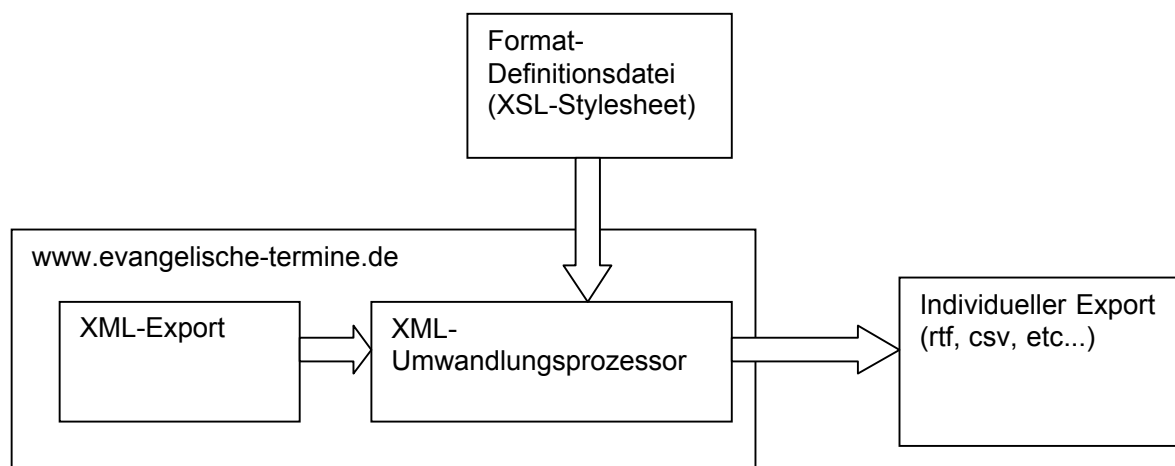


Anleitung zur Erstellung eigener Format-Definitionsdateien

Mit Hilfe sog. Format-Definitionsdateien (XSL-Stylesheets) können aus www.evangelische-termine.de individuelle Exporte realisiert werden. Format-Definitionsdateien, im weiteren 'Stylesheet' genannt, sind reine Textdateien, die sie mit einem Standard-Texteditor (Windows: Editor, Wordpad) bearbeiten können. Bitte verwenden Sie für die Bearbeitung dieser Stylesheets keine Textverarbeitungsprogramme wie Word.

Für die Erstellung eigene Stylsheets ist es hilfreich zu verstehen, wie der Export mit Stylesheets unter www.evangelische-termine.de funktioniert:



Auf www.evangelische-termine.de wird zunächst eine XML-Datei erstellt. Von Ihrem lokalen PC wird eine XSL-Stylesheetdatei auf den Server geladen (Upload über Datei-Eingabefeld). In einem XML-Umwandlungsprozessor wird die XML-Datei gemäß den Vorgaben Ihres Stylsheets zu ihrem individuellen Export umgewandelt. Als Ergebnis erhalten Sie als Download eine rtf, csv, oder txt-Datei.

XML-Export-Dateien

In Ihrem individuellen Export können also alle Angaben enthalten sein, die in der XML-Datei enthalten sind. Es ist hilfreich, sich einen XML-Export (Standard!) einer Veranstaltung zu holen. Dabei ist der Aufbau einer solchen Datei immer gleich. Tipp: Öffnen Sie die XML-Datei mit dem Internet-Explorer.

Bitte lassen Sie sich an dieser Stelle nicht abschrecken! Es sieht zunächst komplizierter aus, als es ist. Schauen Sie sich das Beispiel kurz an und lesen Sie anschließend weiter. Dort folgt eine Erläuterung.

Beispiel XML-Datei einer einzelnen Veranstaltung:

```

<?xml version="1.0"?>
<root>
  <Export>
    <Veranstaltung>
      <_event_ID Label="Veranstaltungs-ID">83620</_event_ID>
      <_event_INPUTMASK_ID>52</_event_INPUTMASK_ID>
      <_event_TITLE Label="Titel">Gottesdienst</_event_TITLE>
      <_event_EVENTTYPE Label="Art der Veranstaltung /
Kategorie">Gottesdienste</_event_EVENTTYPE>
      <_event_PEOPLE Label="Zielgruppe">Alle</_event_PEOPLE>
    </Veranstaltung>
  </Export>
</root>
  
```

```

    <_event_SHORT_DESCRIPTION Label="Kurzbeschreibung /
Untertitel"></_event_SHORT_DESCRIPTION>
    <_event_LONG_DESCRIPTION Label="Ausf&#xFC;hrliche
Beschreibung"></_event_LONG_DESCRIPTION>
    <_event_LINK Label="Internetadresse (eigene Infos im
Internet)"></_event_LINK>
    <_event_KAT Label="Handlungsfeld">1. Spiritualit&#xE4;t, Gottesdienst,
Verk&#xFC;ndigung und Kirchenmusik</_event_KAT>
    <_event_SUBKAT></_event_SUBKAT>
    <_event_SUBSUBKAT></_event_SUBSUBKAT>
    <_event_KAT_2 Label="Handlungsfeld 2 (optional)">1. Spiritualit&#xE4;t,
Gottesdienst, Verk&#xFC;ndigung und Kirchenmusik</_event_KAT_2>
    <_event_EMAIL Label="Email des Pfarramts"></_event_EMAIL>
    <_event_PERSON_ID Label="Predigt">5128</_event_PERSON_ID>
    <_event_PLACE_ID Label="Veranstaltungsort">144</_event_PLACE_ID>
    <_event_REGION_ID Label="Landkreis">N&#xFC;rnb erg (Stadt)</_event_REGION_ID>
    <_event_SUBREGION_ID
Label="Dekanatsbezirk">N&#xFC;rnb erg</_event_SUBREGION_ID>
    <_event_REGION_2_ID Label="Freizeitregion">&gt; bitte w&#xE4;hlen
...</_event_REGION_2_ID>
    <_event_REGION_3_ID
Label="Regierungsbezirk">Unterfranken</_event_REGION_3_ID>
    <_event_PROFESSION_ID Label="Berufsgruppe">Alle MA</_event_PROFESSION_ID>
    <_event_MUSIC_KAT_ID Label="musikalische Kategorie"></_event_MUSIC_KAT_ID>
    <_event_FLAG1 Label="Ausgabe auf www.solideo.de [1|0]">0</_event_FLAG1>
    <_event_TEXTLINE_1 Label="Angabe 1"></_event_TEXTLINE_1>
    <_event_TEXTLINE_2 Label="Angabe 2"></_event_TEXTLINE_2>
    <_event_TEXTLINE_3 Label="Angabe 3"></_event_TEXTLINE_3>
    <_event_TEXTBOX_1 Label="Beschreibung 1"></_event_TEXTBOX_1>
    <_event_TEXTBOX_2 Label="Beschreibung 2"></_event_TEXTBOX_2>
    <_event_TEXTBOX_3 Label="Beschreibung 3"></_event_TEXTBOX_3>
    <_event_MENUE_1 Label="Publikation"></_event_MENUE_1>
    <_event_MENUE_2 Label="Auswahl 2"></_event_MENUE_2>
    <_event_YESNO_1 Label="Familienfreundlich">Ja</_event_YESNO_1>
    <_event_YESNO_2 Label="Bergbahn">Ja</_event_YESNO_2>
    <_event_YESNO_3 Label="Veranstaltung durchgef&#xFC;hrt">keine
Angabe</_event_YESNO_3>
    <_event_DESTINATION Label="&#xD6;ffentlich oder
Intern">&#xF6;ffentlich</_event_DESTINATION>
    <_event_STATUS Label="Freigabe">ok</_event_STATUS>
    <_event_HIGHLIGHT Label="Highlight">rhigh</_event_HIGHLIGHT>
    <_inputmask_NAME Label="Eingabformular-Name">Gottesdienste</_inputmask_NAME>
    <_place_NAME Label="Name des Veranstaltungsortes">Gemeindesaal
Moorenbrunn</_place_NAME>
    <_place_STREET_NR Label="Strasse, Nr.">Gutshofstr. 50</_place_STREET_NR>
    <_place_ZIP Label="Postleitzahl">90475</_place_ZIP>
    <_place_CITY Label="postalischer Ort">N&#xFC;rnb erg</_place_CITY>
    <_place_INFO Label="Weitere Informationen zum Ort"></_place_INFO>
    <_place_HIDDEN Label="Verwaltungsinformationen"></_place_HIDDEN>
    <_user_REALNAME Label="Veranstalter">N&#xFC;rnb erg - Christuskirche
Altenfurt</_user_REALNAME>
    <_user_DESCRIPTION Label="Veranstalter Beschreibung">Evang.-Luth.
Kirchengemeinde</_user_DESCRIPTION>
    <_user_STREET_NR Label="Veranstalter Strasse, Nr">Schornbaumstr.
12</_user_STREET_NR>
    <_user_ZIP Label="Veranstalter Postleitzahl">90475</_user_ZIP>
    <_user_CITY Label="Veranstalter postalischer Ort">N&#xFC;rnb erg</_user_CITY>
    <_user_EMAIL Label="Veranstalter
Email">pfarramt@woauchimmer.de</_user_EMAIL>
    <_user_CONTACT Label="Veranstalter Kontaktdaten">Pfarramt&#xD;
Mo-Fr 9:00-11:30 Uhr&#xD;
Tel.: 0911 834214&#xD;
Fax: 0911 8370532&#xD;
Buslinien 56,57 u. 59 Haltestelle L&#xF6;wenberger Str.</_user_CONTACT>
    <_person_NAME Label="Name des Ansprechpartners">Pfr. Gerhard
Werner</_person_NAME>
    <_person_EMAIL Label="Email des Ansprechpartners"></_person_EMAIL>

```

```

    <_person_CONTACT Label="Kontaktinformationen des
Ansprechpartners"></_person_CONTACT>
    <ID Label="Datum-ID">273140</ID>
    <START Label="am/von">2005-12-31 16:00:00</START>
    <END Label="bis">0000-00-00 00:00:00</END>
    <MODE Label="Start-End-Modus">vonbis</MODE>
    <DATUM>Samstag, 31.12. 16.00 Uhr</DATUM>
    <WOCHENTAG_START_LANG>Samstag</WOCHENTAG_START_LANG>
    <WOCHENTAG_START_KURZ>Sa</WOCHENTAG_START_KURZ>
    <START_DATUM>31.12.</START_DATUM>
    <START_UHRZEIT>16.00</START_UHRZEIT>
    <START_MONAT>Dezember</START_MONAT>
    <START_JAHR>2005</START_JAHR>
    <_person_SURNAME>Werner</_person_SURNAME>
  </Veranstaltung>
</Export>
</root>

```

Der Rahmen der XML-Datei ist immer gleich:

```

<?xml version="1.0"?>
<root>
  <Export>
    <Veranstaltung>
      ...
    </Veranstaltung>
  </Export>
</root>

```

XML-Dateien bestehen aus sog. „Tags“: ein Tag besteht meistens aus einem Anfangs- und einem Schlusstag: <Veranstaltung> ... </Veranstaltung>. Schlusstags sind an dem Schrägstrich vor dem Tagnamen erkennbar. Innerhalb des Tags steht der Inhalt. In diesem Falle die Daten der Veranstaltung, die wiederum von Tags umschlossen sind. Durch die Tags ist erkennbar, um welche Angabe es sich handelt.

Jede einzelne von Ihnen eingegebene Veranstaltung steht innerhalb eines <Veranstaltung>-Tags. Der Titel einer Veranstaltung ist folgendermaßen angegeben:

```

<_event_TITLE Label="Titel">Gottesdienst</_event_TITLE>

```

Der Tagname (blau) lautet in diesem Falle „_event_TITLE“, der Inhalt (rot) ist „Gottesdienst“. Tags können Attribute (grün) enthalten: hier das Attribut „Label“ mit dem Wert „Titel“. Wie Sie unschwer erraten werden, ist dies die Feldbezeichnung in dem Eingabeformular.

Zur Erläuterung: Die Datenbank hat feste Datenbankfelder, die den Tagnamen entsprechen. In diesen Datenbankfeldern können verschiedene Inhalte abgespeichert werden. Je nach Eingabeformular können die Labels unterschiedlich sein. In dem Eingabeformular „Solideo“ ist z.B. das Datenbankfeld _event_TEXTBOX_1 mit der Angabe für die „aufgeführten Werke“ belegt:

```

<_event_TEXTBOX_1 Label="aufgef&#xFC;hrte Werke">Virtuose Orgelmusik von Johann
Sebastian Bach, Naji Hakim u. a. </_event_TEXTBOX_1>

```

Im Eingabeformular „Kirche im Grünen/Berggottesdienst“ nimmt das gleiche Datenbankfeld die Werte für den Ausweichtermin auf:

```

<_event_TEXTBOX_1 Label="Ausweichtermin und/oder Ort">Kapelle am
Kochelsee</_event_TEXTBOX_1>

```

Für die Erstellung von Stylesheets sind die Tagnamen von zentraler Bedeutung. Die Inhalte der Tags können durch die Stylesheets individuell angeordnet werden.

Grundaufbau eines XSL-Stylesheets

In der Zip-Datei „XSL_Stylesheets.zip“ sind einige Beispiele enthalten, die Sie erweitern und verändern können. Öffnen Sie bitte die Textdatei list1.txt.xsl im Ordner „Anleitung“ mit einem Texteditor (nicht Word!):

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output
  method="text"
  indent="no"
  encoding="ISO-8859-1"
  media-type="text/plain"
/>
<xsl:strip-space elements="*" />

<xsl:template match="/root/Export">
  <xsl:call-template name="veranstaltungen"/>
</xsl:template>

<xsl:template name="veranstaltungen">
  <xsl:for-each select="Veranstaltung">
    <xsl:value-of select="WOCHENTAG_START_LANG"/>, <xsl:value-of
select="START_DATUM"/> <xsl:value-of select="START_UHRZEIT"/> Uhr: <xsl:value-of
select="_event_TITLE"/>
  </xsl:for-each>
</xsl:template>

</xsl:stylesheet>
```

Wie Sie sehen, handelt es sich hierbei wiederum um eine XML-Datei. XSL ist eine besonderer XML-„Dialekt“, der Handlungsanweisungen für den XML-Umwandlungsprozessor enthält.

Die verschiedenen Teile sind hier farblich markiert. Der Teil, den Sie verändern sollen, ist blau eingefärbt. Lassen Sie die anderen Teile zunächst unbeachtet.

Der blaue Teil ist ein „Template“ mit dem Namen „veranstaltung“ (er könnte jeden anderen beliebigen Namen haben) und ist mit dem xsl:template-Tag umschlossen:

```
<xsl:template name="veranstaltungen">
  ...
</xsl:template>
```

Ein Template ist ein Handlungsanweisungsabschnitt. In einem Stylesheet können mehrere Templates enthalten sein. Mehr brauchen Sie zunächst nicht zu wissen.

Innerhalb des Templates gibt es eine Wiederholungsfunktion:

```
<xsl:for-each select="Veranstaltung">
  ...
</xsl:for-each>
```

Beachten Sie bitte wieder den Anfangs- und Endtag! Denken Sie an den Aufbau der XML-Exportdatei: in ihr gibt es für jede Veranstaltung ein <Veranstaltung> -Tag.

Der Bereich innerhalb des <xsl:for-each > -Tags wird für jede exportierte Veranstaltung einmal durchgeführt. Es ist also eine Schleife, die für jedes Veranstaltungs-Tag wiederholt wird (select="Veranstaltung"). Übersetzt: „Für jede Veranstaltung tue folgendes“.

Innerhalb der Schleife (xsl:for-each) finden Sie vier gleiche Tags mit jeweils unterschiedlichen Werten:

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

Durch diese <xsl:value-of >-Tags wird der Inhalt des entsprechenden XML-Tags der Exportdatei in das Ergebnis ausgegeben. <xsl:value-of select="_event_TITLE" /> ist also wie ein Platzhalter für den Titel der entsprechenden Veranstaltung.

In unserem Beispiel werden damit nacheinander von jeder Veranstaltung der Wochentag, Datum, Uhrzeit und Titel aneinandergereiht:

```
<xsl:value-of select="WOCHENTAG_START_LANG" />, <xsl:value-of select="START_DATUM" />  
<xsl:value-of select="START_UHRZEIT" /> Uhr: <xsl:value-of select="_event_TITLE" />
```

Vergleichen Sie bitte die blauen Bezeichnungen mit den Tags der XML-Exportdatei!

Bitte beachten: Da ein XML-Tag immer aus Anfangs- und Endtag bestehen muss, wird hier die Kurzform mit dem Schrägstrich am Ende verwendet: <tagname />

Das zwischen den Tags stehende Komma und „Uhr:“ wird in den Export unverändert als Text übernommen.

Stylesheet erweitern

Wenn Sie nun mit diesem Stylesheet einen Export machen, erhalten Sie folgendes:

```
Sonntag, 8.01.10.00 Uhr: GottesdienstSonntag, 15.01.10.00 Uhr:  
GottesdienstSonntag, 22.01.10.00 Uhr: Gottesdienst
```

Einen unformatierten Text ohne Zeilenumbrüche. Da bestimmte Leerzeichen automatisch entfernt werden, müssen wir den Zeilenumbruch gesondert einfügen. Dazu hilft uns eine besondere Variable mit der Bezeichnung „newline“, die wir vor unser Template einfügen:

```
<xsl:variable name="newline"><xsl:text>  
</xsl:text></xsl:variable>
```

Nur soviel: Die Variable enthält Text - und zwar einen Zeilenumbruch!

Diese Variable können wir nun im Template verwenden (\$-Zeichen beachten!):

```
<xsl:value-of select="WOCHENTAG_START_LANG" />, <xsl:value-of select="START_DATUM" />  
<xsl:value-of select="START_UHRZEIT" /> Uhr: <xsl:value-of select="_event_TITLE" />  
select="_event_TITLE" /><xsl:value-of select="$newline" />
```

Zusätzlich möchten wir noch die Kurzbeschreibung und den Ansprechpartner der jeweiligen Veranstaltung anzeigen lassen. Suchen Sie sich die entsprechenden Tag-namen aus der XML-Datei und fügen sie ein:

```
<xsl:value-of select="WOCHENTAG_START_LANG" />, <xsl:value-of select="START_DATUM" />  
<xsl:value-of select="START_UHRZEIT" /> Uhr: <xsl:value-of select="_event_TITLE" /> -  
<xsl:value-of select="_event_SHORT_DESCRIPTION" />, mit <xsl:value-of  
select="_person_NAME" /><xsl:value-of select="$newline" />
```

Ein Export damit (list2.txt.xsl) sieht nun so aus:

```
Sonntag, 8.01.10.00 Uhr: Gottesdienst - , mit Pfr. Marius Strecker  
Sonntag, 15.01.10.00 Uhr: Gottesdienst - mit musikalischer Gestaltung durch den  
Posaunenchor, mit Pfr. Miklós Geyer
```

Sonntag, 22.01.10.00 Uhr: Gottesdienst - , mit Pfr. Marius Strecker

So ganz optimal ist es noch nicht: Zwischen dem Datum und der Uhrzeit wird das Leerzeichen verschluckt und der Bindestrich vor der Kurzbeschreibung wird auch angezeigt, wenn keine Kurzbeschreibung vorhanden ist.

Der Umwandlungsprozessor entfernt Leerzeichen aufgrund der Anweisung `<xsl:strip-space elements="*" />`. Dies ist grundsätzlich sinnvoll, aber an manchen Stellen unerwünscht. Hier hilft uns wieder eine Variable,

```
<xsl:variable name="space">
<xsl:text> </xsl:text>
</xsl:variable>
```

die wir zwischen Datum und Uhrzeit einfügen können:

```
<xsl:value-of select="START_DATUM" /><xsl:value-of select="$space" /><xsl:value-of
select="START_UHRZEIT" />
```

Das zweite Problem lösen wir durch ein „Bedingungs-Tag“: Nur wenn in der Kurzbeschreibung ein Text steht, soll der Inhalt des Bedingungs-Tags ausgegeben werden. In XSL sieht dies so aus:

```
<xsl:if test="_event_SHORT_DESCRIPTION!=''" > - <xsl:value-of
select="_event_SHORT_DESCRIPTION" /></xsl:if>
```

Blau: Anfang und Ende der Bedingung. Alles innerhalb davon (rot) ist betroffen und wird nur ausgegeben, wenn die Bedingung wahr ist.

Grün: die Bedingung: wenn `_event_SHORT_DESCRIPTION` nicht leer ist. „Nicht gleich“ wird in XSL als `!=` notiert. In den einfachen Anführungszeichen ist kein Text. Es ist also wie eine doppelte Verneinung: wenn es nicht Nichts ist!

Ein anderes Beispiel zur Verdeutlichung: Der Titel soll nur ausgegeben werden, wenn er nicht ‚Gottesdienst‘ lautet: `_event_TITLE != 'Gottesdienst'`. Dieser Ausdruck muss in das `test`-Attribut (`test=""`) des `if`-Tags geschrieben werden: `test="_event_TITLE!='Gottesdienst'"`

Wenn Sie das Prinzip verstanden haben, hilft Copy und Paste um die verschiedensten Abfragen und Bedingungen in Ihre Stylesheets einzubauen.

Testen Sie nun das Stylesheet (list3.txt.xsl):

```
Sonntag, 8.01. 10.00 Uhr: Gottesdienst, mit Pfr. Marius Strecker
Sonntag, 15.01. 10.00 Uhr: Gottesdienstmit musikalischer Gestaltung durch den
Posaunenchor, mit Pfr. Miklós Geyer
Sonntag, 22.01. 10.00 Uhr: Gottesdienst, mit Pfr. Marius Strecker
```

Stylesheets mit Schriftformatierungen

Bisher hatten die Stylesheets jeweils einen Dateinamen in der Form *name.txt.xsl*. *name* kann hierbei eine beliebige Bezeichnung sein. Der zweite Bestandteil nach dem Punkt steuert die Dateieindung der Exportdatei. In den oberen Beispielen erhielten wir eine *.txt* - Datei. *xsl* ist die Dateieindung für XSL-Stylesheets und darf nicht verändert werden.

Wenn Sie eine CSV (Comma-separated-Values) Datei erstellen möchten, die mit z.B. Excel geöffnet werden kann, müssen Sie das Stylesheet *name.csv.xsl* benennen, wobei *name* eine beliebige Bezeichnung sein kann, z.B. also *meineListe.csv.xsl*.

Damit es eine echte CSV-Datei wird, müssen Sie sich natürlich darum kümmern, dass auch die richtigen Werte in die Datei geschrieben werden. Vergleichen Sie dazu die Stylesheets in der zip-Datei.

Eine Ausnahme bilden RTF (Rich-Text-Format) - Dateien, die Sie z.B. mit Word öffnen können. Wenn Sie eine RTF-Datei erstellen möchten, geben sie als 2. Namensbestandteil „html2rtf“ an, also z.B. *meinExport.html2rtf.xsl*.

Hintergrund dafür ist, dass die Formatierungsanweisungen für RTF-Dokumente in einem HTML-ähnlichen Stil geschrieben werden müssen, damit der Umwandlungsprozessor sie verarbeiten kann.

Schauen Sie sich nun bitte das Stylesheet list1.html2rtf.xsl aus dem Ordner Anleitung an:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output
  method="text"
  indent="no"
  encoding="ISO-8859-1"
  media-type="text/plain"
/>
<xsl:strip-space elements="*" />

<xsl:template match="/root/Export">
<!-- Definition von Kopf und Fusszeile -->
{header}Wochenzettel <xsl:value-of
select="Veranstaltung/_user_REALNAME[1]" />{/header}
{footer}{font color=#333333 size=8 face=arial}powered by www.evangelische-
termine.de{/font}{/footer}
<xsl:call-template name="veranstaltungen" />
</xsl:template>

<!-- Fügen Sie bei Bedarf weiter Datenbankfelder in die Ausgabe ein: xsl:value-of
-->

<xsl:template name="veranstaltungen">
  <xsl:for-each select="Veranstaltung">
    <xsl:variable name="last" select="START_DATUM" />

    <xsl:if test="not(preceding-sibling::Veranstaltung[START_DATUM=$last])">
      {p before=3}{font face=arial size=12}{b}<xsl:value-of
select="WOCHENTAG_START_LANG" />, <xsl:value-of select="START_DATUM" />{/b}{/font}
    </xsl:if>

    {br}{font face=arial size=12}
      <xsl:value-of select="START_UHRZEIT" /> Uhr
    {tab}<xsl:value-of select="_event_TITLE" />{/font}

  </xsl:for-each>
</xsl:template>
</xsl:stylesheet>
```

Die (violetten) Texte in <!-- .. --> sind lediglich Kommentare, die keine Auswirkung haben.

Die blauen „Tags“ in geschweiften Klammern steuern nun die Formatierungen des RTF-Dokuments. Die Terminologie ist an HTML angelehnt. Bitte beachten Sie: es ist wirklich nur eine Anlehnung! Die genaue Verwendung und Schreibweise muss beachtet werden, sonst bricht der Umwandlungsprozessor seine Arbeit mit einer Fehlermeldung ab oder das Dokument kann nicht geöffnet werden.

Ein ausführliche Beschreibung der einzelnen Tags mit ihren Möglichkeiten finden Sie in der Datei RTF-Tags.pdf.

Unser Stylesheet verwendet {p} für einen Absatz und {br} für einen Zeilenumbruch. Diese beiden Tags benötigen **kein** Endtag. Text innerhalb von {b} und {/b} wird fett dargestellt. Mit dem {font} - Tag kann Schriftart und -größe festgelegt werden. {header} ... {/header} und {footer} ... {/footer} sind vermutlich selbsterklärend. Sie können aber nur im Kopfbereich des Stylesheets verwendet werden.

Viel Erfolg!

Bitte beachten Sie immer auch unser Forum im Intranet. Dort können Sie Anfragen zu Stylesheets stellen, damit auch andere davon profitieren können.