

<http://www.sqlservercentral.com/articles/XML/63535/>
 Printed 2008/07/18 08:31AM

XML Workshop XXI - Generating an ATOM 1.0 Feed with FOR XML EXPLICIT

By [Jacob Sebastian](#), 2008/06/25

Introduction

The last [few sessions](#) of [XML Workshop](#) were focussing on generating RSS and ATOM feeds. Web Feeds (RSS or ATOM) are a must-have component of any modern web site. In the previous session, we have seen how to generate an ATOM 1.0 feed using FOR XML PATH. PATH is a new keyword introduced with SQL Server 2005 and is not available in SQL Server 2000. In this session, we will use FOR XML with EXPLICIT to generate an ATOM 1.0 feed. EXPLICIT is available in SQL Server 2000, 2005 and 2008.

You can find the previous sessions on RSS/ATOM feed generation [here](#). All the previous sessions of XML Workshop are listed [here](#). Several people asked me for an RSS feed for the XML Workshop series and I have created one [here](#).

Let us move to the sample feed and sample data. Then we will start writing the FOR XML EXPLICIT code needed to generate the sample feed based on the sample data.

Sample Feed

This is the feed that we are trying to generate with FOR XML EXPLICIT.

```
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQLAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
  <title type="html">Welcome to XML Workshop</title>
  <subtitle type="html">
    A collection of short articles on SQL Server and XML
  </subtitle>
  <link rel="alternate" type="text/html" href="http://www.sqlserverandxml.com/" />
  <link rel="self" type="application/atom+xml"
    href="http://www.sqlserverandxml.com-a.googlepages.com/TSQLAtom10.xml" />
  <generator uri="http://www.sqlserverandxml.com/" version="1.0">
    FOR XML
  </generator>
  <entry>
    <title>Sales Order Workshop</title>
    <id>http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop</id>
    <published>2005-11-24T00:25:00Z</published>
    <updated>2005-11-24T00:25:00Z</updated>
    <content>
      A series of 4 articles which explain
      how to pass variable number of parameters to a stored procedure
      using XML
    </content>
  </entry>
</feed>
```

```

    <link rel="alternate" type="text/html"
        href="http://www.sqlserverandxml.com-
a.googlepages.com/salesorderworkshop">
    http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop</link>
    <author>
        <name>Jacob Sebastian</name>
        <uri>http://www.sqlserverandxml.com</uri>
    </author>
</entry>
<entry>
    <title>FOR XML Workshop</title>
    <id>http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop</id>
    <published>2005-10-14T02:17:00Z</published>
    <updated>2005-10-14T02:17:00Z</updated>
    <content>
        A collection of short articles that
        explain how to generate XML output using TSQL keyword FOR
        XML.
    </content>
    <link rel="alternate" type="text/html"
        href="http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop">
    http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop</link>
    <author>
        <name>Jacob Sebastian</name>
        <uri>http://www.sqlserverandxml.com</uri>
    </author>
</entry>
</feed>

```

Now let us have a look at the tables and the sample data. Here is the script to generate the sample tables.

```

-- table for the feed information
IF OBJECT_ID('feed') IS NOT NULL DROP TABLE feed
CREATE TABLE feed(
    title VARCHAR(100),
    subtitle VARCHAR(200),
    id VARCHAR(100),
    link VARCHAR(100),
    generator VARCHAR(20),
    updated DATETIME )
GO

IF OBJECT_ID('entry') IS NOT NULL DROP TABLE entry
-- table to store the entries
CREATE TABLE entry(
    entryID INT IDENTITY,
    title VARCHAR(100),
    link VARCHAR(100),
    published DATETIME,
    updated DATETIME,
    content VARCHAR(1000),
    authorname VARCHAR(30),
    authorurl VARCHAR(100))
GO

```

And here is script to populate the tables with some sample data.

```

-- populate the 'feed' table
INSERT INTO feed (
    title,
    subtitle,
    id,
    link,
    generator,
    updated )
SELECT
    'Welcome to XML Workshop',
    'A collection of short articles on SQL Server and XML',
    'http://www.sqlserverandxml.com-a.googlepages.com/TSQlAtom10.xml',
    'http://www.sqlserverandxml.com/',
    'FOR XML',
    '2005-10-14T03:17:00'

-- populate the 'entry' table
INSERT INTO entry(
    title,
    link,
    published,
    updated,
    content,
    authorname,
    authorurl )
SELECT
    'Sales Order Workshop',
    'http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop',
    '2005-11-24T00:25:00',
    '2005-11-24T00:25:00',
    'A series of 4 articles which explain
    how to pass variable number of parameters to a stored procedure
    using XML',
    'Jacob Sebastian',
    'http://www.sqlserverandxml.com'
UNION ALL
SELECT
    'FOR XML Workshop',
    'http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop',
    '2005-10-14T02:17:00',
    '2005-10-14T02:17:00',
    'A collection of short articles that
    explain how to generate XML output using TSQL keyword FOR
    XML.',
    'Jacob Sebastian',
    'http://www.sqlserverandxml.com/'

```

We have got the sample tables and sample data. Let us move ahead and write the code. We will develop the entire code in several small steps. We will start with a basic piece of code and will enhance it at every step. There will be some repetition, but I think it will help to understand FOR XML EXPLICIT behavior better.

Step 1 - Create the root element

At the step, lets create root element of the feed. The query is quite simple.

```

SELECT

```

```

1 AS Tag,
NULL AS Parent,
NULL AS 'feed!1!',
'http://www.w3.org/2005/Atom' AS 'feed!1!xmlns'
FROM feed
FOR XML EXPLICIT

```

```

<feed xmlns="http://www.w3.org/2005/Atom" />

```

Step 2

In the previous step we created the root element "feed". At this step, we will generate the child elements of "feed". "feed" has two types of child elements. Some of the child elements have their own children and others do not have. In this step, let's generate "id" and "updated", the two child elements of "feed" that do not have other children.

```

SELECT
1 AS Tag,
NULL AS Parent,
NULL AS 'feed!1!',
'http://www.w3.org/2005/Atom' AS 'feed!1!xmlns',
id AS 'feed!1!id!element',
CONVERT(nvarchar,updated,127) + 'Z' AS 'feed!1!updated!element'
FROM feed
FOR XML EXPLICIT

```

```

<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQLEAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
</feed>

```

Step 3

Some of the children of "feed" do have their own children. One such example is the "title" element, which has an attribute: "type".

```

SELECT
1 AS Tag,
NULL AS Parent,
NULL AS 'feed!1!',
'http://www.w3.org/2005/Atom' AS 'feed!1!xmlns',
id AS 'feed!1!id!element',
CONVERT(nvarchar,updated,127) + 'Z' AS 'feed!1!updated!element',
NULL AS 'title!2!',
NULL AS 'title!2!type'
FROM feed
UNION ALL
SELECT
2 AS Tag,
1 AS Parent,
NULL, NULL, NULL, NULL,
title, 'html'
FROM feed

```

```
FOR XML EXPLICIT
```

```
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQLAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
  <title type="html">Welcome to XML Workshop</title>
</feed>
```

Step 4

Now let us generate the "subtitle" element.

```
SELECT
  1 AS Tag,
  NULL AS Parent,
  NULL AS 'feed!1!',
  'http://www.w3.org/2005/Atom' AS 'feed!1!xmlns',
  id AS 'feed!1!id!element',
  CONVERT(nvarchar,updated,127) + 'Z' AS 'feed!1!updated!element',
  NULL AS 'title!2!',
  NULL AS 'title!2!type',
  NULL AS 'subtitle!3!',
  NULL AS 'subtitle!3!type'
FROM feed
UNION ALL
SELECT
  2 AS Tag,
  1 AS Parent,
  NULL, NULL, NULL, NULL,
  title, 'html',
  NULL, NULL
FROM feed
UNION ALL
SELECT
  3 AS Tag,
  1 AS Parent,
  NULL, NULL, NULL, NULL,
  NULL, NULL,
  subtitle, 'html'
FROM feed
FOR XML EXPLICIT
```

```
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQLAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
  <title type="html">Welcome to XML Workshop</title>
  <subtitle type="html">
    A collection of short articles on SQL Server and XML
  </subtitle>
</feed>
```

Step 5

Generating the "link" element.

```

SELECT
    1 AS Tag,
    NULL AS Parent,
    NULL AS 'feed!1!',
    'http://www.w3.org/2005/Atom' AS 'feed!1!xmlns',
    id AS 'feed!1!id!element',
    CONVERT(nvarchar,updated,127) + 'Z' AS 'feed!1!updated!element',
    NULL AS 'title!2!',
    NULL AS 'title!2!type',
    NULL AS 'subtitle!3!',
    NULL AS 'subtitle!3!type',
    NULL AS 'link!4!rel',
    NULL AS 'link!4!type',
    NULL AS 'link!4!href'
FROM feed
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    title, 'html',
    NULL, NULL,
    NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    3 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    subtitle, 'html',
    NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'alternate', 'text/html', link
FROM feed
FOR XML EXPLICIT

```

```

<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQlAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
  <title type="html">Welcome to XML Workshop</title>
  <subtitle type="html">
    A collection of short articles on SQL Server and XML
  </subtitle>
  <link rel="alternate" type="text/html"
    href="http://www.sqlserverandxml.com/" />
</feed>

```

Step 6

Since we need two link elements ("self" and "alternate"), we need to add one more UNION ALL.

```

SELECT
    1 AS Tag,
    NULL AS Parent,
    NULL AS 'feed!1!',
    'http://www.w3.org/2005/Atom' AS 'feed!1!xmlns',
    id AS 'feed!1!id!element',
    CONVERT(nvarchar,updated,127) + 'Z'
        AS 'feed!1!updated!element',
    NULL AS 'title!2!',
    NULL AS 'title!2!type',
    NULL AS 'subtitle!3!',
    NULL AS 'subtitle!3!type',
    NULL AS 'link!4!rel',
    NULL AS 'link!4!type',
    NULL AS 'link!4!href'
FROM feed
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    title, 'html',
    NULL, NULL,
    NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    3 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    subtitle, 'html',
    NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'alternate', 'text/html', link
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'self', 'application/atom+xml', id
FROM feed
FOR XML EXPLICIT

```

```

<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQIAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
  <title type="html">Welcome to XML Workshop</title>
  <subtitle type="html">

```

```

A collection of short articles on SQL Server and XML
</subtitle>
<link rel="alternate" type="text/html"
      href="http://www.sqlserverandxml.com/" />
<link rel="self" type="application/atom+xml"
      href="http://www.sqlserverandxml.com-a.googlepages.com/TSQlAtom10.xml" />
</feed>

```

Step 7

Let us write the code for adding the "generator" element.

```

SELECT
  1 AS Tag,
  NULL AS Parent,
  NULL AS 'feed!1!',
  'http://www.w3.org/2005/Atom' AS 'feed!1!xmlns',
  id AS 'feed!1!id!element',
  CONVERT(nvarchar,updated,127) + 'Z'
    AS 'feed!1!updated!element',
  NULL AS 'title!2!',
  NULL AS 'title!2!type',
  NULL AS 'subtitle!3!',
  NULL AS 'subtitle!3!type',
  NULL AS 'link!4!rel',
  NULL AS 'link!4!type',
  NULL AS 'link!4!href',
  NULL AS 'generator!5',
  NULL AS 'generator!5!uri',
  NULL AS 'generator!5!version'
FROM feed
UNION ALL
SELECT
  2 AS Tag,
  1 AS Parent,
  NULL, NULL, NULL, NULL,
  title, 'html',
  NULL, NULL,
  NULL, NULL, NULL,
  NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
  3 AS Tag,
  1 AS Parent,
  NULL, NULL, NULL, NULL,
  NULL, NULL,
  subtitle, 'html',
  NULL, NULL, NULL,
  NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
  4 AS Tag,
  1 AS Parent,
  NULL, NULL, NULL, NULL,
  NULL, NULL,
  NULL, NULL,

```



```

        'alternate', 'text/html', link,
        NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'self', 'application/atom+xml', id,
    NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    5 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    generator, link, '1.0'
FROM feed
FOR XML EXPLICIT

```

```

<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQlAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
  <title type="html">Welcome to XML Workshop</title>
  <subtitle type="html">
    A collection of short articles on SQL Server and XML
  </subtitle>
  <link rel="alternate" type="text/html"
    href="http://www.sqlserverandxml.com/" />
  <link rel="self" type="application/atom+xml"
    href="http://www.sqlserverandxml.com-a.googlepages.com/TSQlAtom10.xml" />
  <generator uri="http://www.sqlserverandxml.com/" version="1.0">
    FOR XML
  </generator>
</feed>

```

Step 8

We are done with the "feed" element. Let us now start adding the "entries". Some of the children of "entry" have children and others do not. At this step, let us generate the elements, which do not have children.

```

SELECT
    1 AS Tag,
    NULL AS Parent,
    NULL AS 'feed!1!',
    'http://www.w3.org/2005/Atom' AS 'feed!1!xmlns',
    id AS 'feed!1!id!element',
    CONVERT(nvarchar, updated, 127) + 'Z'
        AS 'feed!1!updated!element',
    NULL AS 'title!2!',

```

```

        NULL AS 'title!2!type',
        NULL AS 'subtitle!3',
        NULL AS 'subtitle!3!type',
        NULL AS 'link!4!rel',
        NULL AS 'link!4!type',
        NULL AS 'link!4!href',
        NULL AS 'generator!5',
        NULL AS 'generator!5!uri',
        NULL AS 'generator!5!version',
        NULL AS 'entry!6!title!element',
        NULL AS 'entry!6!id!element',
        NULL AS 'entry!6!published!element',
        NULL AS 'entry!6!updated!element',
        NULL AS 'entry!6!content!element'
FROM feed
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    title, 'html',
    NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    3 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    subtitle, 'html',
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    'alternate', 'text/html', link,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'self', 'application/atom+xml', id,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL
FROM feed

```

```

UNION ALL
SELECT
    5 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    generator, link, '1.0',
    NULL, NULL, NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    6 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    title, link, CONVERT(nvarchar,published,127) + 'Z',
    CONVERT(nvarchar,updated,127) + 'Z', content
FROM entry
FOR XML EXPLICIT

```

```

<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQLAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
  <title type="html">Welcome to XML Workshop</title>
  <subtitle type="html">
    A collection of short articles on SQL Server and XML
  </subtitle>
  <link rel="alternate" type="text/html"
    href="http://www.sqlserverandxml.com/" />
  <link rel="self" type="application/atom+xml"
    href="http://www.sqlserverandxml.com-a.googlepages.com/TSQLAtom10.xml" />
  <generator uri="http://www.sqlserverandxml.com/" version="1.0">
    FOR XML
  </generator>
  <entry>
    <title>Sales Order Workshop</title>
    <id>http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop</id>
    <published>2005-11-24T00:25:00Z</published>
    <updated>2005-11-24T00:25:00Z</updated>
    <content>
      A series of 4 articles which explain
      how to pass variable number of parameters to a stored procedure
      using XML
    </content>
  </entry>
  <entry>
    <title>FOR XML Workshop</title>
    <id>http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop</id>
    <published>2005-10-14T02:17:00Z</published>
    <updated>2005-10-14T02:17:00Z</updated>
    <content>
      A collection of short articles that
      explain how to generate XML output using TSQL keyword FOR
      XML.
    </content>
  </entry>
</feed>

```

```

    </content>
  </entry>
</feed>

```

Step 9

Let us add the "link" element.

```

SELECT
  1 AS Tag,
  NULL AS Parent,
  NULL AS 'feed!1!',
  'http://www.w3.org/2005/Atom' AS 'feed!1:xmlns',
  id AS 'feed!1:id!element',
  CONVERT(nvarchar,updated,127) + 'Z'
    AS 'feed!1:updated!element',
  NULL AS 'title!2!',
  NULL AS 'title!2:type',
  NULL AS 'subtitle!3',
  NULL AS 'subtitle!3:type',
  NULL AS 'link!4:rel',
  NULL AS 'link!4:type',
  NULL AS 'link!4:href',
  NULL AS 'generator!5',
  NULL AS 'generator!5:uri',
  NULL AS 'generator!5:version',
  NULL AS 'entry!6:title!element',
  NULL AS 'entry!6:id!element',
  NULL AS 'entry!6:published!element',
  NULL AS 'entry!6:updated!element',
  NULL AS 'entry!6:content!element',
  NULL AS 'link!7!',
  NULL AS 'link!7:rel',
  NULL AS 'link!7:type',
  NULL AS 'link!7:href'
FROM feed
UNION ALL
SELECT
  2 AS Tag,
  1 AS Parent,
  NULL, NULL, NULL, NULL,
  title, 'html',
  NULL, NULL,
  NULL, NULL, NULL,
  NULL, NULL, NULL,
  NULL, NULL, NULL, NULL, NULL,
  NULL, NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
  3 AS Tag,
  1 AS Parent,
  NULL, NULL, NULL, NULL,
  NULL, NULL,
  subtitle, 'html',
  NULL, NULL, NULL,
  NULL, NULL, NULL,
  NULL, NULL, NULL, NULL, NULL,

```

```

        NULL, NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'alternate', 'text/html', link,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'self', 'application/atom+xml', id,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    5 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    generator, link, '1.0',
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL
FROM feed
UNION ALL
SELECT
    6 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    title, link, CONVERT(nvarchar,published,127) + 'Z',
    CONVERT(nvarchar,updated,127) + 'Z', content,
    NULL, NULL, NULL, NULL
FROM entry
UNION ALL
SELECT
    7 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,

```

```

        NULL, NULL, NULL,
        NULL, NULL, NULL, NULL, NULL,
        link, 'alternate', 'text/html', link
FROM entry
FOR XML EXPLICIT

```

```

<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQLAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
  <title type="html">Welcome to XML Workshop</title>
  <subtitle type="html">
    A collection of short articles on SQL Server and XML
  </subtitle>
  <link rel="alternate" type="text/html"
    href="http://www.sqlserverandxml.com/" />
  <link rel="self" type="application/atom+xml"
    href="http://www.sqlserverandxml.com-a.googlepages.com/TSQLAtom10.xml" />
  <generator uri="http://www.sqlserverandxml.com/" version="1.0">
    FOR XML
  </generator>
  <entry>
    <title>Sales Order Workshop</title>
    <id>http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop</id>
    <published>2005-11-24T00:25:00Z</published>
    <updated>2005-11-24T00:25:00Z</updated>
    <content>
      A series of 4 articles which explain
      how to pass variable number of parameters to a stored procedure
      using XML
    </content>
  </entry>
  <entry>
    <title>FOR XML Workshop</title>
    <id>http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop</id>
    <published>2005-10-14T02:17:00Z</published>
    <updated>2005-10-14T02:17:00Z</updated>
    <content>
      A collection of short articles that
      explain how to generate XML output using TSQL keyword FOR
      XML.
    </content>
  </entry>
  <link rel="alternate" type="text/html"
    href="http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop">
    http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop
  </link>
  <link rel="alternate" type="text/html"
    href="http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop">
    http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop
  </link>
</feed>

```

Well, we landed with the sorting problem again. We saw this in the previous sessions where we discussed FOR XML EXPLICIT. When using FOR XML EXPLICIT, we need to make sure that the query returns rows in the same order as you need them in the XML document. You will see that the above XML is not correct, because the "link" element of both "entry" nodes are positioned at the bottom

of the XML document. Each "link" element should be placed inside its own "entry" node. To achieve this, we need to add some kind of sorting logic to the query. We need to make sure that the row for the "link" element of the first "entry" should appear immediately after the row of the first "entry".

Step 10

Let us add some sorting logic to make sure that the query returns rows in the correct order.

```
SELECT
    Tag,
    Parent,
    [feed!1!],
    [feed!1!xmlns],
    [feed!1!id!element],
    [feed!1!updated!element],
    [title!2!],
    [title!2!type],
    [subtitle!3!],
    [subtitle!3!type],
    [link!4!rel],
    [link!4!type],
    [link!4!href],
    [generator!5!],
    [generator!5!uri],
    [generator!5!version],
    [entry!6!title!element],
    [entry!6!id!element],
    [entry!6!published!element],
    [entry!6!updated!element],
    [entry!6!content!element],
    [link!7!],
    [link!7!rel],
    [link!7!type],
    [link!7!href]
FROM (
    SELECT
        1 AS Tag,
        NULL AS Parent,
        NULL AS 'feed!1!',
        'http://www.w3.org/2005/Atom' AS 'feed!1!xmlns',
        id AS 'feed!1!id!element',
        CONVERT(nvarchar,updated,127) + 'Z'
            AS 'feed!1!updated!element',
        NULL AS 'title!2!',
        NULL AS 'title!2!type',
        NULL AS 'subtitle!3!',
        NULL AS 'subtitle!3!type',
        NULL AS 'link!4!rel',
        NULL AS 'link!4!type',
        NULL AS 'link!4!href',
        NULL AS 'generator!5!',
        NULL AS 'generator!5!uri',
        NULL AS 'generator!5!version',
        NULL AS 'entry!6!title!element',
        NULL AS 'entry!6!id!element',
        NULL AS 'entry!6!published!element',
        NULL AS 'entry!6!updated!element',
        NULL AS 'entry!6!content!element',
```

```

        NULL AS 'link!7!',
        NULL AS 'link!7!rel',
        NULL AS 'link!7!type',
        NULL AS 'link!7!href',
        0 AS EntryID, 0 AS Sort
FROM feed
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    title, 'html',
    NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    0, 0
FROM feed
UNION ALL
SELECT
    3 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    subtitle, 'html',
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    0, 0
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'alternate', 'text/html', link,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    0, 0
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'self', 'application/atom+xml', id,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    0, 0
FROM feed
UNION ALL

```



```

SELECT
    5 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    generator, link, '1.0',
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    0, 0
FROM feed
UNION ALL
SELECT
    6 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    title, link, CONVERT(nvarchar,published,127) + 'Z',
    CONVERT(nvarchar,updated,127) + 'Z', content,
    NULL, NULL, NULL, NULL,
    entryID, 1
FROM entry
UNION ALL
SELECT
    7 AS Tag,
    6 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    link, 'alternate', 'text/html', link,
    entryID, 2
FROM entry
) A
ORDER BY EntryID, Sort
FOR XML EXPLICIT

```

```

<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQLAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
  <title type="html">Welcome to XML Workshop</title>
  <subtitle type="html">
    A collection of short articles on SQL Server and XML
  </subtitle>
  <link rel="alternate" type="text/html" href="http://www.sqlserverandxml.com/" />
  <link rel="self" type="application/atom+xml"
    href="http://www.sqlserverandxml.com-a.googlepages.com/TSQLAtom10.xml" />
  <generator uri="http://www.sqlserverandxml.com/" version="1.0">
    FOR XML
  </generator>
  <entry>
    <title>Sales Order Workshop</title>
    <id>http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop</id>

```

```

<published>2005-11-24T00:25:00Z</published>
<updated>2005-11-24T00:25:00Z</updated>
<content>
  A series of 4 articles which explain
  how to pass variable number of parameters to a stored procedure
  using XML
</content>
<link rel="alternate" type="text/html"
      href="http://www.sqlserverandxml.com-
a.googlepages.com/salesorderworkshop">
  http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop
</link>
</entry>
<entry>
  <title>FOR XML Workshop</title>
  <id>http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop</id>
  <published>2005-10-14T02:17:00Z</published>
  <updated>2005-10-14T02:17:00Z</updated>
  <content>
    A collection of short articles that
    explain how to generate XML output using TSQL keyword FOR
    XML.
  </content>
  <link rel="alternate" type="text/html"
        href="http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop">
    http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop
  </link>
</entry>
</feed>

```

Step 11

We are almost done. This is the last step and we need to generate the "author" element.

```

SELECT
  Tag,
  Parent,
  [feed!1!],
  [feed!1!xmlns],
  [feed!1!id!element],
  [feed!1!updated!element],
  [title!2!],
  [title!2!type],
  [subtitle!3],
  [subtitle!3!type],
  [link!4!rel],
  [link!4!type],
  [link!4!href],
  [generator!5],
  [generator!5!uri],
  [generator!5!version],
  [entry!6!title!element],
  [entry!6!id!element],
  [entry!6!published!element],
  [entry!6!updated!element],
  [entry!6!content!element],
  [link!7!],
  [link!7!rel],

```

```

[link!7!type],
[link!7!href],
[author!8!name!element],
[author!8!uri!element]
FROM (
SELECT
    1 AS Tag,
    NULL AS Parent,
    NULL AS 'feed!1!',
    'http://www.w3.org/2005/Atom' AS 'feed!1!xmlns',
    id AS 'feed!1!id!element',
    CONVERT(nvarchar,updated,127) + 'Z'
        AS 'feed!1!updated!element',
    NULL AS 'title!2!',
    NULL AS 'title!2!type',
    NULL AS 'subtitle!3!',
    NULL AS 'subtitle!3!type',
    NULL AS 'link!4!rel',
    NULL AS 'link!4!type',
    NULL AS 'link!4!href',
    NULL AS 'generator!5',
    NULL AS 'generator!5!uri',
    NULL AS 'generator!5!version',
    NULL AS 'entry!6!title!element',
    NULL AS 'entry!6!id!element',
    NULL AS 'entry!6!published!element',
    NULL AS 'entry!6!updated!element',
    NULL AS 'entry!6!content!element',
    NULL AS 'link!7!',
    NULL AS 'link!7!rel',
    NULL AS 'link!7!type',
    NULL AS 'link!7!href',
    NULL AS 'author!8!name!element',
    NULL AS 'author!8!uri!element',
    0 AS EntryID, 0 AS Sort
FROM feed
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    title, 'html',
    NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    0, 0
FROM feed
UNION ALL
SELECT
    3 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    subtitle, 'html',
    NULL, NULL, NULL,
    NULL, NULL, NULL,

```

```

NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL,
NULL, NULL,
0, 0
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'alternate', 'text/html', link,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    0, 0
FROM feed
UNION ALL
SELECT
    4 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    'self', 'application/atom+xml', id,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    0, 0
FROM feed
UNION ALL
SELECT
    5 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    generator, link, '1.0',
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    0, 0
FROM feed
UNION ALL
SELECT
    6 AS Tag,
    1 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    title, link, CONVERT(nvarchar,published,127) + 'Z',
    CONVERT(nvarchar,updated,127) + 'Z', content,
    NULL, NULL, NULL, NULL,

```

```

        NULL, NULL,
        entryID, 1
FROM entry
UNION ALL
SELECT
    7 AS Tag,
    6 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    link, 'alternate', 'text/html', link,
    NULL, NULL,
    entryID, 2
FROM entry
UNION ALL
SELECT
    8 AS Tag,
    6 AS Parent,
    NULL, NULL, NULL, NULL,
    NULL, NULL,
    NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    authername, authorurl,
    entryID, 3
FROM entry
) A
ORDER BY EntryID, Sort
FOR XML EXPLICIT

```

```

<feed xmlns="http://www.w3.org/2005/Atom">
  <id>http://www.sqlserverandxml.com-a.googlepages.com/TSQIAAtom10.xml</id>
  <updated>2005-10-14T03:17:00Z</updated>
  <title type="html">Welcome to XML Workshop</title>
  <subtitle type="html">
    A collection of short articles on SQL Server and XML
  </subtitle>
  <link rel="alternate" type="text/html" href="http://www.sqlserverandxml.com/" />
  <link rel="self" type="application/atom+xml"
    href="http://www.sqlserverandxml.com-a.googlepages.com/TSQIAAtom10.xml" />
  <generator uri="http://www.sqlserverandxml.com/" version="1.0">
    FOR XML
  </generator>
  <entry>
    <title>Sales Order Workshop</title>
    <id>http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop</id>
    <published>2005-11-24T00:25:00Z</published>
    <updated>2005-11-24T00:25:00Z</updated>
    <content>
      A series of 4 articles which explain
      how to pass variable number of parameters to a stored procedure
      using XML
    </content>
    <link rel="alternate" type="text/html"

```

```

        href="http://www.sqlserverandxml.com-
a.googlepages.com/salesorderworkshop">
    http://www.sqlserverandxml.com-a.googlepages.com/salesorderworkshop
</link>
<author>
    <name>Jacob Sebastian</name>
    <uri>http://www.sqlserverandxml.com</uri>
</author>
</entry>
<entry>
    <title>FOR XML Workshop</title>
    <id>http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop</id>
    <published>2005-10-14T02:17:00Z</published>
    <updated>2005-10-14T02:17:00Z</updated>
    <content>
        A collection of short articles that
        explain how to generate XML output using TSQL keyword FOR
        XML.
    </content>
    <link rel="alternate" type="text/html"
        href="http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop">
        http://www.sqlserverandxml.com-a.googlepages.com/forxmlworkshop
    </link>
    <author>
        <name>Jacob Sebastian</name>
        <uri>http://www.sqlserverandxml.com</uri>
    </author>
</entry>
</feed>

```

Congratulations! We have the feed we wanted. You can validate the feed we just generated with FeedValidator.org or any of the validator of your choice. I have validated it only with FeedValidator.org. If your feed validator reports a problem with the above feed, please let me know.

Conclusions

We just saw another session on FOR XML EXPLICIT. Those of you who have good control over FOR XML EXPLICIT might find quite a lot of repetitive stuff here. FOR XML EXPLICIT seems to be very confusing to many people. The most complicated part is that people often land up with errors after writing a lengthy query. I am more in favor of taking "baby-steps" by using a step-by-step approach for writing FOR XML EXPLICIT queries. At each step we could add the code for a new element and test the results. If an error occurs, we could rollback to the previous step and quickly fix/spot the problem.

I have used this step-by-step approach to help many people who contacted me with strange errors/behaviors with lengthy FOR XML EXPLICIT queries. If you are a new-bie and trying to write a FOR XML EXPLICIT query, I would suggest using a step-by-step approach as given in this session.

Copyright © 2002-2008 Simple Talk Publishing. All Rights Reserved. [Privacy Policy](#). [Terms of Use](#)