



xsdTransformer

Generiert Code basierend auf XML Schema

Joerg Moebius

Die Beschreibung ist einsetzbar. Sie wird laufend ergänzt. Die Übersetzung ist in Arbeit. Noch nicht übersetzte Texte werden in englischer Sprache ausgegeben.

Release: 0.6 / 2006-11-22

Inhaltsverzeichnis

xsdTransformer	2
Was ist xsdTransformer?	2
xsd2crud - Generiert CRUD Applikationen	2
Zweck	2
Schnellstart	2
Leistungsumfang	3
Transformationsregeln	3
Benutzung	5
xsd2fx - generates Xforms	5
Zweck	5
Schnellstart	5
Ansatz	6
Leistungsumfang	6
Transformationsregeln	7
Benutzung	9
Beispiele	14
Bemerkungen / Erfahrungen mit Renderern	14
OPS - Orbeon Presentation Server	14
Rechtlicher Rahmen	14
Powered by	15



xsdTransformer

Generiert Code basierend auf XML Schema

Was ist xsdTransformer?

xsdTransformer besteht aus einem Set von xsl Scripten, die Datenstruktur-bezogenen Code, Descriptoren u.ä. auf Basis von XML Schemata generieren. Bis dato liegt ein Script zur Generierung von XForms Code vor (xsd2xf). Weitere Scripte folgen.

Zur Anwendung dieser Scripte können Sie einen beliebigen xsl Prozessor verwenden. Wenn Sie nicht schon einen in Ihrer Umgebung betreiben können Sie eine der drei im Folgenden dieser Beschreibung vorgestellten Lösungen einsetzen:

- Starten Sie *Xalan* [<http://xml.apache.org/xalan-j/commandline.html>] vom Betriebssystem Prompt aus.
- Verwenden Sie die *style* task vom  *ant* [<http://ant.apache.org/>] framework.
- Verwenden Sie die eclipse Benutzerschnittstelle vom xsdTransformer  *xsdTransformer for eclipse* [<http://xsdeclipse.sourceforge.net/>].

xsd2crud - Generiert CRUD Applikationen

CRUD ist die Abkürzung für Create(=Anlegen), Read(=Lesen), Update(=Modifizieren), Delete(=Löschen)

Zweck

xsd2crud generiert, basierend auf einem XML Schema, komplette CRUD Applikationen.

Schnellstart

Voraussetzungen

Da es sich bei xsd2crud.xsl um ein (Set von) xslt Skript(en) handelt, können Sie es in jeder Umgebung einsetzen, die über einen geeigneten XSL Prozessor verfügt. Wenigstens muss ein aktuelles Java installiert sein (Java 5 enthält den XSLTC Prozessor, der auf Xalan 2.6.0 plus bestimmte Fehlerbereinigungen basiert). you can use it easily at any environment containing an XSL processor. So at least a current java runtime environment has to be installed.

Installieren Sie wie folgt:

Starten der Generierung vom Prompt

führen Sie folgende Schritte aus:

- `java net.sf.saxon.Transform test1.xsd scripts/xsd2crud.xsl -o output/test1.xhtml` (Saxon) or
- `java org.apache.xalan.xslt.Process -in test1.xsd -xsl scripts/xsd2crud.xsl -out output/test1.xhtml` (Xalan)

Der Prozess läuft und endet ohne Ausgabe von Meldungen. Nachdem der Prozess beendet ist, sollten sie die Datei test1.xhtml im Ausgabeverzeichnis finden.

Für weitergehende Informationen bezüglich des Start von der command-line siehe:

entweder <http://www.saxonica.com/documentation/using-xsl/commandline.html> [http://www.saxonica.com/documentation/using-xsl/commandline.html] for using Saxon

oder <http://xml.apache.org/xalan-j/commandline.html>
commandline.html] for using Xalan

[<http://xml.apache.org/xalan-j/>

Anmerkung

Wenn Sie das Skript auf diese Weise ausführen lassen, wird die Eingabedatei auf Wohlgeformtheit, nicht aber auf Validität geprüft. Stellen Sie deshalb sicher, dass Sie nur mit validen XML Schemas arbeiten.

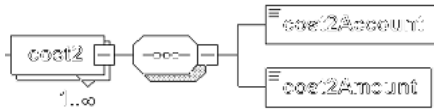
Leistungsumfang

Tabelle 1. Leistungsumfang

XForms Konstrukt	Control/Transformationsmodus	Status	Bemerkung
input/output controls			
group	automatisch	erledigt	
repeat	automatisch	erledigt	
input	automatisch	erledigt	
textarea	konfigurieren	erledigt	
secret	konfigurieren	erledigt	
output	konfigurieren	erledigt	
select1	automatisch	erledigt	
select	automatisch/ konfigurieren	erledigt	
upload	./.	nicht geplant	
range	./.	nicht geplant	
complex list	?	noch nicht entschieden	
generiere control controls			
trigger	konfigurieren	begonnen	
submit	konfigurieren	begonnen	
generiere Inhaltsattribute			
label	automatisch/ konfigurieren	erledigt	
alert	automatisch/ konfigurieren	erledigt	
hint	automatisch/ konfigurieren	erledigt	
help	automatisch/ konfigurieren	erledigt	
generiere Steuerungsattribute			
appearance	automatisch/ konfigurieren	weitgehend erledigt	
navindex	konfigurieren	geplant	
selection=open	konfigurieren	geplant	

Transformationsregeln

Das Wesentliche der Generierung



- 1 Element mit dem Attribut 'maxOccurs' >1 wird zum 'repeat' Steuerungselement.
- 2 Element mit einem 'sequence' Kindelement wird zum 'group' Steuerungselement
- 21 Element mit einem 'sequence' Kindelement und dem Attribut 'maxOccurs' >1 wird zum 'group' Steuerungselement mit einem 'repeat' Kindelement.
- 3 Element mit einem 'restriction' Kindelement wird zum 'select1' Steuerungselement.

Quelle

```

        <xs:element name="cost2" maxOccurs="unbounded">1
<xs:complexType>
  <xs:sequence>2
    <xs:element name="cost2Account" default="1002">
      <xs:simpleType>
        <xs:restriction base="xs:integer">3
          .
          .
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="cost2Amount"
      type="xs:decimal" default="1">
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

Ziel

```

<xforms:group2 id="xsd2crud.cost2GroupUI" appearance="full">
.
.

<xforms:repeat1 id="xsd2crud.cost2RepeatUI" appearance="compact" bind="B01300">
<xforms:select13 id="xsd2crud.cost2AccountUI" xforms:bind="B01400" xforms:appearance="minimal">
.
.
<xforms:input id="cost2AmountUI" xforms:bind="B01500">
</xforms:input>
</xforms:repeat>
</xforms:group>

```

Die Transformationsregeln im Detail

Tabelle 2. Die Transformationsregeln im Detail

bestimme passendes xforms ui control	1	2	3	4	5	6	7	8	9
ist ein Element	n	j	j	j	j	j	j	j	j
hat ein 'sequence' Element als Kindelement (oder Enkelement bevor ein anderes element als Kind oder Enkel auftritt)	-	n	n	n	n	n	n	j	-
minOccurs<1	-	n	n	n	n	n	?	-	
maxOccurs>1 (incl. unbounded)	-	-	-	-	-	-	?	-	j
hat ein 'restriction' Element als Kindelement (oder Enkelement bevor ein anderes element als Kind oder Enkel auftritt)	-	n	n	n	n	j	j	-	-
annotation/appinfo/appearance=textarea	-	n	j	-	-	-	-	-	-

bestimme passendes xforms ui control	1	2	3	4	5	6	7	8	9
annotation/appinfo/appearance=secret	-	n	-	j	-	-	-	-	-
annotation/appinfo/appearance=output	-	n	-	-	j	-	-	-	-
generiere group								x	
generiere repeat									x
generiere input		x							
			x						
generiere secret				x					
generiere output					x				
generiere select1						x			
generiere select							x		
generiere repeat triggers								x	
tue nichts								x	
erledigt		x	x	x	x	x		x	
todo						x	x		

Regeln 5 und 6 müssen noch überarbeitet werden.

Benutzung

Da es sich bei `xsd2xf.xml` um ein einzelnes xslt Script handelt, können Sie es einfach in jeder Umgebung einsetzen, die über einen xsl Prozessor verfügt.

Einsatz unter ant

`xsd2xrud` (Skripte und Dokumentation) sind noch in der ersten Phase der Entwicklung. Für's erste steht ihnen das ant Skript `transAddress.xml` zum Ausprobieren von `xsd2crud` zur Verfügung.

xsd2fx - generates Xforms

Zweck

`xsd2fx` generiert aus einem XML Schema eine XForms Form, die in einer XHTML site eingebettet ist.

Schnellstart

Voraussetzungen

Da `xsd2xf.xml` ein xslt script ist, können Sie es einfach in jeder Umgebung einsetzen, die einen geeigneten xsl Prozessor enthält. Dieser xsl Prozessor wiederum setzt eine aktuelle Java Laufzeitumgebung voraus.

Zur Installation führen Sie bitte folgende Schritte aus:

Kopieren Sie die heruntergeladenen Verzeichnisse und Dateien in ein `installationDirectory`

Start der Generierung vom BS Prompt

führen Sie folgende Schritte aus

- `java net.sf.saxon.Transform test1.xsd scripts/xsd2xf.xsl -o output/test1.xhtml` (Saxon) oder
- `java org.apache.xalan.xslt.Process -in test1.xsd -xsl scripts/xsd2xf.xsl -out output/test1.xhtml` (Xalan)

Der Programmablauf endet ohne irgendeine Meldung. Nach dem Programmablauf müsste eine Datei mit Namen test1.xhtml im Verzeichnis *output* zu finden sein.

Näheres zur Benutzung der Xalan Command-Line Utility finden Sie unter:

<http://www.saxonica.com/documentation/using-xsl/commandline.html> [<http://www.saxonica.com/documentation/using-xsl/commandline.html>] für die Nutzung von Saxon

oder <http://xml.apache.org/xalan-j/commandline.html> für die Nutzung von Xalan

Anmerkung

Wenn Sie die Generierung in dieser Weise betreiben, wird das Quell-XML-Schema zwar auf Wohlgeformtheit, nicht aber auf Validität geprüft. Stellen Sie deshalb sicher, dass Sie nur valide XML Schemata verwenden.

Run Generation with ant

For easy standalone usage xsd2fx comes with an ant script named xsdTransform.xml. If you want to use this script, ant must be installed. For details concerning ant see <http://ant.apache.org/>.

proceed the following steps:

- change to <installation directory>.
- run 'ant -f xsdTransform.xml xsd2xf1' on os prompt.

This process should terminate with the message 'BUILD SUCCESSFUL' and you should find two files - test1.xhtml and costVoucher.xhtml - in the output directory.

Anmerkung

Running the transformation in this manner includes a well formed check but not a valid check. So make sure to use only valid schemata. (For details how to parse the input schema see 'Usage' -> 'Start with ant script'.)

Ansatz

xsd2fx ist so einfach wie möglich aufgebaut. Sie können dieses Script ohne Änderungen auf Ihre XML Schemata anwenden und erhalten XForms Code, den Sie in Ihrer weiteren Entwicklungsarbeit einsetzen können. Sie können die Generierung allerdings auch mit der Änderung der Programmparameter nach Ihren Bedürfnissen anpassen. Näheres hierzu finden Sie im Abschnitt 'Benutzung'.

Weiters können Sie das Script selbst an Ihre Anforderungen anpassen.

Leistungsumfang

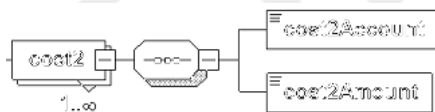
Tabelle 3. Leistungsumfang

XForms Konstrukt	Control/	Transformations-modus	Status	Bemerkung
input/output controls				
group		automatisch	erledigt	
repeat		automatisch	erledigt	
input		automatisch	erledigt	
textarea		konfigurieren	erledigt	
secret		konfigurieren	erledigt	

XForms Konstrukt	Control/Transformationsmodus	Status	Bemerkung
output	konfigurieren	erledigt	
select1	automatisch	erledigt	
select	automatisch/ konfigurieren	erledigt	
upload	./.	nicht geplant	
range	./.	nicht geplant	
complex list	?	noch nicht entschieden	
generiere control controls			
trigger	konfigurieren	begonnen	
submit	konfigurieren	begonnen	
generiere Inhaltsattribute			
label	automatisch/ konfigurieren	erledigt	
alert	automatisch/ konfigurieren	erledigt	
hint	automatisch/ konfigurieren	erledigt	
help	automatisch/ konfigurieren	erledigt	
generiere Steuerungsattribute			
appearance	automatisch/ konfigurieren	weitgehend erledigt	
navindex	konfigurieren	geplant	
selection=open	konfigurieren	geplant	

Transformationsregeln

Das Wesentliche der Generierung



- 1 Element mit dem Attribut 'maxOccurs' >1 wird zum 'repeat' Steuerungselement.
- 2 Element mit einem 'sequence' Kindelement wird zum 'group' Steuerungselement
- 21 Element mit einem 'sequence' Kindelement und dem Attribut 'maxOccurs' >1 wird zum 'group' Steuerungselement mit einem 'repeat' Kindelement.
- 3 Element mit einem 'restriction' Kindelement wird zum 'select1' Steuerungselement.

Quelle

```
<xs:element name="cost2" maxOccurs="unbounded">1
  <xs:complexType>
    <xs:sequence>2
      <xs:element name="cost2Account" default="1002">
        <xs:simpleType>
          <xs:restriction base="xs:integer">3
            .
            .
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
```

```

    <xs:element name="cost2Amount"
      type="xs:decimal" default="1">
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

```

Ziel

```

<xforms:group2 id="cost2GroupUI" appearance="full">
.
.
  <xforms:repeat1 id="cost2RepeatUI" appearance="compact" bind="B01300">
    <xforms:select1 id="cost2AccountUI" xforms:bind="B01400" xforms:appearance="minimal">
      .
      <xforms:input id="cost2AmountUI" xforms:bind="B01500">
    </xforms:input>
  </xforms:repeat>
</xforms:group>

```

Die Transformationsregeln im Detail

Tabelle 4. Die Transformationsregeln im Detail

bestimme passendes xforms ui control	1	2	3	4	5	6	7	8	9
ist ein Element	n	j	j	j	j	j	j	j	j
hat ein 'sequence' Element als Kindelement (oder Enkelelement bevor ein anderes element als Kind oder Enkel auftritt)	-	n	n	n	n	n	n	j	-
minOccurs<1	-	n	n	n	n	n	?	-	
maxOccurs>1 (incl. unbounded)	-	-	-	-	-	-	?	-	j
hat ein 'restriction' Element als Kindelement (oder Enkelelement bevor ein anderes element als Kind oder Enkel auftritt)	-	n	n	n	n	j	j	-	-
annotation/appinfo/appearance=textarea	-	n	j	-	-	-	-	-	-
annotation/appinfo/appearance=secret	-	n	-	j	-	-	-	-	-
annotation/appinfo/appearance=output	-	n	-	-	j	-	-	-	-
generiere group								x	
generiere repeat									x
generiere input		x							
generiere secret			x						
generiere output				x					
generiere select1						x			
generiere select							x		
generiere repeat triggers								x	
tue nichts								x	
erledigt		x	x	x	x	x		x	
todo						x	x		

Regeln 5 und 6 müssen noch überarbeitet werden.

Benutzung

Da es sich bei `xsd2xf.xml` um ein einzelnes xslt Script handelt, können Sie es einfach in jeder Umgebung einsetzen, die über einen xsl Prozessor verfügt.

Einsatz unter ant

Für den standalone Betrieb steht das ant script `xsdtrans.xml` zur Verfügung. Das Script enthält drei *targets*, von denen die ersten beiden Transformationsprozesse starten. Das dritte *target* können Sie bei Bedarf zum parsen Ihres XML Schemas verwenden.

In `xsdtrans.xml` können Sie sich außerdem über die Verwendung der Programmparameter informieren, die im Folgenden detailliert erklärt werden.

Tabelle 5. Benutzung

Parameter	Beschreibung	Pflicht
in	Speicherort der XML-Schema-Datei, die transformiert werden soll. (Ist Parameter sowohl für ant's style als auch für das Script. Für das Script hat er die Funktion, Meldungen zu erzeugen und kann weggelassen werden.) Wertebereich: gültiger Speicherort relativ zum aktuellen Verzeichnis (Pfad/Dateiname) oder eine vollständiger Speicherort (/Pfad/Dateiname) Vorgabewert: keiner	yes
out	Speicherort der Ausgabedatei (hier eine xhtml Datei, die XForms enthält). (Ist Parameter sowohl für ant's style als auch für das Script. Für das Script hat er die Funktion, Meldungen zu erzeugen und kann weggelassen werden.) Wertebereich: gültiger Speicherort relativ zum aktuellen Verzeichnis (Pfad/Dateiname) oder eine vollständiger Speicherort (/Pfad/Dateiname) Vorgabewert: keiner	yes
style	location of name of the transformer script (including path relative to project directory). (This parameter is used as input for ant's style and as input for the script. The script related parameter is only used for generating messages and can be omitted.) valid values: existent file location including path relative to current directory (path/filename) or absolute location (/path/filename) default: none	yes
force	Recreate target files, even if they are newer than their corresponding source files or the stylesheet. [ant manual]	no

	<p>valid values: yes/no</p> <p>default: no</p>	
inElement	<p>element which is used as root element. The root element is starting point of the transformation.</p> <p>valid values: any string</p> <p>default: none</p>	no
outForm	<p>desired name of the xforms form which is to generate.</p> <p>valid values: any string</p> <p>default: form1</p>	no
language	<p>language in which labels and messages will be transferred to the new form.</p> <p>valid values: en (for english, de (for german)</p> <p>default: en</p>	no
bindPrefix	<p>The id of bind elements is composed of a prefix and a current number. Sample:</p> <pre><xforms:bind id="B00300" xforms:nodeset="vchr:title" xforms:required="true()" xforms:type="string"/></pre> <p>With this parameter you can determine the value of this prefix.</p> <p>valid values: any string</p> <p>default: B</p>	no
minRestrictionsForFull	<p>This parameter determines the appearance of select1 and select controls. Is the amount of restrictions concerning a certain select* control is less than minRestrictionsForFull, the appearance of this control is set to 'minimal' otherwise it is set to 'full'.</p> <p>valid values: any integer</p> <p>default value: 4</p>	
generateAbsentAlert	<p>If this parameter is set to 'yes' and no alert value is determined within the appinfo's (see configuration) the value of the alert element is derived by default from 'minOccurs' and 'type' attributes of the source element. Sample:</p> <pre><xs:element name="title" type="xs:string" default="titleDefaultValue"></pre>	no

	<p>(when 'minOccurs' is omitted it means 'minOccurs=1', so input is required)</p> <p>becomes</p> <pre><xforms:input id="title" xforms:bind="B00300"> <xforms:label>title</xforms:label> <xforms:alert>Input is obligative. Only input of type 'string' allowed.</xforms:alert> <xforms:hint>hint for title</xforms:hint> <xforms:help>help for title</xforms:help> </xforms:input></pre> <p>valid values: yes/no</p> <p>default value: yes</p>	
<p>generateAbsentHint</p>	<p>If this parameter is set to 'yes' and no hint value is determined within the appinfo's (see configuration) a hint element with a place holder value will be created. This value must be manually changed later. Sample:</p> <pre><xs:element name="title"...></pre> <p>becomes</p> <pre><xforms:input id="title" xforms:bind="B00300"> . <xforms:hint>hint for title</xforms:hint> . </xforms:input></pre> <p>To avoid unnecessary manual work this parameter is set by default to 'no'.</p> <p>valid values: yes/no</p>	<p>no</p>

	default value: no	
generateAbsentHelp	<p>If this parameter is set to 'yes' and no help value is determined within the appinfo's (see configuration) a help element with a place holder value will be created. This value must be manually changed later. Sample:</p> <pre><xs:element name="title" ...></pre> <p>becomes</p> <pre><xforms:input id="title" xforms:bind="B00300"></pre> <p>.</p> <pre><xforms:help>help for title</xforms:help></pre> <p>.</p> <pre></xforms:input></pre> <p>To avoid unnecessary manual work this parameter is set by default to 'no'.</p> <p>valid values: yes/no</p> <p>default value: no</p>	no

Mit ant parsen

With the latter target within xsdtrans.xml script you can simply parse your schema.

Ant uses for xml parsing an optional task called 'XMLValidate'. For the prerequisites using this task please refer to <http://ant.apache.org/manual/install.html>, here the chapter concerning 'Optional Tasks'.

To parse your input schema simply type in at os prompt:

```
'ant -f xsdTransform.xml xsdparse'
```

Run within eclipse Einsatz in eclipse

The eclipse userinterface of xsdTransformer is realized as separate project named xsdeclipse. you can download xsdeclipse plugin either via

xsdeclipse or

xsdtrans

For information concerning the usage within eclipse please refer to the xsdeclipse documentation.

Anpassen mit *annotation/appinfo* Elementen

In einem XML schema können Sie Datenschema neutrale Information in einem Element namens *annotation*. *annotation* kann zwei Elemente enthalten. Das erste heißt *documentation* und

das zweite *appinfo*. Wie die Namen bereits sind diese Elemente für die Dokumentation und die Applikationssteuerung vorgesehen.

Sie können die Transformation beeinflussen, indem Sie bestimmte Elemente im Element *annotation/appinfo* platzierten. Im Einzelnen:

Tabelle 6. Konfigurationin mit zusätzlichen *annotation/appinfo* Elementen

Element	Beschreibung	Einsatz in/für Elementen
label	xsd2xf fügt basierend auf dem Wert dieses Elements dem control Element ein Kindelement namens <i>label</i> zu. Ist dieses Element nicht vorhanden aber der Prozessparamter <i>generate absent labels</i> gesetzt, fügt xsd2xf das <i>label</i> Kindelement ebenfalls hinzu und setzt dann als Wert den Namen des control Elements ein. Da es sich bei <i>label</i> um einen sprachsensitiven Ausdruck handelt, wird für jede Sprache ein Kindelement erzeugt.	alle
alert	xsd2xf fügt basierend auf dem Wert dieses Elements dem control Element ein Kindelement namens <i>alert</i> zu. Ist dieses Element nicht vorhanden aber der Prozessparamter <i>generate absent alerts</i> gesetzt, fügt xsd2xf das <i>alert</i> Kindelement ebenfalls hinzu und setzt seinen Wert dann auf eine geeignete Meldung, die von den Feldspezifikationen (Feldtyp, Mindestanzahl in einer Instanz) abgeleitet ist. Da es sich bei <i>label</i> um einen sprachsensitiven Ausdruck handelt, wird für jede Sprache ein Kindelement erzeugt.	alle editierbaren/ änderbaren
hint	xsd2xf fügt basierend auf dem Wert dieses Elements dem control Element ein Kindelement namens <i>hint</i> zu. Ist dieses Element nicht vorhanden aber der Prozessparamter <i>generate absent hints</i> gesetzt, fügt xsd2xf das <i>hint</i> Kindelement ebenfalls als Platzhatler für die weitere Bearbeitung hinzu und setzt seinen Wert dann auf einen Platzhaltertext.	alle
help	xsd2xf fügt basierend auf dem Wert dieses Elements dem control Element ein Kindelement namens <i>help</i> zu. Ist dieses Element nicht vorhanden aber der Prozessparamter <i>generate absent helps</i> gesetzt, fügt xsd2xf das <i>help</i> Kindelement ebenfalls als Platzhatler für die weitere Bearbeitung hinzu und setzt seinen Wert dann auf einen Platzhaltertext.	alle
appearance	Normale Elemente werden standardmäßig zu <i>input</i> control Elementen in XForms. Um diesen Standard zu beeinflussen, können Sie dieses Element folgende Werte verwenden: <ul style="list-style-type: none"> • textarea • secret • output Weiters gibt es Elemente, die <i>selectx</i> control Elementen transformiert werden. Für diese Fälle	<ul style="list-style-type: none"> • Elemente, die zu <i>input</i> control Elementen werden. • Elemente, die zu <i>selectx</i> control Elementen werden.

Element	Beschreibung	Einsatz in/für Elementen
	können Sie die Standardwert für das <i>appearance</i> attribute ändern, indem Sie dieses Element auf den von Ihnen gewünschten Wert setzen (<i>minimal</i> oder <i>full</i>).	

If you are working with these supplementary elements and parse you parse your schema or a toll do, you might get error messages concerning these elements. You can avoid the appearance of such messages placinig an internal DTD subset into your XML Schema: Wenn Sie mit diesen zusätzlichen Elementen arbeiten und Sie (oder ein Werkzeug) parsen Ihr Schema, ist es möglich, dass Fehlermeldungen betreffend der Zusatzelemente ausgegeben werden. Das können Sie vermeiden, indem Sie ein DTD subset in Ihr Schema einfügen:

```
<?xml version="1.0"?>
<!DOCTYPE xs:schema PUBLIC "-//W3C//DTD XMLSchema 2001//EN"
"http://www.w3.org/2001/XMLSchema.dtd" [
  <!ELEMENT label (#PCDATA)>
  <!ELEMENT alert (#PCDATA)>
  <!ELEMENT hint (#PCDATA)>
  <!ELEMENT help (#PCDATA)>
  <!ATTLIST label xml:lang (en | de) "en">
  <!ATTLIST alert xml:lang (en | de) "en">
  <!ATTLIST hint xml:lang (en | de) "en">
  <!ATTLIST help xml:lang (en | de) "en">
]
>
<xs:schema targetNamespace="http://xsdtrans.sourceforge.net/costVoucher"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="unqualified"
version="1.0">
  <xs:element name="voucher">
    .
    .
  </xs:element>
</xs:schema>
```

Beispiele

todo

Bemerkungen / Erfahrungen mit Renderern

todo

OPS - Orbeon Presentation Server

Besonderheiten

- binds dürfen nicht hierarchisch aufgebaut sein, weil die Regelattribute sonst keinen Effekt haben.
- Zwei (und wahrscheinlich auch mehr) input Elemente mit einem label Element in einem group Element, dass ebenfalls ein label Element hat, führen zu der Fehlermeldung:

'A sequence of more than one item is not allowed as the value of variable \$label'.

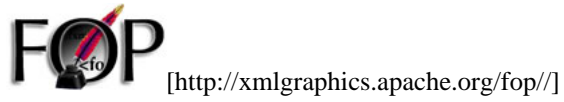
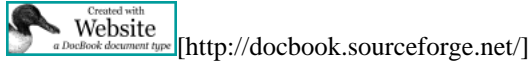
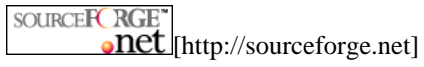
Wenn die input elemente jeweils in einem <p> Element plziert werden, wird diese Fehlermeldung vermieden.

Rechtlicher Rahmen

xsdTranformer wird unter den Bedingungen der GNU LGPL Lizenz (see <http://www.gnu.org/copyleft/lesser.html>) unter Ausschluss jeglicher Garantien und Gewährleistungen veröffentlicht.

Copyright © 2006 joerg.moebius@hamburg.de [mailto:joerg.moebius@hamburg.de]

Powered by



DRAFT