

Anpassen der XSLT-Transformation in SAP

Beitrag von „Claus Wild“ vom 11. Januar 2013, 18:02

In SAP wird eine CGI-Transformation für CAMT.053 und CAMT.054 ausgeliefert. Diese passt allerdings oftmals nicht in Verbindung mit den Banken und füllt die Tabellenfelder nicht korrekt. Somit muss ggf. pro Bank eine eigene Transformation im eigenen Namensraum angelegt und angepasst werden.

Beispiel: Geschäftsvorfallcode (GVC)

Eine Beispielauslieferung der Deutschen Kreditwirtschaft (DK) liefert den GVC wie folgt aus:

```
<!--Gutschrift aufgrund eines SEPA-Ueberweisungseinganges-->
- <Ntry>
  <Amt Ccy="EUR">100.00</Amt>
  <CdtDbtInd>CRDT</CdtDbtInd>
  <Sts>BOOK</Sts>
  - <BookgDt>
    <Dt>2016-05-11</Dt>
  </BookgDt>
  - <ValDt>
    <Dt>2016-05-11</Dt>
  </ValDt>
  <AcctSvcrRef>Bankreferenz</AcctSvcrRef>
  - <BkTxCd>
    - <Domn>
      <Cd>PMNT</Cd>
    - <Fmly>
      <Cd>RCDT</Cd>
      <SubFmlyCd>ESCT</SubFmlyCd>
    </Fmly>
    </Domn>
  - <Prtry>
    <Cd>166</Cd>
    <Issr>DK</Issr>
  </Prtry>
</BkTxCd>
- <NtryDtls>
  - <TxDtls>
    - <Refs>
      <EndToEndId>Ende-zu-Ende-Id des Ueberweisenden</EndToEndId>
    -
```

```

<!--Transaction Code-->
<VGEXT>
  <xsl:choose>
    <xsl:when test="ns0:BkTxCd/ns0:Domn">
      <xsl:value-of select="concat(ns0:BkTxCd/ns0:Domn/ns0:Cd, ns0:BkTxCd/ns0:Domn/ns0:Fmly/ns0:Cd, ns0:BkTxCd/ns0:Domn/ns0:Fmly/ns0:SubFmlyCd)"/></xsl:value-of>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="ns0:BkTxCd/ns0:Prtry/ns0:Cd"/></xsl:value-of>
    </xsl:otherwise>
  </xsl:choose>
</VGEXT>

```

In der Beispieldatei wird der GVC mit **NTRF+166** in das Tabellenfeld **VGEXT** geschrieben. Bestehende Einträge werden somit bei der Interpretation nicht mehr angesprochen. Die Transformation kann nun beispielsweise wie folgt angepasst werden:

```

<!--Transaction Code-->
<VGEXT>
  <xsl:choose>
    <xsl:when test="ns0:BkTxCd/ns0:Domn">
      <xsl:value-of select="concat(ns0:TxDtIs/BkTxCd/ns0:Domn/ns0:Cd, ns0:BkTxCd/ns0:Domn/ns0:Fmly/ns0:Cd, ns0:BkTxCd/ns0:Domn/ns0:Fmly/ns0:SubFmlyCd)"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="substring(ns0:NtryDtIs/ns0:TxDtIs/ns0:BkTxCd/ns0:Prtry/ns0:Cd,1,3)"/>
    </xsl:otherwise>
  </xsl:choose>
</VGEXT>

```

Die Position des GVC wird ausgelesen und die drei Stellen des GVC werden in das Feld VGEXT geschrieben.

```

- <ITEMS_FEB>
  - <ITEM_FEB>
    - <FEBEP>
      <ESNUM/>
      <BVDAT>2013-11-06</BVDAT>
      <BUDAT/>
      <VALUT>2013-11-06</VALUT>
      <BUTIM/>
      <KWAER>EUR</KWAER>
      <KWBTR>221.87</KWBTR>
      <SPESF/>
      <VGEXT>833</VGEXT>

```

Beitrag von „EC1055“ vom 10. September 2014, 13:05

Hallo Herr Wild,

wenn die substring-Funktion den Aufbau "substring(string,start,len)" hat, müsste es dann in der angepassten Transformation in Zeile 205 für einen **dreistelligen** GVC nicht "substring(...,6,**3**)" heißen?

Viele Grüße
EC1055

Beitrag von „Claus Wild“ vom 10. September 2014, 18:54

Hallo EC1055,

hmm, das muss ich mir nochmals anschauen - nicht das ich die falschen Screens eingebaut habe :-(
Wird aber Ende der Woche werden ...

Viele Grüße

Claus Wild

Beitrag von „Claus Wild“ vom 12. September 2014, 15:15

Hallo EC1055,

ja, es war der falsche Screen (6,3) ist an der Stelle richtig ...

```
<!--Transaction Code-->
<VGEXT>
  <xsl:choose>
    <xsl:when test="ns0:BkTxCd/ns0:Down">
      <xsl:value-of select="concat(ns0:TxDtIs/BkTxCd/ns0:Down/ns0:Cd, ns0:BkTxCd/ns0:Down/ns0:Fmly/ns0:Cd, ns0:BkTxC
    </xsl:when>
    <xsl:otherwise>
      <xsl:value-of select="substring(ns0:NtryDtIs/ns0:TxDtIs/ns0:BkTxCd/ns0:Prtry/ns0:Cd,6,3)"/>
    </xsl:otherwise>
  </xsl:choose>
</VGEXT>
```

Viele Grüße

Claus Wild

Beitrag von „sap_claudi“ vom 29. Februar 2016, 11:35

Hallo Herr Wild,

wann läuft die Transformation eigentlich in den "Test"-Zweig
<xsl:when test="ns0:BkTxCd

Seltsamerweise scheint das beim österreichischen XML-Format so zu sein...

Vielen Dank schonmal für Ihre Hilfe !

Viele Grüße,
sap_claudi

Beitrag von „Claus Wild“ vom 29. Februar 2016, 12:16

Hallo sap_claudi,

das wird durchlaufen bis ein test-Attribut für das Element die Bedingungen in dem Quelldokument beschreibt oder erreicht wird.

Viele Grüße

Claus Wild

Beitrag von „sap_claudi“ vom 2. März 2016, 12:58

Hallo Herr Wild,

vielen Dank für die Rückmeldung. Hätten Sie dafür evtl. ein Beispiel ?
Und haben Sie Erfahrungen mit dem österreichischen CAMT-Format ? Könnte ich Sie dazu befragen ?

Viele Grüße,
sap_claudi

Beitrag von „Claus Wild“ vom 2. März 2016, 13:15

Hallo sap_claudi,

ich baue und verzeifle gerade an österreichischen camt-Formaten :-(

Der camt.053 sollte dabei (hoffentlich) nicht das große Problem sein (wenn sie nach CGI geliefert werden).

Der camt.054 ist etwas frickeliger :-(

Viele Grüße

Claus Wild

Beitrag von „SAP_Consultant“ vom 2. August 2016, 12:10

Hallo Herr Wild,

ich verzweifle auch gerade am österreichischem Format der RVS in Salzburg

So sieht die Originaldatei aus, der GVC ist unter BkTxCd/Prtry (siehe Word)



So sieht die neu angelegte Transformation schreibt nicht den GVC in die Zeile



Können Sie mir weiterhelfen?

Vielen Dank - witzigerweise ist die Zahlung an Würth :-)

Viele Grüße,
Birgit

Beitrag von „Claus Wild“ vom 2. August 2016, 12:22

Hallo Birgit,

prüfen Sie bitte nochmals Ihre Transformation für den VGEXT - Sie lesen wahrscheinlich vom Wurzelement <BkTxCd> -> <Domn>
aus. Der Value-of-select muss aber <BkTxCd> -> <Prtry><Cd> lauten.

Ohne die XSLT-Transformation zu kennen ;-)

Viele Grüße

Claus Wild

Beitrag von „SAP_Consultant“ vom 2. August 2016, 13:20

Hallo Herr Wild,

bei mir sieht die Transformation so aus:

```
<!--Transaction Code-->
<VGEXT>
<xsl:choose>
<xsl:when test="ns0:BkTxCd/ns0:Domn">
<xsl:value-of                                select="concat(ns0:BkTxCd/ns0:Domn/ns0:Cd,
ns0:BkTxCd/ns0:Domn/ns0:Fmly/ns0:Cd,
ns0:BkTxCd/ns0:Domn/ns0:Fmly/ns0:SubFmlyCd)"></xsl:value-of>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="ns0:BkTxCd/ns0:Prtry/ns0:Cd"></xsl:value-of>
</xsl:otherwise>
```



```
</xsl:choose>
</VGEXT>
<VGMAN></VGMAN>
<KFMOD />
<BUTXT />
<ANZSP />
<XBLNR />
<ZUONR />
<!--Parnter Details-->
```

Was muss ich ändern? (bin kein Programmierer)

Vielen Dank!

Viele Grüße,
Birgit

Beitrag von „Claus Wild“ vom 2. August 2016, 14:50

Hallo Birgit,

okay, in dem Fall trifft die erste Bedingung zu, da in dem XML-Tag ein Wert enthalten ist:

```
<xsl:value-of                                select="concat(ns0:BkTxCd/ns0:Domn/ns0:Cd,
ns0:BkTxCd/ns0:Domn/ns0:Fmly/ns0:Cd,
ns0:BkTxCd/ns0:Domn/ns0:Fmly/ns0:SubFmlyCd)"></xsl:value-of>
```

Was dann auch das Ergebnis in der Validierung ist. Ich würde für die Bank eine eigene Transformation anlegen, die nur die zweite Bedingung abfragt. Also in etwa so:

```

<!--Transaction Code-->
<!--We didn't use this part of transformation due to existing reason code in XML-Tag Prtry.-->
<VGEXT>
<!--We only use this XML-Tag due to the right information in camt-053-->
<xsl:value-of select="ns0:NtryDtIs/ns0:TxDtIs/ns0:BkTxCd/ns0:Prtry/ns0:Cd"/>
<!--<xsl:choose>
<xsl:when test="ns0:BkTxCd/ns0:Domn">
<xsl:value-of
                                select="concat(ns0:TxDtIs/BkTxCd/ns0:Domn/ns0:Cd,
ns0:BkTxCd/ns0:Domn/ns0:Fmly/ns0:Cd, ns0:BkTxCd/ns0:Domn/ns0:Fmly/ns0:SubFmlyCd)"/>
</xsl:when>
<xsl:otherwise>
<xsl:value-of select="substring(ns0:NtryDtIs/ns0:TxDtIs/ns0:BkTxCd/ns0:Prtry/ns0:Cd,0,3)"/>
</xsl:otherwise>
</xsl:choose>-->
</VGEXT>

```

Die Abfrage ist somit immer nur für die eine Bank gültig und liefert den GVC korrekt an -
Vorausgesetzt die Bank
ändert die Bereitstellung nicht.

Viele Grüße

Claus Wild