BlueBridge Wiki Extensions: Notes for Using XSLTs when exporting to PDF

This document describes how to use XSLTs to define the format of PDF exports.

Important Note
This documentation is designed to help you modify basic elements of the PDF export for yourself. It is not possible to document every single function, and our maintenance and support agreement does not include support for creating and modifying XSLTs. If you would like to modify anything beyond the scope of this document, please contact us to discuss how we can help you in the form of a small project.

Table of Contents

1. General Information

What are XSLTs?

XSLT stands for Extensible Stylesheet Language Transformation, meaning that it defines the transformation rules for XML documents. This programming language can be found often in Microsoft SharePoint and can be used e.g. to format web parts.

The BlueBridge Wiki Extensions use this technology as well, and XSLTs are used to create XSL-FOs (Formatting Objects). These are, in turn, used to define the format of the exported PDF. The main purpose of doing this is to ensure that your exported documents all conform to your Corporate Design guidelines. Essentially, this technology "reads" the HTML code of a wiki page converts it into a PDF according to a number of pre-defined rules. For example, you can set the default font and define things like margins, headers and footers.

The Wiki Extensions already include an invisible XSLT, which defines the default format for exported documents. Please note that since this is part of the Wiki Extensions' source code, it will not be made available to customers or partners. However, it has been built in such a way that you can complement these default settings with your own. That is to say, you do not need to define everything in your XSLT file since your changes are applied *in addition to* the standard settings. In the case of a conflict, your changes have higher priority.

You can find more information about these technologies under the following links:

XSLT:

- http://en.wikipedia.org/wiki/XSLT

XSL FO:

- http://en.wikipedia.org/wiki/XSL_Formatting_Objects
- http://www.xmlpdf.com/builds/ibex.pdf

Please note that BlueBridge cannot be held responsible for the content or accuracy of external sites, nor can we confirm whether the information is up-to-date.

Modifications

Since XSLTs can be very complicated, we recommend that you only make changes to these if you already have experience with the technology. Please note that we do not offer any support for the creation or modification of XSLTs as part of our Maintenance & Support Programme.

We would of course be happy to provide you with a quote if you would like us to help you create an XSLT.

2. Applying XSLTs

Where to save the XSLT file

After you have finished creating the XSLT Stylesheet, you will need to save it so that the Wiki Extensions can access it. The easiest way to do this is to rename your file as pdf.xslt and then copy it into the following directory:

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

Or

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

Using several XSLT files for different scenarios

If you want to use several XSLT files on the same server (e.g. if different wiki libraries require different export formats), then you will need to make a change to the web.config. In most simple configurations, this will be located here:
`C:\Inetpub\wwwroot\wss\VirtualDirectories\Default80`

Important: Before making any changes to the web.config, you should <u>make a backup copy of the web.config file</u>!! Please only perform this following step if you have experience with the web.config, as this file is critical for the running of SharePoint.

You should define the following in the header:

```
<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>
```

You will then have to define the following in the body:

```
<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 this case, we have defined three XSLT files, which should all be saved in `C:\Program Files\Common Files\Microsoft Shared\web server extensions\12\TEMPLATE\LAYOUTS\BlueBridge.Wiki\export` (SharePoint 2007)

Or

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

`These files are called` pdf, pdf_test1 und pdf_category. Of course you can change these names to something that will make more sense in your environment.

Assigning the XSLT file

After doing this, you can assign these XSLT files to the relevant libraries. Open the *Wiki Settings* and then click on PDF Export – *Default Settings*. Use the drop-down box to select the relevant XSLT (note that you may need to reset the application pool or run an IISreset before the entries are displayed).

3. Creating XSLTs

In this section, you will find some examples of XSLT snippets that can be used to make simple modifications to your PDF export formats. As stated above, we recommend that you only do this if you have XSLT experience. We cannot provide support for the creation or modification of XSLTs (unless this is agreed in a separate project) or templates.

Please do not forget to save the snippets as an .xslt file and to copy them into the correct directory. Files should be encoded as UTF-8.

Example 1: Renaming the Table of Contents

In this example, the text "TABLE OF CONTENTS" will be renamed to "**TABLE OF CONTENTS**". The crucial part of the XSLT code is highlighted in bold.

```
<?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>
    <!--footer end-->
  </xsl:template>

</xsl:stylesheet>
```

Example 2: Formatting the text

In this example, the default fonts can be defined. This allows you to set a specific font that should be used for rendering e.g. H1 and H2 tags.

```
<?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>
```

Example 3: Header and footer

In this example, a header and footer have been defined.

```
<?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>
    <!-- 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>
```

Example 4: Varying header and footer texts

If you would like to display different headers and footers for each wiki article, you can do this by using the standard functions in XSL:FO. The relevant functions are called **marker** and **retrieve-marker**.

You will have to set a marker on each page e.g. within the title area; you can then reference these in the footer.

Please note that the following examples do not contain complete XSLT-content but rather snippets of possible content. If you simply copy the text into an empty XSLT file and try to export the file, then the export will fail.

```
<?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>
```

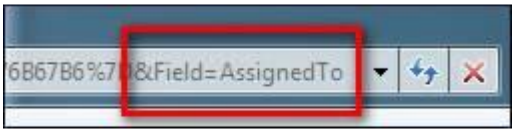
```
                                                      <fo:leader leader-
length="0pt"/> Chapter:
                                                      <fo:retrieve-marker
retrieve-class-name="Wiki-Chapter" retrieve-position="first-including-
carryover"/>
                                                </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 this XSLT snippet you will see the following code:

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

This refers to the column with the internal field name "Editor". This has nothing to do with the Wiki Extensions and is standard SharePoint. Note that Microsoft always using the English terms for this field name and if you want to reference the column by using code, you will need to know the correct field name.

In most cases (and especially for custom columns), you can find out these names by opening the list or library settings. Scroll to the bottom where the columns are listed and click on the relevant column. If you now look in the address bar of your browser, you will see the field name at the end of the URL. In this case, it is AssignedTo:



6B67B6%7D&Field=AssignedTo

The following field names are used most commonly for the footer:

Author (The person who created the article)

Created (Date when the articled was created)

Editor (The person who changed/edited the article most recently)

Modified (Date of most recent change)

You can also use:

WikiChapterTitle (The Chapter Title, as given by the Wiki Fields). Note that these have to be activated for this to work.

4. Closing Remarks

In this document, we have given you a few tips on how to create simple XSLT files if you already have experience with the XSLT technology. If you have any questions or other requirements, then we would be happy to discuss the possibilities and provide you with a quote for further assistance.