

XML Workshop XIII - XSD And Variable Content Containers

By [Jacob Sebastian](#), 2007/11/01

Introduction

In the past few sessions of *XML Workshop*, we had a good discussion about schema validation. We saw that, by defining a *SCHEMA*, we could strictly validate the data being assigned to an XML column or variable. When a schema is bound to a column or variable, the *XML* value should *EXACTLY* match the schema definition. This is what we need almost always.

There are times when we need to allow data with different *XML* structures. Let us take an example. Assume that we are working on a billing application and we need to store the payment details. Our application supports the following payment methods: *Check*, *Cash* and *Credit Card*. Based on the payment method, we will have to store additional information into the tables. If it is *Check*, then we need to store the *Bank Name*, *Branch* and *Check Number*. If the payment method is *Cash* then we need to store the *Currency* and the *denomination*. If the payment is made by *Credit Card*, then we need to store *Card Type*, *Number*, *Expiry date* etc.

The *XML* that our application should support are the following.

Cash Payment

```

1 <PaymentDetails>
2   <Type>Cash</Type>
3   <CashDetails>
4     <Currency>USD</Currency>
5     <Denomination>
6       <Hundreds>5</Hundreds>
7       <Tens>3</Tens>
8       <Ones>2</Ones>
9     </Denomination>
10  </CashDetails>
11 </PaymentDetails>

```

Payment By Check

```

1 <PaymentDetails>
2   <Type>Check</Type>
3   <CheckDetails>
4     <BankName>HSBC</BankName>
5     <Branch>NY City</Branch>
6     <CheckNumber>445908</CheckNumber>
7   </CheckDetails>
8 </PaymentDetails>

```

Payment By Credit Card

```

1 <PaymentDetails>
2   <Type>CreditCard</Type>
3   <CreditCardDetails>
4     <CardType>Visa</CardType>
5     <CardNumber>xxxx xxxx xxxx 9090</CardNumber>
6     <VerificationCode>896</VerificationCode>
7     <ExpiryDate>0911</ExpiryDate>
8   </CreditCardDetails>
9 </PaymentDetails>

```

How do we define a schema which supports all these different *XML* structures? Well, there are many ways to do it. The easiest way is to use the *xsd:choice* sequence indicator.

Step 1 - Define the Payment Method

Let us start defining the schema. The first step is to create the *Payment Method* enumeration.

```

1 <xsd:simpleType name="PaymentMethod">
2   <xsd:restriction base="xsd:string">
3     <xsd:enumeration value="Cash"/>
4     <xsd:enumeration value="Check" />
5     <xsd:enumeration value="CreditCard" />
6   </xsd:restriction>
7 </xsd:simpleType>

```

Step 2 - Define Cash Details

Now let us define the schema for the cash payment details.

```

1 <xsd:complexType name="CashPayment">
2   <xsd:all>
3     <xsd:element name="Currency" type="xsd:string"/>
4     <xsd:element name="Denomination">
5       <xsd:complexType>
6         <xsd:all>
7           <xsd:element name="Hundreds" type="xsd:integer"/>
8           <xsd:element name="Tens" type="xsd:integer" />
9           <xsd:element name="Ones" type="xsd:integer" />
10        </xsd:all>
11      </xsd:complexType>
12    </xsd:element>
13  </xsd:all>
14 </xsd:complexType>

```

Step 3 - Define Check Details

Here is the schema for the *Check Details*

```

1 <xsd:complexType name="CheckPayment">
2   <xsd:all>
3     <xsd:element name="BankName" type="xsd:string" />
4     <xsd:element name="Branch" type="xsd:string" />
5     <xsd:element name="CheckNumber" type="xsd:string" />
6   </xsd:all>
7 </xsd:complexType>

```

Step 4 - Define Credit Card Details

And finally, let us define the schema for *Credit Card Details*

```

1 <xsd:complexType name="CreditCardPayment">
2   <xsd:all>
3     <xsd:element name="CardType" type="xsd:string"/>
4     <xsd:element name="CardNumber" type="xsd:string" />
5     <xsd:element name="VerificationCode" type="xsd:string"/>
6     <xsd:element name="ExpiryDate" type="xsd:string" />
7   </xsd:all>
8 </xsd:complexType>

```

Step 5 - The final schema

Here is the complete schema. [[code](#)]

```

1 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
2   <xsd:element name="PaymentDetails">
3     <xsd:complexType>
4       <xsd:sequence>
5         <xsd:element name="Type" type="PaymentMethod" />
6         <xsd:choice>
7           <xsd:element name="CashDetails" type="CashPayment" />
8           <xsd:element name="CheckDetails" type="CheckPayment" />
9           <xsd:element name="CreditCardDetails" type="CreditCardPayment" />
10        </xsd:choice>
11      </xsd:sequence>
12    </xsd:complexType>
13  </xsd:element>
14  <xsd:simpleType name="PaymentMethod">
15    <xsd:restriction base="xsd:string">
16      <xsd:enumeration value="Cash"/>
17      <xsd:enumeration value="Check" />
18      <xsd:enumeration value="CreditCard" />
19    </xsd:restriction>
20  </xsd:simpleType>
21  <xsd:complexType name="CashPayment">
22    <xsd:all>
23      <xsd:element name="Currency" type="xsd:string"/>
24      <xsd:element name="Denomination">
25        <xsd:complexType>
26          <xsd:all>
27            <xsd:element name="Hundreds" type="xsd:integer"/>
28            <xsd:element name="Tens" type="xsd:integer" />
29            <xsd:element name="Ones" type="xsd:integer" />
30          </xsd:all>
31        </xsd:complexType>
32      </xsd:element>
33    </xsd:all>
34  </xsd:complexType>
35  <xsd:complexType name="CreditCardPayment">
36    <xsd:all>
37      <xsd:element name="CardType" type="xsd:string"/>
38      <xsd:element name="CardNumber" type="xsd:string" />
39      <xsd:element name="VerificationCode" type="xsd:string"/>
40      <xsd:element name="ExpiryDate" type="xsd:string" />
41    </xsd:all>
42  </xsd:complexType>
43  <xsd:complexType name="CheckPayment">
44    <xsd:all>
45      <xsd:element name="BankName" type="xsd:string" />
46      <xsd:element name="Branch" type="xsd:string" />
47      <xsd:element name="CheckNumber" type="xsd:string" />
48    </xsd:all>
49  </xsd:complexType>
50 </xsd:schema>

```

Creating the Schema Collection

Let us create the schema collection now. [\[code\]](#)

```

1 CREATE XML SCHEMA COLLECTION PaymentType
2 AS
3 ' <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
4   <xsd:element name="PaymentDetails">
5     <xsd:complexType>
6       <xsd:sequence>
7         <xsd:element name="Type" type="PaymentMethod" />
8         <xsd:choice>
9           <xsd:element name="CashDetails" type="CashPayment" />
10          <xsd:element name="CheckDetails" type="CheckPayment" />
11          <xsd:element name="CreditCardDetails" type="CreditCardPayment" />
12        </xsd:choice>
13      </xsd:sequence>
14    </xsd:complexType>
15  </xsd:element>
16  <xsd:simpleType name="PaymentMethod">

```

```

17     <xsd:restriction base ="xsd:string">
18         <xsd:enumeration value="Cash"/>
19         <xsd:enumeration value="Check" />
20         <xsd:enumeration value="CreditCard" />
21     </xsd:restriction>
22 </xsd:simpleType>
23 <xsd:complexType name="CashPayment">
24     <xsd:all>
25         <xsd:element name="Currency" type="xsd:string"/>
26         <xsd:element name="Denomination">
27             <xsd:complexType>
28                 <xsd:all>
29                     <xsd:element name="Hundreds" type="xsd:integer"/>
30                     <xsd:element name="Tens" type="xsd:integer" />
31                     <xsd:element name="Ones" type="xsd:integer" />
32                 </xsd:all>
33             </xsd:complexType>
34         </xsd:element>
35     </xsd:all>
36 </xsd:complexType>
37 <xsd:complexType name="CreditCardPayment">
38     <xsd:all>
39         <xsd:element name="CardType" type="xsd:string"/>
40         <xsd:element name="CardNumber" type="xsd:string" />
41         <xsd:element name="VerificationCode" type="xsd:string"/>
42         <xsd:element name="ExpiryDate" type="xsd:string" />
43     </xsd:all>
44 </xsd:complexType>
45 <xsd:complexType name="CheckPayment">
46     <xsd:all>
47         <xsd:element name="BankName" type="xsd:string" />
48         <xsd:element name="Branch" type="xsd:string" />
49         <xsd:element name="CheckNumber" type="xsd:string" />
50     </xsd:all>
51 </xsd:complexType>
52 </xsd:schema>'

```

Testing the schema

We have created the schema that we needed. Now it is time to test it. Let us try to assign the different *XML* data that we discussed earlier and see if *SQL Server 2005* validates them correctly. [\[1\]](#), [\[2\]](#), [\[3\]](#)

```

1 DECLARE @x XML(PaymentType)
2 SET @x = '
3 <PaymentDetails>
4   <Type>CreditCard</Type>
5   <CreditCardDetails>
6     <CardType>Visa</CardType>
7     <CardNumber>xxxx xxxx xxxx 9090</CardNumber>
8     <VerificationCode>896</VerificationCode>
9     <ExpiryDate>0911</ExpiryDate>
10  </CreditCardDetails>
11 </PaymentDetails>
12 '

```

```

1 DECLARE @x XML(PaymentType)
2 SET @x = '
3 <PaymentDetails>
4   <Type>Check</Type>
5   <CheckDetails>
6     <BankName>HSBC</BankName>
7     <Branch>NY City</Branch>
8     <CheckNumber>445908</CheckNumber>
9   </CheckDetails>
10 </PaymentDetails>
11 '

```

```

1 DECLARE @x XML(PaymentType)
2 SET @x = '
3 <PaymentDetails>

```

```
4    <Type>Cash</Type>
5    <CashDetails>
6      <Currency>USD</Currency>
7      <Denomination>
8        <Hundreds>5</Hundreds>
9        <Tens>3</Tens>
10       <Ones>2</Ones>
11     </Denomination>
12   </CashDetails>
13 </PaymentDetails>
14 '
```

Conclusions

This session of *XML Workshop* focussed on explaining how to create variable content containers. The schema that we defined in this session, supports 3 different *XML* structures. Based on the payment type, a different *XML* structure can be assigned to the *XML* variable or column bound to the schema that we defined. You can find the other articles in this series [here](#).

Copyright © 2002-2007 Simple Talk Publishing. All Rights Reserved. [Privacy Policy](#). [Terms of Use](#)