http://www.sqlservercentral.com/articles/Miscellaneous/2996/
Printed 2007/11/07 02:20PM

More Advanced XML Processing Examples

By Jacob Sebastian, 2007/05/15

Introduction

In Part IV of my *Sales Order Workshop*, I had presented a basic example using the XML data type introduced by SQL Server 2005. Recently there were some questions and requests in the discussion forum asking for more detailed examples showing XML processing.

I also had another article published with some advanced XML examples and I am trying to present a few more examples in this article.

One of the most useful methods exposed by the XML data type is the *Value()* method. Here, I am presenting 9 more examples which demonstrates the different XML operations that we could perform with the *Value()* method.

Examples

Example 1

```
/*
    The following TSQL retrieves attribute values from the XML variable.
    Attribute names are prefixed with "@".
*/

DECLARE @x XML
SET @x = '
<orderInfo>
    <item code="A001" category="FOOD" subcategory="Candies">
        <description>Nestle Munch</description>
        <qty>10</qty>
        <rate>11.25</rate>
    </item>
    <item code="A002" category="FOOD" subcategory="Biscuits">
        <description>Britania Good Day</description>
        <qty>15</qty>
        <rate>12.25</rate>
    </item>
</orderInfo>'

SELECT
    x.item.value('@code[1]', 'VARCHAR(20)') AS ItemCode,
    x.item.value('@category[1]', 'VARCHAR(20)') AS category,
    x.item.value('@subcategory[1]', 'VARCHAR(20)') AS subcategory
FROM @x.nodes('//orderInfo/item') AS x(item)

/*
OUTPUT:

ItemCode             category             subcategory
-------------------- -------------------- --------------------
A001                 FOOD                 Candies
A002                 FOOD                 Biscuits
```

```
34
35 (2 row(s) affected)
36 */
```

Example 2

```
 1 /*
 2     The following TSQL retrives values from XML nodes.
 3     Note that, here we dont use the "@" sign to indicate that
 4     we need values of nodes not attributes.
 5 */
 6
 7 DECLARE @x XML
 8 SET @x = '
 9 <orderInfo>
10     <item code="A001" category="FOOD" subcategory="Candies">
11         <description>Nestle Munch</description>
12         <qty>10</qty>
13         <rate>11.25</rate>
14     </item>
15     <item code="A002" category="FOOD" subcategory="Biscuits">
16         <description>Britania Good Day</description>
17         <qty>15</qty>
18         <rate>12.25</rate>
19     </item>
20 </orderInfo>'
21
22 SELECT
23     x.item.value('description[1]', 'VARCHAR(20)') AS description,
24     x.item.value('qty[1]', 'INT') AS qty,
25     x.item.value('rate[1]', 'FLOAT') AS rate
26 FROM @x.nodes('//orderInfo/item') as x(item)
27
28 /*
29 OUTPUT:
30
31 description          qty         rate
32 -------------------- ----------- ----------------------
33 Nestle Munch         10          11.25
34 Britania Good Day    15          12.25
35
36 (2 row(s) affected)
37 */
```

Example 3

```
 1 /*
 2     Well, this query retrieves attribute values as well as values
 3     from nodes. Note that attribute values are specified with an "@"
 4     character.
 5 */
 6 DECLARE @x XML
 7 SET @x = '
 8 <orderInfo>
 9     <item code="A001" category="FOOD" subcategory="Candies">
10         <description>Nestle Munch</description>
11         <qty>10</qty>
12         <rate>11.25</rate>
13     </item>
14     <item code="A002" category="FOOD" subcategory="Biscuits">
```

```
        <description>Britania Good Day</description>
        <qty>15</qty>
        <rate>12.25</rate>
    </item>
</orderInfo>'

SELECT
    x.item.value('@code[1]', 'VARCHAR(20)') AS ItemCode,
    x.item.value('@category[1]', 'VARCHAR(20)') AS category,
    x.item.value('@subcategory[1]', 'VARCHAR(20)') AS subcategory,
    x.item.value('description[1]', 'VARCHAR(20)') AS description,
    x.item.value('qty[1]', 'INT') AS qty,
    x.item.value('rate[1]', 'FLOAT') AS rate
FROM @x.nodes('//orderInfo/item') AS x(item)

/*
OUTPUT:

ItemCode             category             subcategory         description
  qty         rate
-------------------- -------------------- -------------------- ------------
-------- ----------- ----------------------
A001                 FOOD                 Candies             Nestle Munch
  10         11.25
A002                 FOOD                 Biscuits             Britania
Good Day    15         12.25

(2 row(s) affected)
*/
```

Example 4

```
/*
    The following example demonstrates how to extract the value
    from a given row. This example extracts a value from the first
    row.

    The first example selects the value from the first row.
    The second example adds an alias to the result column.
    The third example assigns the result to a variable.
*/

DECLARE @x XML
SET @x = '
<orderInfo>
    <item code="A001" category="FOOD" subcategory="Candies">
        <description>Nestle Munch</description>
        <qty>10</qty>
        <rate>11.25</rate>
    </item>
    <item code="A002" category="FOOD" subcategory="Biscuits">
        <description>Britania Good Day</description>
        <qty>15</qty>
        <rate>12.25</rate>
    </item>
</orderInfo>'

SELECT @x.value('(/orderInfo/item/@code)[1]', 'VARCHAR(20)')

SELECT @x.value('(/orderInfo/item/@code)[1]', 'VARCHAR(20)') AS Code

DECLARE @code VARCHAR(20)
SELECT @code = @x.value('(/orderInfo/item/@code)[1]', 'VARCHAR(20)')
SELECT @code as Code
```

```
33
34 /*
35 OUTPUT:
36
37 --------------------
38 A001
39
40 (1 row(s) affected)
41
42 Code
43 --------------------
44 A001
45
46 (1 row(s) affected)
47
48 Code
49 --------------------
50 A001
51
52 (1 row(s) affected)
53 */
54
```

Example 5

```
1 /*
2     The following example retrieves the value from the second row.
3 */
4
5 DECLARE @x XML
6 SET @x = '
7 <orderInfo>
8     <item code="A001" category="FOOD" subcategory="Candies">
9         <description>Nestle Munch</description>
10        <qty>10</qty>
11        <rate>11.25</rate>
12    </item>
13    <item code="A002" category="FOOD" subcategory="Biscuits">
14        <description>Britania Good Day</description>
15        <qty>15</qty>
16        <rate>12.25</rate>
17    </item>
18 </orderInfo>'
19
20 SELECT @x.value('(/orderInfo/item/@code)[2]', 'VARCHAR(20)')
21
22 /*
23 OUTPUT:
24
25 --------------------
26 A002
27
28 (1 row(s) affected)
29 */
30
```

Example 6

```
/*
    The following example retrieves the value of an
    element from the first row.
*/
DECLARE @x XML
SET @x = '
<orderInfo>
    <item code="A001" category="FOOD" subcategory="Candies">
        <description>Nestle Munch</description>
        <qty>10</qty>
        <rate>11.25</rate>
    </item>
    <item code="A002" category="FOOD" subcategory="Biscuits">
        <description>Britania Good Day</description>
        <qty>15</qty>
        <rate>12.25</rate>
    </item>
</orderInfo>'

SELECT @x.value('(/orderInfo/item/description)[1]', 'VARCHAR(20)')
/*
OUTPUT:

--------------------
Nestle Munch

(1 row(s) affected)
*/

```

Example 7

```
/*
    The following example retrieves the value of an
    element from the second row.
*/
DECLARE @x XML
SET @x = '
<orderInfo>
    <item code="A001" category="FOOD" subcategory="Candies">
        <description>Nestle Munch</description>
        <qty>10</qty>
        <rate>11.25</rate>
    </item>
    <item code="A002" category="FOOD" subcategory="Biscuits">
        <description>Britania Good Day</description>
        <qty>15</qty>
        <rate>12.25</rate>
    </item>
</orderInfo>'

SELECT @x.value('(/orderInfo/item/description)[2]', 'VARCHAR(20)')
/*
OUTPUT:

--------------------
Britania Good Day

(1 row(s) affected)
*/

```

Example 8

```
/*
    Now let us have a look at filtering results. The following
    example applies a filter on an attribute value.
*/
DECLARE @x XML
SET @x = '
<orderInfo>
    <item code="A001" category="FOOD" subcategory="Candies">
        <description>Nestle Munch</description>
        <qty>10</qty>
        <rate>11.25</rate>
    </item>
    <item code="A002" category="FOOD" subcategory="Biscuits">
        <description>Britania Good Day</description>
        <qty>15</qty>
        <rate>12.25</rate>
    </item>
</orderInfo>'

SELECT
    x.item.value('@code[1]', 'VARCHAR(20)') AS ItemCode,
    x.item.value('@subcategory[1]', 'VARCHAR(20)') AS subcategory,
    x.item.value('description[1]', 'VARCHAR(20)') AS description,
    x.item.value('qty[1]', 'INT') AS qty
FROM @x.nodes('//orderInfo/item') AS x(item)
WHERE x.item.value('@code[1]', 'VARCHAR(20)') = 'A002'

/*
OUTPUT:

ItemCode             subcategory         description         qty
-------------------- -------------------- -------------------- -----------
A002                 Biscuits             Britania Good Day    15

(1 row(s) affected)
*/
```

Example 9

```
/*
    The following example applies a filter on the value
    of an element.
*/
DECLARE @x XML
SET @x = '
<orderInfo>
    <item code="A001" category="FOOD" subcategory="Candies">
        <description>Nestle Munch</description>
        <qty>10</qty>
        <rate>11.25</rate>
    </item>
    <item code="A002" category="FOOD" subcategory="Biscuits">
        <description>Britania Good Day</description>
        <qty>15</qty>
        <rate>12.25</rate>
    </item>
```

```
</orderInfo>'

SELECT
    x.item.value('@code[1]', 'VARCHAR(20)') AS ItemCode,
    x.item.value('@subcategory[1]', 'VARCHAR(20)') AS subcategory,
    x.item.value('description[1]', 'VARCHAR(20)') AS description,
    x.item.value('qty[1]', 'INT') AS qty
FROM @x.nodes('//orderInfo/item') AS x(item)
WHERE x.item.value('description[1]', 'VARCHAR(20)') = 'Britania Good Day'

/*
OUTPUT:

ItemCode             subcategory         description         qty
-------------------- -------------------- -------------------- -----------
A002                 Biscuits             Britania Good Day    15

(1 row(s) affected)
*/
```

Conclusions

In this article, I have presented a few examples using the *Value()* method exposed by the XML data type. I will cover the other methods in a later article.
