl:otherwise>
      </xsl:choose>
    </xsl:attribute>
    <fo:table-column column-width="80%" />
    <fo:table-column column-width="20%" />
    <fo:table-body>
      <fo:table-row>
        <fo:table-cell>
          <fo:block language="DE" font-family="Arial" line-
height="1.7205">
            <fo:inline>
              <fo:leader leader-length="0pt" /> ** Author: <xsl:value-of
select="$author" />
            </fo:inline>
            <fo:inline>
              <fo:leader leader-length="0pt" />
            </fo:inline>
          </fo:block>
          <fo:block font-family="Arial" language="DE" line-
height="1.7205">
            <fo:inline>
              <fo:leader leader-length="0pt" /> Date: <xsl:value-of
select="$date" />
            </fo:inline>
          </fo:block>
        </fo:table-cell>
        <fo:table-cell>
          <xsl:if test="$type = 'simple'">
            <fo:block language="DE" font-family="Arial" line-
height="1.7205">
              <fo:inline font-size="10pt">
                <fo:leader leader-length="0pt" /> Page <fo:page-number />
              </fo:inline>
            </fo:block>
          </xsl:if>
        </fo:table-cell>
      </fo:table-row>
    </fo:table-body>
  </fo:table>
</fo:static-content>
```

```

        </fo:table-cell>
    </fo:table-row>
</fo:table-body>
</fo:table>
</fo:static-content>
<!-- footer end-->
</xsl:template>

</xsl:stylesheet>

```

## Beispiel 2: Formatierung der Texte

In diesem Beispiel können verschiedene Standard-Schriftarten geändert werden. Somit können Sie bspw. festlegen, wie H1 und H2 Tags sowie der Körpertext gerendert werden sollen.

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata" xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:ibex="http://www.xmlpdf.com/2003/ibex/Format" xmlns:bb-
tidy="http://schemas.bluebridge.de/xslt/htmltidy" xmlns:bb-
request="http://schemas.bluebridge.de/xslt/request" exclude-result-
prefixes="xs msdata bb-request bb-tidy">
  <xsl:import href="res://./pdf/wiki2pdf.xslt"/>

  <!-- styles for the block containing the wiki -->
  <xsl:attribute-set name="wikipage">
    <xsl:attribute name="font-size">10pt</xsl:attribute>
    <xsl:attribute name="font-family">Arial</xsl:attribute>
    <xsl:attribute name="line-height">1.1</xsl:attribute>
    <xsl:attribute name="text-align">justify</xsl:attribute>
    <xsl:attribute name="start-indent">0pt</xsl:attribute>
    <xsl:attribute name="color">green</xsl:attribute>
  </xsl:attribute-set>

  <!-- attributes of H1/H2 -->
  <xsl:attribute-set name="h1">
    <xsl:attribute name="font-size">12pt</xsl:attribute>
    <xsl:attribute name="font-weight">bold</xsl:attribute>
  </xsl:attribute-set>

  <xsl:attribute-set name="h2">
    <xsl:attribute name="font-size">12pt</xsl:attribute>
    <xsl:attribute name="font-weight">bold</xsl:attribute>
  </xsl:attribute-set>

  <!-- override the wiki node generation -->
  <xsl:template name="layout.wikinodes">
    <xsl:param name="nodes"/>
    <fo:block xsl:use-attribute-sets="wikipage">

```

```

        <xsl:apply-templates select="$nodes/node()" />
    </fo:block>
</xsl:template>

<!-- Page layouts -->

<xsl:template name="layout.masterset">
    <fo:layout-master-set >
        <!-- Standard page -->
        <fo:simple-page-master master-name="simple" page-
width="8.268055555555555in" page-height="11.693055555555556in" margin-
top="36pt" margin-bottom="28.35pt" margin-right="70.9pt" margin-
left="70.9pt">
            <fo:region-body region-name="simple-body" margin-top="32.275pt"
margin-bottom="92.55pt"/>
            <fo:region-before region-name="simple-header" extent="11in"/>
            <fo:region-after region-name="simple-footer" extent="11in" display-
align="after"/>
        </fo:simple-page-master>
        <!-- Front Page -->
        <fo:simple-page-master master-name="front" page-
width="8.268055555555555in" page-height="11.693055555555556in" margin-
top="36pt" margin-bottom="28.35pt" margin-right="40.9pt" margin-
left="70.9pt">
            <fo:region-body region-name="front-body" margin-top="77.275pt"
margin-bottom="42.55pt" margin-right="70pt"/>
            <!--background-position-horizontal="right" background-repeat="no-
repeat" background-image="url('../image002.gif') "-->
            <fo:region-before region-name="front-header" extent="11in"/>
            <fo:region-after region-name="front-footer" extent="11in" display-
align="after"/>
        </fo:simple-page-master>
        <!-- Back Page -->
        <fo:simple-page-master master-name="back" page-
width="8.268055555555555in" page-height="11.693055555555556in" margin-
top="36pt" margin-bottom="28.35pt" margin-right="40.9pt" margin-
left="70.9pt">
            <fo:region-body region-name="front-body" margin-top="77.275pt"
margin-bottom="42.55pt" margin-right="70pt"/>
            <!--background-position-horizontal="right" background-repeat="no-
repeat" background-image="url('../image002.gif') "-->
            <fo:region-before region-name="back-header" extent="11in"/>
            <fo:region-after region-name="back-footer" extent="11in" display-
align="after"/>
        </fo:simple-page-master>
    </fo:layout-master-set>
</xsl:template>

</xsl:stylesheet>

```



### Beispiel 3: Kopf- und Fußzeilen

In diesem Beispiel können Kopf- und Fußzeilen definiert werden.

```
<?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"
xmlns:ibex="http://www.xmlpdf.com/2003/ibex/Format"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
xmlns:msprop="urn:schemas-microsoft-com:xml-msprop" xmlns:bb-
tidy="http://schemas.bluebridge.de/xslt/htmltidy" xmlns:bb-
sp="http://schemas.bluebridge.de/xslt/sharepoint" exclude-result-
prefixes="xs msprop msdata bb-tidy bb-sp">
  <xsl:import href="res://./pdf/wiki2pdf.xslt"/>

  <!-- =====
  PAGE SEQUENCES
  ===== -->

  <xsl:template match="WikiExport">
    <xsl:if test="not($isSinglePage) or
not($hideFirstPageInSinglePageView)">
      <!-- frontpage page sequence -->
      <fo:page-sequence master-reference="front" initial-page-number="1"
format="1">
        <xsl:call-template name="layout.pageheader">
          <xsl:with-param name="type" select="'front'"/>
        </xsl:call-template>
        <xsl:call-template name="layout.pagefooter">
          <xsl:with-param name="type" select="'front'"/>
        </xsl:call-template>

        <fo:flow flow-name="front-body">
          <fo:block font="18pt Arial" space-after="0.5cm" space-
before="0.5cm" line-height="15" text-align="center" z-index="2">
            <fo:inline font-weight="bold">
              <xsl:call-template name="document.title">
                </xsl:call-template>
            </fo:inline>
          </fo:block>
          <fo:block font="10pt Arial" space-after="0.5cm" space-before="1cm"
text-align="center" z-index="2">
            <xsl:call-template name="document.description">
              </xsl:call-template>
          </fo:block>
        </fo:flow>

      </fo:page-sequence>
    </xsl:if>
  </xsl:template>

```

```

<!-- wiki pages start -->
<fo:page-sequence master-reference="simple" >
  <xsl:call-template name="layout.pageheader">
    <xsl:with-param name="type" select="'simple'"/>
  </xsl:call-template>
  <xsl:call-template name="layout.pagefooter">
    <xsl:with-param name="type" select="'simple'"/>
  </xsl:call-template>
  <!-- page body start-->
  <fo:flow flow-name="simple-body">
    <xsl:if test="not($isSinglePage)">
      <xsl:call-template name="layout.toc"/>
    </xsl:if>
    <xsl:apply-templates select="." mode="pages"/>
  </fo:flow>
  <!-- page body end-->
</fo:page-sequence>
<!-- wiki pages end -->

</xsl:template>

<!-- =====
PAGE HEADERS
===== -->

<!-- FRONTPAGE -->
<xsl:template name="layout.pageheader.front">
  <fo:block>
    <xsl:call-template name="layout.logo"/>
  </fo:block>
</xsl:template>

<!-- PAGES -->
<xsl:template name="layout.pageheader.simple">
  <xsl:param name="title">
    <xsl:call-template name="document.title">
    </xsl:call-template>
  </xsl:param>

  <fo:block xsl:use-attribute-sets="font.family">
    <fo:inline font-weight="bold">
      <fo:leader leader-length="0pt"/>
      <xsl:value-of select="$title"/>
    </fo:inline>
  </fo:block>

  <fo:block border-before-style="solid" border-before-color="black"
border-before-width="0.5pt" border-before-width.conditionality="retain"
padding-before="1pt" padding-before.conditionality="retain">
    <fo:block xsl:use-attribute-sets="font.family" space-before="0pt"
space-after="0pt" >
      <fo:leader/>

```

```

    </fo:block>
  </fo:block>
</xsl:template>

<!-- =====
PAGE FOOTERS
===== -->

<!-- FRONTPAGE -->
<xsl:template name="layout.pagefooter.front">
</xsl:template>

<!-- PAGES -->
<xsl:template name="layout.pagefooter.simple">
  <xsl:param name="type"/>
  <xsl:param name="author">
    <xsl:call-template name="webparam">
      <xsl:with-param name="name">spuser_name</xsl:with-param>
    </xsl:call-template>
  </xsl:param>
  <xsl:param name="date">
    <xsl:call-template name="webparam">
      <xsl:with-param name="name">spdate_now</xsl:with-param>
    </xsl:call-template>
  </xsl:param>
  <xsl:param name="doctitle">
    <xsl:call-template name="document.title"></xsl:call-template>
  </xsl:param>

  <!-- footer start-->
  <fo:static-content flow-name="{ $type }-footer" display-align="after">
    <fo:table width="100%">
      <xsl:attribute name="font-weight">
        <xsl:choose>
          <xsl:when test="$type = 'front'">bold</xsl:when>
          <xsl:otherwise>normal</xsl:otherwise>
        </xsl:choose>
      </xsl:attribute>
      <xsl:attribute name="font-size">
        <xsl:choose>
          <xsl:when test="$type = 'front'">11pt</xsl:when>
          <xsl:otherwise>8pt</xsl:otherwise>
        </xsl:choose>
      </xsl:attribute>
      <fo:table-column column-width="80%"/>
      <fo:table-column column-width="20%"/>
      <fo:table-body>
        <fo:table-row>
          <fo:table-cell>
            <fo:block >

```

```

        <fo:inline>
            <fo:leader leader-length="0pt"/>Author: <xsl:value-of
select="$author"/>
        </fo:inline>
        <fo:inline>
            <fo:leader leader-length="0pt"/>
        </fo:inline>
    </fo:block>
</fo:block >
    <fo:inline>
        <fo:leader leader-length="0pt"/>Date: <xsl:value-of
select="$date"/>
    </fo:inline>
</fo:block>
</fo:table-cell>
<fo:table-cell>
    <xsl:if test="$type = 'simple'">
        <fo:block >
            <fo:inline >
                <fo:leader leader-length="0pt"/>Page <fo:page-number/>
            </fo:inline>
        </fo:block>
    </xsl:if>
</fo:table-cell>
</fo:table-row>
</fo:table-body>
</fo:table>
</fo:static-content>
<!--footer end-->
</xsl:template>

</xsl:stylesheet>

```

#### **Beispiel 4: Unterschiedliche Kopf- und Fußzeilen**

Wenn Sie übrigens unterschiedliche Einträge für Kopf- und Fußzeilen pro Wiki-Artikel haben möchten, lässt sich über Standard-Funktionen von XSL:FO lösen. Die Funktionen nennen sich **marker** bzw. **retrieve-marker**.

Man muss nur auf jeder Seite einen oder mehrere Marker setzen (z.B. innerhalb der Überschrift); innerhalb der Fußzeile kann man dann darauf referenzieren.

Bitte beachten Sie, dass es sich hier um keine fertigen XSLT-Inhalte handelt, sondern nur ein Snippet von möglichen Inhalten. Wenn Sie diesen Text einfach in eine leere XSLT-Datei kopieren und versuchen, damit zu exportieren, wird es somit fehlschlagen.

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata" xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:ibex="http://www.xmlpdf.com/2003/ibex/Format" xmlns:bb-
request="http://schemas.bluebridge.de/xslt/request" exclude-result-
prefixes="xs msdata bb-request">
  <xsl:import href="res://./pdf/wiki2pdf.xslt"/>

  <xsl:template name="layout.toc">
    <!--generate a block of table of contents-->
    <fo:block id="toc" font-family="Arial" font-weight="bold" font-
size="12pt">
      <!--generate the title "Table of Contents"-->
      <fo:block language="DE" keep-with-next.within-page="always"
font-family="Arial" font-weight="bold" font-size="12pt">
        <fo:inline>
          <fo:leader leader-length="0pt"/>
          <xsl:text>Table of Contents</xsl:text>
        </fo:inline>
      </fo:block>
      <xsl:apply-templates select="." mode="toc"/>
    </fo:block>
  </xsl:template>

<xsl:template name="layout.wikicaption">
  <xsl:param name="caption"/>
  <fo:block xsl:use-attribute-sets="wiki.caption" >
    <xsl:attribute name="id">
      <xsl:value-of select="generate-id(.)"/>
    </xsl:attribute>
    <xsl:value-of select="$caption"/>
  </fo:block>
  <fo:marker marker-class-name="Wiki-Editor"><xsl:value-of
select="Editor"/></fo:marker>
  <fo:marker marker-class-name="Wiki-Chapter"><xsl:value-of
select="WikiChapterTitle"/></fo:marker>

  </xsl:template>

  <xsl:template name="layout.pagefooter.simple">

    <xsl:param name="type"/>
    <xsl:param name="author">
      <xsl:call-template name="webparam">
        <xsl:with-param name="name">spuser_name</xsl:with-
param>
      </xsl:call-template>
    </xsl:param>
    <xsl:param name="date">

```

```

        <xsl:call-template name="webparam">
            <xsl:with-param name="name">spdate_now</xsl:with-
param>
        </xsl:call-template>
    </xsl:param>

    <!-- footer start-->
    <fo:static-content flow-name="{ $type }-footer" display-
align="after">
        <fo:table width="100%">
            <xsl:attribute name="font-weight">
                <xsl:choose>
                    <xsl:when test="$type =
'front'">bold</xsl:when>
                    <xsl:otherwise>normal</xsl:otherwise>
                </xsl:choose>
            </xsl:attribute>
            <xsl:attribute name="font-size">
                <xsl:choose>
                    <xsl:when test="$type =
'front'">11pt</xsl:when>
                    <xsl:otherwise>8pt</xsl:otherwise>
                </xsl:choose>
            </xsl:attribute>
            <fo:table-column column-width="80%" />
            <fo:table-column column-width="20%" />
            <fo:table-body>
                <fo:table-row border-top-style="solid" border-
top-width="0.5pt">
                    <fo:table-cell>
                        <fo:block language="DE" font-
family="Arial" line-height="1.7205">
                            <fo:inline>
                                <fo:leader leader-
length="0pt"/>Editor:
                                <fo:retrieve-marker
retrieve-class-name="Wiki-Editor" retrieve-position="first-including-
carryover"/>
                            </fo:inline>
                            <fo:inline>
                                <fo:leader leader-
length="0pt"/>
                            </fo:inline>
                        </fo:block>
                    <fo:block font-family="Arial"
language="DE" line-height="1.7205">
                        <fo:inline>
                            <fo:leader leader-
length="0pt"/>Date: <xsl:value-of select="$date"/>
                        </fo:inline>
                    </fo:inline>
                </fo:table-row>
            </fo:table-body>
        </fo:table>
    </fo:static-content>

```

```

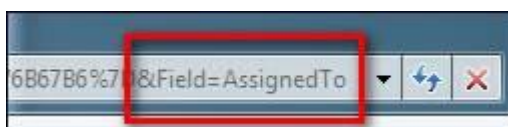
length="0pt"/> Chapter:
retrieve-class-name="Wiki-Chapter" retrieve-position="first-including-
carryover"/>
</fo:inline>
</fo:block>
</fo:table-cell>
<fo:table-cell>
  <xsl:if test="$type = 'simple'">
    <fo:block language="DE" font-
family="Arial" line-height="1.7205">
      <fo:inline font-
size="10pt">
        <fo:leader leader-
length="0pt"/>Page <fo:page-number/>
      </fo:inline>
    </fo:block>
  </xsl:if>
</fo:table-cell>
</fo:table-row>
</fo:table-body>
</fo:table>
</fo:static-content>
<!--footer end-->
</xsl:template>
</xsl:stylesheet>

```

In diesem XSLT sehen Sie zum Beispiel den Code:

```
<xsl:value-of select="Editor"/>
```

Dies greift auf die Spalte mit dem internen Feldnamen "Editor". Das hat an sich nichts mit den Wiki Extensions zu tun und ist Standard-SharePoint: Microsoft benutzt für die internen Feldnamen immer englische Begriffe und wenn man mit Code darauf zugreifen will, muss man den internen Feldnamen kennen. In den meisten Fällen (und auf jeden Fall für die benutzerdefinierten Spalten) kann man diese Namen herausfinden, wenn man in die Listen- bzw. Bibliothekseinstellungen geht. Scrollen Sie dann nach unten und Sie sehen einen Bereich „Spalten“. Klicken Sie auf die gewünschte Spalte und Sie sehen dann rechts in der URL-Leiste den Feldnamen. In diesem Beispiel sieht man den Feldnamen für das Feld „Zugewiesen an“ (AssignedTo):



Am häufigsten werden in den Fußzeilen folgende internen Namen verwendet:

Author (der, der den Artikel erstellt hat)

Created (Datum, an dem der Artikel erstellt wurde)

Editor (der, der den Artikel zuletzt geändert hat)

Modified (Datum, an dem der Artikel zuletzt geändert wurde)

Außerdem können Sie WikiChapterTitle (wie bei den Wiki-Feldern vorgegeben). Bitte beachten Sie, dass die Wiki Felder zuerst aktiviert sein müssen.

#### **4. Schlussbemerkungen**

In diesem Dokument haben wir ein paar Hinweise gegeben, wie Sie einfache XSLT-Dateien erstellen können, wenn Sie bereits Erfahrung mit der XSLT-Technologie haben. Sollten Sie andere Anforderungen oder Fragen haben, können wir gerne ein Angebot für die Anpassung unterbreiten.