

# **System Concept DMS**

## **Einrichtung von Exports**

**Peter Pinnau**

---

# **System Concept DMS: Einrichtung von Exports**

Peter Pinnau

Version 1.0

Veröffentlicht 24.7.2018

---

---

# Inhaltsverzeichnis

1. Einleitung .....	1
2. Erstellung und Arbeitsweise eines Text Exports .....	2
3. Beschreibung XML Vorlagenformat .....	3
3.1. export.xml .....	3
3.2. indexexport.xml .....	4
3.3. biz.pinnau.dms.export.DatevExport .....	12

---

# Kapitel 1. Einleitung

Alle reden davon, nur wenige haben es: **Das papierlose Büro** - oder zumindest eine umfassende Digitalisierung und systematische Ablage von Dokumenten. Unzählige Document Management Systeme (DMS) verschiedener Hersteller versprechen dazu die richtige Software zu liefern. Einige halten was sie versprechen.

So gut wie alle DMS Produkte schaffen aber eine Blackbox: Die Dokumente verschwinden im System und sind in der Regel nur noch zusammen mit der Software nutzbar. Der Kunde geht eine starke Abhängigkeit ein.

**System Concept DMS geht hier andere Wege – bessere Wege!**

Diese Dokumentation richtet sich an Systemverwalter und Administratoren. Sie beschreibt, wie mit Hilfe der in System Concept DMS verschlagworteten Dokumente **Exports** eingerichtet werden.

**Exports** sind in der Regel CSV-basierte Textdateien, mit denen Daten in andere IT-Systeme übernommen werden können. Die **System Concept GmbH** hat mit der Export-Funktion Buchungsübergaben für Eingangsrechnungen mit Kontierung in verschiedene Buchführungssysteme realisiert.

---

# Kapitel 2. Erstellung und Arbeitsweise eines Text Exports

Zur Erstellung von Text-Exporten werden folgende Schritte durchlaufen:

- Sammeln der zu exportierenden Dokumente (z.B. Eingangsrechnungen) in einem Ordner. Dies kann z.B. ein spezieller Export-Ordner auf dem Schreibtisch sein, in den die Dokumente verschoben und dort gesammelt werden.
- Um den Export zu starten, markieren Sie den entsprechenden Ordner und wählen Sie **Datei - Exportieren**. Es wird ein Assistent gestartet, der Sie durch den Exportvorgang führt. Weitere Informationen finden Sie im **Benutzerhandbuch** unter 11. Exportieren von Dokumenten
- Nach Fertigstellen des Export-Assistenten wird die Export-Datei an der in der XML-Vorlage konfigurierten Speicherposition erstellt. Alle exportierten Dokumenten erhalten einen entsprechenden Export-Eintrag im **Feed** (Dokumenthistorie).

Die Konfiguration der **Exports** wird bisher noch nicht durch die Oberfläche von **System Concept DMS** unterstützt. Die Bearbeitung der XML-Dateien muss daher in einem Texteditor erfolgen. Weitere Informationen zum XML-Format finden Sie unter Kapitel 3, *Beschreibung XML Vorlagenformat*.

Jeder Export wird durch eine oder mehrere XML-Dateien konfiguriert. Die XML-Dateien befinden sich im Verzeichnis **{instance.home}/configuration/export**.

Um einen neuen Export einzurichten, erstellen Sie zunächst einen entsprechenden Unterordner im Verzeichnis **{instance.home}/configuration/export**. Der Name ist frei wählbar und sollte einen Rückschluss auf den Zweck des Exports zulassen.

Erstellen Sie im neu angelegten Verzeichnis die Datei **export.xml** bzw. kopieren Sie diese von einem bereits vorhandenen Export. Die Datei **export.xml**.

Erstellen Sie im neu angelegten Verzeichnis zusätzlich die Datei **indexexport.xml** bzw. kopieren Sie diese von einem bereits vorhandenen Export. Die Datei **export.xml**.

Weitere Informationen zum XML-Format finden Sie unter Kapitel 3, *Beschreibung XML Vorlagenformat*.

---

# Kapitel 3. Beschreibung XML Vorlagenformat

Jeder Text-Export wird durch ein oder mehrere XML-Vorlagendefinition beschrieben. Die XML-Dateien befinden sich im Verzeichnis `{instance.home}/configuration/export/EXPORT_ID`. Der Aufbau der XML-Dateien wird nachfolgend beschrieben.

## 3.1. export.xml

Die Datei `export.xml` enthält folgende Informationen:

- **name**: Bezeichnung des Exports. Dieser wird in der Oberfläche angezeigt.
- **exporter**: zu verwendender Exporter. Momentan verfügbar sind:
  - biz.pinnau.dms.export.IndexBasedExport**: Erstellung CSV-Export aus Verschlagwortung
  - biz.pinnau.dms.export.DatevExport**: CSV-Export für DATEV
- **docType**: Dokumentart für die der Export verfügbar ist.
- **charset**: Zeichensatz der zu erzeugenden Export-Datei (z.B. utf8, iso8859-15). Falls es beim Import der Daten in das Zielsystem zu Problemen mit Umlauten bzw. Sonderzeichen kommt, kann über diesen Parameter der Zeichensatz eingestellt werden.

Das folgende Listing zeigt eine beispielhafte `export.xml` Datei:

```
<export>
  <name>ER für Buchhaltung</name>
  <exporter>biz.pinnau.dms.export.IndexBasedExport</exporter>
  <docType>ER</docType>
  <charset>iso8859-15</charset>
</export>
```

## 3.2. indexexport.xml

Die Datei **indexexport.xml** wird von den Exportern **biz.pinnau.dms.export.IndexBasedExport** bzw. dem darauf aufbauenden **biz.pinnau.dms.export.DatevExport** benötigt. In dieser Datei werden Regelwerke zur Erstellung der CSV-Datei konfiguriert (ähnlich der Ablageregeln zur Archivierung).

Das folgende Listing zeigt eine beispielhafte indexexport.xml. Bitte beachten Sie, dass header, document und relation noch keine Regeln enthalten.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<export>
  <fileName>X:\rewe\dms_er_${USER}_${TIMESTAMP}.csv</fileName>

  <lineFeed>win</lineFeed>

  <quote>"</quote>

  <header>
    ...
  </header>

  <header>
    ...
  </header>

  <document>
    ...
  </document>

  <relation>
    ...
  </relation>
</export>
```

**fileName** : Vollständiger Zielpfad für die Export-Datei. `{USER}` wird mit dem Benutzernamen des angemeldeten Benutzers ersetzt. `{TIMESTAMP}` wird durch das aktuelle Datum + Uhrzeit im Format `JAHR_MONAT_TAG_STUNDE_MINUTE` ersetzt.

**lineFeed**, optional: zu verwendender Zeilenbruch. Mögliche Werte sind **win** (CR+LF), **unix** (LF) oder **system**. Falls **lineFeed** nicht angegeben wird gilt die Einstellung **system**, d.h. auf Windows-Systemen wird CR+LF und auf Unix/Linux LF als Zeilenbruch verwendet.

**quote**, optional: Zeichen zum Einschliessen von Zeichenketten. Verwenden Sie „-Elemente in den entsprechenden Bereichen (header, document, relation), um ein Quote-Zeichen auszugeben.

**header**, optional, default = " : Bildungsvorschrift für Kopfzeile(n) der Exportdatei. Es können ein oder mehrere Header angegeben werden. Die Header sind unabhängig von den exportierten Dokumenten und werden nur ein Mal ausgegeben. Header können z.B. Spaltenbeschriftungen enthalten.

**document** : Bildungsvorschrift für Dokumentzeile. Für jedes zu exportierende Dokument wird genau 1 Zeile erzeugt.

**relation**, optional : Bildungsvorschrift für Relationszeilen. Relationen sind 1:N Beziehungen innerhalb der Verschlagwortung. Im Szenario Eingangsrechnungen ist z.B. die Kontierung der Rechnung in N Einzelpositionen eine Relation.

Die Relationszeilen werden pro Dokument 1 Mal für jede Zeile in der entsprechenden Relation ausgegeben. Bei der Kontierung von Rechnungen wird pro Kontierungsposition eine Zeile erzeugt.

## 1. Header

Im optionalen Header können Sie keine Verschlagwortungsfelder ausgeben, da der Header unabhängig von den zu exportierenden Dokumenten genau 1 Mal ausgegeben wird. Es können mehrere Header angegeben werden, die dann in der Reihenfolge wie sie in der XML-Datei notiert sind zu Beginn des Export-Files ausgegeben werden. Falls ihr Export-File keinen Header benötigt lassen Sie das **header**-Element einfach weg.

Verwenden Sie **constant** und **variable** Elemente, um konstante Texte oder Variablen auszugeben. Verwenden Sie **quote** Elemente, um Zeichenkettenbegrenzer auszugeben. Das folgende Beispiel zeigt, wie eine headerzeile gebildet werden kann:

```
<header>
  <!-- HF = Header Fremdbuchung -->
  <constant>
    <text>HF</text>
  </constant>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <!-- Übergabenummer (aus HF) -->
  <variable>
    <name>export.pseudo.id</name>
  </variable>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <!-- Erstellungsdatum der Datei -->
  <variable>
    <name>export.date</name>
    <format>dd.MM.yyyy</format>
    <formatter>java.time.format.DateTimeFormatter</formatter>
  </variable>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <!-- Name des übergebenen Programms -->
  <quote />
  <constant>
    <text>DMS</text>
  </constant>
  <quote />
</header>
```

Das Beispiel würde folgende Headerzeile generieren:



HF;812112221177;22.06.2018;"DMS"

Folgende Variablen können im Header genutzt werden:

- **export.pseudo.id** : Zufällig erzeugte ID für den Export. Die ID besteht aus maximal 12 Ziffern (0-9)
- **export.date** : Datum+Uhrzeit der Export-Erstellung. Zur Ausgabe muss ein Formatter+Format angegeben werden (siehe XML-Listing oben)
- **export.user** : Benutzer der den Export ausführt.

## 2. Document

Im **document** Bereich können Sie auf Verschlagwortungsfelder des Dokuments zugeifen. Für jedes Dokument wird genau eine Zeile im Export generiert. Verwenden Sie **constant**, **rule** und **variable** Elemente, um konstante Texte, Indexfelder oder Variablen auszugeben. Verwenden Sie **quote** Elemente, um Zeichenkettenbegrenzer auszugeben. Das folgende Beispiel zeigt, wie eine Document-Zeile gebildet werden kann:

```
<document>
  <!-- HB = Header Buchung -->
  <constant>
    <text>HB</text>
  </constant>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <!-- Übergabenummer (aus HF) -->
  <variable>
    <name>export.pseudo.id</name>
  </variable>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <variable>
    <name>export.document.index</name>
  </variable>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <!-- Konto (Kreditor-Nr) -->
  <rule>
    <documentClass>ER</documentClass>
    <indexField>Kunden-Nr</indexField>
  </rule>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <!-- Belegdatum -->
  <rule>
    <documentClass>DMS</documentClass>
    <indexField>Datum</indexField>
    <format>dd.MM.yyyy</format>
    <formatter>java.time.format.DateTimeFormatter</formatter>
  </rule>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <!-- Buchungstext -->
  <quote />
  <rule>
    <documentClass>ER</documentClass>
    <indexField>Betreff</indexField>
  </rule>
  <quote />
```

```
</document>
```

Das Beispiel würde für 2 Rechnungsdokumente folgende Zeilen generieren:

```
HB;812112221177;1;70001;10.06.2018;"Büromaterial"  
HB;812112221177;2;70050;11.06.2018;"Wartung Fahrstuhl"
```

Folgende Variablen können im **Document** genutzt werden:

- **export.pseudo.id** : siehe Header
- **export.date** : siehe Header
- **export.user** : siehe Header
- **export.document.index** : Nummer des Dokuments im Export beginnend bei 1

### 3. Relation

Im **relation** Bereich können Sie auf Verschlagwortungsfelder des Dokuments und einer Relation zugeifen. Pro Dokument wird für jede Position einer Relation genau eine Zeile im Export generiert. Relationen sind optional. Falls ihr Export-File keine Relationen benötigt lassen Sie das **relation**-Element einfach weg.

Mit dem Attribut **name** muss die ID der Relation angegeben werden (wie in der Verschlagwortung eingerichtet). Verwenden Sie **constant**, **rule** und **variable** Elemente, um konstante Texte, Indexfelder oder Variablen auszugeben. Verwenden Sie **quote** Elemente, um Zeichenkettenbegrenzer auszugeben. Das folgende Beispiel zeigt, wie eine Relation-Zeile gebildet werden kann:

```
<relation>
  <name>Kontierung</name>

  <!-- BP = Buchungsposition -->
  <constant>
    <text>BP</text>
  </constant>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <!-- Übergabenummer (aus HF) -->
  <variable>
    <name>export.pseudo.id</name>
  </variable>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <variable>
    <name>export.document.index</name>
  </variable>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <variable>
    <name>export.relation.index</name>
  </variable>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <!-- Sachkonto (Sachkonto-Nr) -->
  <rule>
    <documentClass>ER</documentClass>
    <indexField>Sachkonto-Nr</indexField>
  </rule>

  <constant>
    <text>{file.seperator}</text>
  </constant>

  <!-- Betrag -->
  <rule>
    <documentClass>ER</documentClass>
    <indexField>Betrag Position</indexField>
    <format>0.00</format>
    <formatter>java.text.DecimalFormat</formatter>
  </rule>

  <constant>
    <text>{file.seperator}</text>
  </constant>
```

Das Beispiel würde für 1 Rechnungsdokument mit einer Kontierung aus 2 Positionen folgende Zeilen generieren:

```
BP;812112221177;1;1;4711;560.00;"Büromaterial"  
BP;812112221177;1;2;4812;100.00;"Telefon"
```

Folgende Variablen können in der **Relation** genutzt werden:

- **export.pseudo.id** : siehe Header
- **export.date** : siehe Header
- **export.user** : siehe Header
- **export.document.index** : siehe Document
- **export.relation.index** : Nummer der Position beginnend bei 1. Zähler wird bei einem neuen Dokument zurück gesetzt.

Die Kombination aus den beispielhaften **header**, **document** und **relation**-Elementen erzeugt aus 2 Rechnungsdokumenten einen Export in folgender Form:

```
HF;812112221177;22.06.2018;"DMS"  
HB;812112221177;1;70001;10.06.2018;"Büromaterial"  
BP;812112221177;1;1;4711;560.00;"Büromaterial"  
BP;812112221177;1;2;4812;100.00;"Telefon"  
HB;812112221177;2;70050;11.06.2018;"Wartung Fahrstuhl"  
BP;812112221177;2;1;4400;200.00;"Erhaltungsaufwand"  
BP;812112221177;2;2;4500;50.00;"Telefon"  
BP;812112221177;2;3;4600;50.00;"Sonstige Aufwendungen"
```

## 4. Verarbeitung von Null-Werten

Falls Sie auf Verschlagwortungsfelder zugreifen, die Null-Werte enthalten können, können Sie für diese mit Hilfe von **biz.pinnau.dms.archive.NotNullFormat** eine spezielle Behandlung einrichten:

```
<!-- Steuerart (1 = ohne Steuer, 2 = USt, 3 = Vorsteuer)  
entweder 1 oder 3, je nach Steuersatz  
biz.pinnau.dms.archive.NotNullFormat (3|1)  
  
Steuersatz != null => 3  
Steuersatz = null => 1  
-->  
<rule>  
  <documentClass>ER</documentClass>  
  <indexField>Steuersatz</indexField>  
  <format>3|1</format>  
  <formatter>biz.pinnau.dms.archive.NotNullFormat</formatter>  
</rule>
```

Das Beispiel setzt das Feld **Steuerart** im Export auf **3 (Vorsteuer)**, falls das Dokument einen Steuersatz hat. Falls das Dokument keinen Steuersatz hat wird im Export der Wert **1 (Ohne Steuer)** eingetragen.

## 5. Formatter

Bei der Verwendung von **rule**-Elementen zum Auslesen von Indexfeldern können Sie mit Hilfe der Attribute **formatter** und **format** eine Formatierung des Wertes konfigurieren.

Sie können dabei grundsätzlich alle Formate nutzen, die auch bei der Einrichtung der Ablageregeln zur Verfügung stehen. Siehe Handbuch **Installation und Administration** Kapitel 9.3.

## 3.3. biz.pinnau.dms.export.DatevExport

Der Exporter **biz.pinnau.dms.export.DatevExport** ist eine Erweiterung von **biz.pinnau.dms.export.IndexBasedExport**, d.h. alle oben genannten Eigenschaften gelten analog. Zusätzlich stellt der DATEV-Exporter folgende Variablen bereit:

- **VAR\_EXPORT\_BEGINN\_WJ**: Datum an dem das Wirtschaftsjahr der übergebenen Belege beginnt. Momentan wird nur der 1.1. unterstützt. Das Jahr wird aus dem ersten zu exportierenden Dokument ermittelt (Feld DMS.Datum). Falls weitere zu exportierende Dokumente in ein anderes Wirtschaftsjahr bricht der Export mit einer entsprechenden Fehlermeldung ab.
- **VAR\_EXPORT\_STAPEL\_VON**: Kleinstes Beleg-Datum (der zu exportierenden Dokumente)
- **VAR\_EXPORT\_STAPEL\_BIS**: Größtes Beleg-Datum (der zu exportierenden Dokumente)

Diese Variablen werden für das DATEV-Format benötigt.