



More Advanced XML Processing Examples

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](#)

```

1  /*
2      The following TSQL retrieves attribute values from the XML variable.
3      Attribute names are prefixed with "@".
4  */
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">
15         <description>Britania Good Day</description>
16         <qty>15</qty>
17         <rate>12.25</rate>
18     </item>
19 </orderInfo>'
20
21 SELECT
22     x.item.value('@code[1]', 'VARCHAR(20)') AS ItemCode,
23     x.item.value('@category[1]', 'VARCHAR(20)') AS category,
24     x.item.value('@subcategory[1]', 'VARCHAR(20)') AS subcategory
25 FROM @x.nodes('//orderInfo/item') AS x(item)
26
27 /*
28 OUTPUT:
29
30 ItemCode          category          subcategory
31 -----
32 A001              FOOD             Candies
33 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
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">
15         <description>Britania Good Day</description>
16         <qty>15</qty>
17         <rate>12.25</rate>
18     </item>
19 </orderInfo>'

```

```

20
21 SELECT
22     x.item.value('@code[1]', 'VARCHAR(20)') AS ItemCode,
23     x.item.value('@category[1]', 'VARCHAR(20)') AS category,
24     x.item.value('@subcategory[1]', 'VARCHAR(20)') AS subcategory,
25     x.item.value('description[1]', 'VARCHAR(20)') AS description,
26     x.item.value('qty[1]', 'INT') AS qty,
27     x.item.value('rate[1]', 'FLOAT') AS rate
28 FROM @x.nodes('//orderInfo/item') AS x(item)
29
30 /*
31 OUTPUT:
32
33 ItemCode          category          subcategory          description
qty      rate
34 -----
35 A001          FOOD          Candies          Nestle Munch
10      11.25
36 A002          FOOD          Biscuits          Britania Good Day
15      12.25
37
38 (2 row(s) affected)
39 */

```

Example 4

```

1  /*
2  The following example demonstrates how to extract the value
3  from a given row. This example extracts a value from the first
4  row.
5
6  The first example selects the value from the first row.
7  The second example adds an alias to the result column.
8  The third example assigns the result to a variable.
9  */
10
11 DECLARE @x XML
12 SET @x = '
13 <orderInfo>
14     <item code="A001" category="FOOD" subcategory="Candies">
15         <description>Nestle Munch</description>
16         <qty>10</qty>
17         <rate>11.25</rate>
18     </item>
19     <item code="A002" category="FOOD" subcategory="Biscuits">
20         <description>Britania Good Day</description>
21         <qty>15</qty>
22         <rate>12.25</rate>
23     </item>
24 </orderInfo>'
25
26 SELECT @x.value('(/orderInfo/item/@code)[1]', 'VARCHAR(20)')
27
28 SELECT @x.value('(/orderInfo/item/@code)[1]', 'VARCHAR(20)') AS Code
29
30 DECLARE @code VARCHAR(20)
31 SELECT @code = @x.value('(/orderInfo/item/@code)[1]', 'VARCHAR(20)')
32 SELECT @code as Code
33
34 /*
35 OUTPUT:
36 -----
37 A001
38
39 (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

```

1  /*
2      The following example retrieves the value of an
3      element from the first row.
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/description)[1]', 'VARCHAR(20)')
21 /*
22 OUTPUT:
23
24 -----
25 Nestle Munch
26
27 (1 row(s) affected)
28 */
29

```

Example 7

```

1  /*
2      The following example retrieves the value of an
3      element from the second row.
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/description)[2]', 'VARCHAR(20)')
21 /*
22 OUTPUT:
23
24 -----
25 Britania Good Day
26
27 (1 row(s) affected)
28 */
29

```

Example 8

```

1  /*
2      Now let us have a look at filtering results. The following
3      example applies a filter on an attribute value.
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
21     x.item.value('@code[1]', 'VARCHAR(20)') AS ItemCode,
22     x.item.value('@subcategory[1]', 'VARCHAR(20)') AS subcategory,
23     x.item.value('description[1]', 'VARCHAR(20)') AS description,
24     x.item.value('qty[1]', 'INT') AS qty
25 FROM @x.nodes('//orderInfo/item') AS x(item)
26 WHERE x.item.value('@code[1]', 'VARCHAR(20)') = 'A002'
27
28 /*
29 OUTPUT:
30
31 ItemCode          subcategory          description          qty
32 -----
33 A002              Biscuits              Britania Good Day    15
34
35 (1 row(s) affected)
36 */

```

Example 9

```

1  /*
2      The following example applies a filter on the value
3      of an element.
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
21     x.item.value('@code[1]', 'VARCHAR(20)') AS ItemCode,
22     x.item.value('@subcategory[1]', 'VARCHAR(20)') AS subcategory,
23     x.item.value('description[1]', 'VARCHAR(20)') AS description,
24     x.item.value('qty[1]', 'INT') AS qty
25 FROM @x.nodes('//orderInfo/item') AS x(item)
26 WHERE x.item.value('description[1]', 'VARCHAR(20)') = 'Britania Good Day'
27
28 /*
29 OUTPUT:
30
31 ItemCode          subcategory          description          qty
32 -----
33 A002              Biscuits              Britania Good Day    15
34
35 (1 row(s) affected)
36 */

```

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.

Copyright © 2002-2007 Simple Talk Publishing. All Rights Reserved.