

# Datenexport beim Druck

Mit der Version vom 04.01.2019 wird eine Exportfunktion für alle bestehenden und zukünftige Druckerzeugnisse eingeführt. Damit können sämtliche Daten, die in den XML-Strukturen der Druckerzeugnisse enthalten sind, in CSV oder einem anderen Textformat in Dateien exportiert werden. Diese können dann anschließend z.B. mit Excel weiterverarbeitet werden.

Um den FlexNow-Nutzern eine größtmögliche Freiheit bzgl. Inhalt und Format der zu exportierenden Daten zu geben, erfolgt dies analog zu den PDF-Druckerzeugnissen anhand von Exportvorlagen in Form von XSL-Dateien.

## Konfiguration im PO-Modul

Die Einstellung von Exportvorlagen für ein bestimmtes Druckerzeugnis erfolgt im PO-Modul unter dem Punkt „Einstellungen - Druckerzeugnisse“. Dabei wird die Exportvorlage als neues Druckformat unterhalb des jeweiligen Druckerzeugnisses angelegt.



Im Dialog der Druckformate gibt es nun neue Einstellungsoptionen für den Export.

Im Eingabefeld **XSL-Datei** wird für ein Exportformat der Dateiname der XSL-Exportvorlage angegeben. Diese Datei ist aber im Gegensatz zu den PDF-Vorlagen nicht auf dem Druckserver gespeichert, sondern standardmäßig im Unterverzeichnis „xsl“ des Ordners, in dem die fn2Module ausgeführt werden. Dieses Verzeichnis kann aber auch mit der Setup-Einstellung **DRUCK\_VERZEICHNIS\_XSL** angepasst werden.

The screenshot shows a dialog box titled "Druckformat" with a close button (X) in the top right corner. It has two tabs: "Allgemein" (selected) and "Zuordnung". Under the "Allgemein" tab, there are several input fields and a radio button group. The "Bezeichnung" field contains "Export CSV" and has a small flag icon to its right. The "Sprache" field is a dropdown menu showing "Deutsch". The "XSL-Datei" field contains "datenblatt\_csv.xsl". Below these is a "Typ" section with two radio buttons: "Druck (Pdf)" and "Export (CSV)", with "Export (CSV)" being selected. To the right of the radio buttons is a "Dateiendung" field containing "csv". At the bottom of the dialog are two buttons: "Ok" with a green checkmark icon and "Abbrechen" with a red X icon.

Zusätzlich muss im Bereich **Typ** eingestellt werden, dass es sich um einen Export handelt und noch die **Dateiendung** vorgegeben werden, die Export-Dateien erhalten sollen. (z.B. csv oder txt).

## XSL-Vorlagen

Welche Dateien exportiert werden und wie diese formatiert sind, wird in der XSL-Datei für den Export bestimmt. Mit dem Update vom 04.01.2019 werden mehrere Beispieldateien für unterschiedliche Druckerzeugnisse mit ausgeliefert. Diese können aus Ausgangspunkt für neu zu erstellende Vorlagen verwendet werden.

Der grundsätzliche vorgeschlagene Aufbau einer XSL-Exportvorlage ist folgender (die anzupassenden Bereiche sind farbig markiert):

```
<HTML> <?xml version=„1.0“ encoding=„ISO-8859-1“?> <xsl:stylesheet
xmlns:xsl=„http://www.w3.org/1999/XSL/Transform“ version=„1.0“> <xsl:output method=„text“
omit-xml-declaration=„yes“ indent=„no“/>
```

```
<xsl:template match=„/“><xsl:call-template name=„header“/> <xsl:for-each select=„###Pfad zum
Knoten der zu exportierenden Datensätze###“> <xsl:apply-templates select=„.“/><xsl:text>
</xsl:text> </xsl:for-each> </xsl:template>
```

```
<xsl:template name=„header“> ###Ausgabe der Kopfzeile### </xsl:template>
```

```
<xsl:template match=„###Knotentyp###“> ###Ausgabezeile der zu exportierenden Daten###
</xsl:template>
```

```
</xsl:stylesheet> </HTML>
```

Für den Export wird die gleiche XML-Datenstruktur generiert wie für die entsprechenden Druckerzeugnisse, so dass in der Export-Vorlage die gleichen Zugriffspfade auf die XML-Struktur wie in den XSL-Dateien der PDF-Druckerzeugnisse verwendet werden können. Um alle für den Export verfügbaren Daten anzuzeigen kann ebenso wie bei der Erstellung von XSL-Dateien für den Druck das System Log verwendet werden.

Beispiel einer einfachen Exportvorlage für einen Report:

```
<HTML> <?xml version="1.0" encoding="ISO-8859-1"?> <xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"> <xsl:output method="text"
omit-xml-declaration="yes" indent="no"/>

<xsl:template match="/"><xsl:call-template name="header"/> <xsl:for-each
select="XMLMsg/Body/ActionCall/ReportParam/Report/Data/Dataset"> <xsl:apply-templates
select="."/><xsl:text> </xsl:text> </xsl:for-each> </xsl:template>

<xsl:template name="header"> MatrNr;Nachname;Vorname </xsl:template>

<xsl:template match="Dataset"> <xsl:value-of select="sMatrNr"/>;<xsl:value-of
select="sNachname"/>;<xsl:value-of select="sVorname"/> </xsl:template>

</xsl:stylesheet> </HTML>
```

## Individuelle Datenbankabfragen für den Export

Sollen im Export auch Daten ausgegeben werden, die nicht in der XML-Struktur enthalten sind, besteht die Möglichkeit individuelle Datenbankabfragen in Form von SQL-Statements in die XSL-Vorlage zu integrieren.

Hierfür stehen beim Export (und auch bei den XSL-Vorlagen für EMail) zwei neue XSL-Funktionen zur Verfügung:

- ext:SqlSingle(statement) - Hiermit kann ein einzelnes Attribut gelesen und direkt ausgegeben werden
- ext:Sql(statement) - Diese Funktion liefert das Ergebnis einer Abfrage als XML-Struktur generiert.

Die Verwendung beider Funktionen soll das folgende Beispiel verdeutlichen:

```
<HTML> <?xml version="1.0" encoding="ISO-8859-1"?> <xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" xmlns:ext="http://ihb-eg.de/xsl">
<xsl:output method="text" omit-xml-declaration="yes" indent="no"/>

<xsl:template match="/"><xsl:call-template name="header"/> <xsl:for-each
select="XMLMsg/Body/ActionCall/ReportParam/Report/Data/Dataset"> <xsl:apply-templates
select="."/><xsl:text> </xsl:text> </xsl:for-each> </xsl:template>

<xsl:template name="header"> MatrNr;Nachname;Vorname;Geburtsdatum;Studienabschnitte...
</xsl:template>

<xsl:template match="Dataset"> <xsl:variable name="sql1">select gdatum from student where
```

```
matrnr=<xsl:value-of select=„nMatrNr“/></xsl:variable> <xsl:variable name=„sql2“>
```

```
select abschnitt.bez, studabver.bez as version
from studentver, abschnitt, studabver
where studentver.matrnr=<xsl:value-of select="nMatrNr"/>
and studentver.studab = abschnitt.studab
and studentver.studfach = studabver.studfach
and studentver.studab = studabver.studab
and studentver.studabver = studabver.studabver
```

```
</xsl:variable> <xsl:value-of select=„sMatrNr“/>;<xsl:value-of select=„sNachname“/>;<xsl:value-of
select=„sVorname“/>;<xsl:value-of select=„ext:SqlSingle($sql1)“/> <xsl:for-each
select=„ext:Sql($sql2)“>;<xsl:value-of select=„bez“/> (<xsl:value-of select=„version“/>)</xsl:for-
each> </xsl:template>
```

```
</xsl:stylesheet> </HTML>
```

Diese Exportvorlage wird für eine XML-Struktur genutzt, die nur Matrikelnummern und Namen von Studierenden liefert, aber keine weiteren Angaben zur Person oder deren Studienverlauf. Der Export soll nun aber zusätzlich um das Geburtsdatum und die belegten Studienabschnittsversionen ergänzt werden.

Zunächst ist es erforderlich, dass im Stylesheet-Tag (hier in der 2. Zeile) die Angabe für den Namespace „ext“ mit angegeben wird. Dies erfolgt durch den zusätzlichen Parameter `*xmlns:ext=„http://ihb-eg.de/xsl“`. Ohne diesen Parameter können die Funktionen nicht verwendet werden bzw. es führt zu einer Fehlermeldung.

Da die Sql-Abfragen umfangreicher sein können und auch darin auf Daten der XML-Struktur verwiesen werden kann, ist es ratsam, dass für jede Abfrage eine eigene XSL-Variable definiert wird. Dies erfolgt im Beispiel für zwei Abfragen in den folgenden Zeilen:

```
<HTML> <xsl:variable name=„sql1“>select gdatum from student where matrnr=<xsl:value-of
select=„nMatrNr“/></xsl:variable>
```

```
<xsl:variable name=„sql2“>
```

```
select abschnitt.bez, studabver.bez as version
from studentver, abschnitt, studabver
where studentver.matrnr=<xsl:value-of select="nMatrNr"/>
and studentver.studab = abschnitt.studab
and studentver.studfach = studabver.studfach
and studentver.studab = studabver.studab
and studentver.studabver = studabver.studabver
```

```
</xsl:variable> </HTML>
```

Die erste Abfrage `*sql1*` liefert das Geburtsdatum der Studierenden. Hier ist auch zu sehen, wie diese Abfrage mit der Matrikelnummer aus der XML-Struktur parametrisiert wird. Die Ausgabe des Geburtsdatums erfolgt nun mit der Funktion `ext:SqlSingle`:

```
<HTML> <xsl:value-of select=„ext:SqlSingle($sql1)“/> </HTML>
```

Die zweite Abfrage \*sql2\* ermittelt die Bezeichnungen der belegten Studienabschnitte und Versionen. Damit selektiert diese Abfrage sowohl mehrere Attribute, wie auch mehrere mögliche Datensätze als Resultat. Daher wird hierfür die Funktion ext:Sql verwendet, die z.B. folgendes Resultat wiederum als XML-Struktur liefert (dies wird zur Kontrolle auch im System Log ausgegeben):

<HTML> <sql>

```
<row>
  <bez>Scheine (Grundstudium)</bez>
  <version>Fachprüfungsordnung WInf (ohne Studienschwerpunkt
Medieninformatik)</version>
</row>
<row>
  <bez>Diplomvorprüfung</bez>
  <version>Fachprüfungsordnung WInf (ohne Studienschwerpunkt
Medieninformatik)</version>
</row>
```

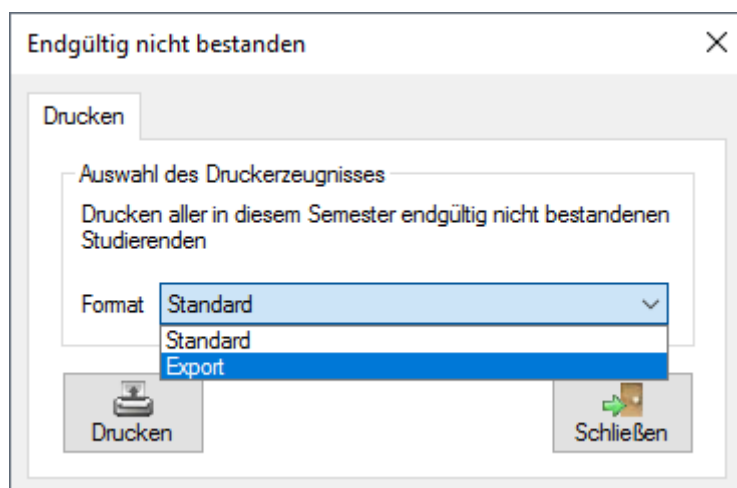
</sql> </HTML>

Über diese XML-Struktur kann nun mit einer for-each-Schleife iteriert und die gewünschten Attribute ausgegeben werden (im Beispiel mit Semikolon getrennt). Dabei ist zu beachten, dass sich die for-each-Schleife bereits auf das row-Tag bezieht, d.h. der Zugriff auf untergeordneten Attribute kann ohne Pfadangabe erfolgen.

<HTML> <xsl:for-each select=„ext:Sql(\$sql2)“>;<xsl:value-of select=„bez“/> (<xsl:value-of select=„version“/>)</xsl:for-each> </HTML>

## Export durchführen

Der Export der Daten erfolgt im Druckdialog des entsprechenden Druckerzeugnisses. Dazu muss in der Auswahlliste der Druckerzeugnisse das gewünschte Exportformat ausgewählt und dann auf den Button **Drucken** geklickt werden.



Dadurch öffnet sich ein Dateidialog, in dem das Verzeichnis und der Dateiname der zu exportierenden

Datei festgelegt werden kann. Anschließend wird die Exportdatei erstellt und in Abhängigkeit von der eingestellten Dateierweiterung wird diese dann automatisch in einem geeigneten Programm geöffnet (sofern eines installiert ist). Bei der Endung csv ist dies üblicherweise Excel, bei txt der Texteditor, der mit dieser Endung in den Windows-Einstellungen verknüpft ist. Im diesem Programm können dann die exportierten Daten weiter bearbeitet werden.

Standardmäßig wird zur Speicherung der Exportdateien das Unterverzeichnis „export“ im FlexNow-Ordner angeboten. Es ist jedoch zu beachten, dass in diesem Verzeichnis Dateien, die älter als 12 Stunden sind, automatisch gelöscht werden. Wenn sie die Daten länger behalten möchten, dann wählen Sie bitte ein anderes Verzeichnis aus. Dieses neu ausgewählte Verzeichnis wird dann auch automatisch beim nächsten Export angeboten.

From:

<https://wiki.ihb-eg.de/> - **FlexWiki**

Permanent link:

<https://wiki.ihb-eg.de/doku.php/export/start?rev=1546878275>

Last update: **2019/01/07 17:24**

