http://www.sqlservercentral.com/articles/XML/63043/
Printed 2008/06/05 06:29PM

XML Workshop XX - Generating an RSS 2.0 Feed with TSQL(SQL server 2000)

By Jacob Sebastian, 2008/05/08

Introduction

In XML Workshop XVIII, we have seen how to generate an RSS 2.0 feed from *TSQL*. The session explained the feed generation process step by step and used *FOR XML PATH* to generate a valid **RSS 2.0** feed.

FOR XML PATH is a very powerful keyword that provides a great deal of control over the structure of the *XML* document being generated. We could generate very complex *XML* structures by using *FOR XML* with *PATH*. *PATH* is a new keyword introduced with SQL Server 2005 and hence it is not available in SQL Server 2000. The focus of this session is to write the *TSQL* code for SQL Server 2000 that generates a valid *RSS 2.0* feed. Since *PATH* is not available in SQL Server 2000, we will use *FOR XML* with *EXPLICIT* to generate the feed. In the previous sessions of XML Workshop, we have had a good look into *TSQL* keyword *FOR XML* along with *AUTO, RAW, PATH* and *EXPLICIT*.

Sample Feed

For the purpose of this example, let us assume that we need to create an *RSS 2.0* feed that contains all the articles in the XML Workshop series. To keep the examples simple, we will process only two records. Here is the output that we expect to generate by the end of this LAB.

```
<rss version="2.0">
  <channel>
    <title>Welcome to XML Workshop</title>
    <link>http://www.sqlserverandxml.com/...central.html</link>
    <description>
      A collection of short articles on SQL Server and XML
    </description>
    <webMaster>jacob@dotnetquest.com (Jacob Sebastian)</webMaster>
    <language>en-us</language>
    <copyright>Jacob Sebastian. All rights reserved.</copyright>
    <lastBuildDate>Wed, 12 Mar 2008 23:45:02 GMT</lastBuildDate>
    <ttl>100</ttl>
    <image>
      <url>http://www.sqlserverandxml.com/image.jpg</url>
      <title>Welcome to XML Workshop</title>
      <link>http://www.sqlserverandxml.com/...central.html</link>
      <width>144</width>
      <height>22</height>
    </image>
    <item>
      <title>XML Workshop I - Generating XML with FOR XML</title>
      <link>http://www.sqlservercentral.com/...2982.asp</link>
      <description>
        A short article that explains how to generate XML output
        with TSQL keyword FOR XML
      </description>
      <pubDate>Wed, 12 Mar 2008 23:45:02 GMT</pubDate>
      <guid isPermaLink="true">http://www.sqlservercentral.com/...2982.asp</guid>
    </item>
    <item>
```

```
      <title>XML Workshop II - Reading values from XML variables</title>
      <link>http://www.sqlservercentral.com/...2996/</link>
      <description>
        This article explains how to read values from an XML variable
        using XQuery
      </description>
      <pubDate>Wed, 12 Mar 2008 23:45:02 GMT</pubDate>
      <guid isPermaLink="true">http://www.sqlservercentral.com/...2996/</guid>
    </item>
  </channel>
</rss>
```

Sample Tables and Data

Let us create two tables to store the data needed for this LAB. We need one table to store the information about the *RSS Channel* and another table for storing the data of each RSS item. Here is the script for those tables.

```
IF OBJECT_ID('channel') IS NOT NULL DROP TABLE Channel
GO

CREATE TABLE channel(
    Title VARCHAR(100),
    Link VARCHAR(100),
    Description VARCHAR(200),
    WebMaster VARCHAR(50),
    Language VARCHAR(20),
    ImageUrl VARCHAR(100),
    ImageTitle VARCHAR(100),
    ImageLink VARCHAR(100),
    ImageWidth SMALLINT,
    ImageHeight SMALLINT,
    CopyRight VARCHAR(100),
    LastBuildDate DATETIME,
    ttl SMALLINT )
GO

IF OBJECT_ID('Articles') IS NOT NULL DROP TABLE Articles
GO

CREATE TABLE Articles(
    ArticleID INT IDENTITY(1,1),
    Title VARCHAR(100),
    Link VARCHAR(100),
    Description VARCHAR(200),
    Guid VARCHAR(100),
    PubDate DATETIME )
GO
```

Here is the code to populate the tables with some sample data

```
INSERT INTO channel (
    Title,
    Link,
    Description,
    Webmaster,
    Language,
    ImageUrl,
    ImageTitle,
    ImageLink,
    ImageWidth,
    ImageHeight,
    CopyRight,
    LastBuildDate,
    ttl)
SELECT
```

```
    'Welcome to XML Workshop',
    'http://www.sqlserverandxml.com/...central.html',
    'A collection of short articles on SQL Server and XML',
    'jacob@dotnetquest.com (Jacob Sebastian)',
    'en-us',
    'http://www.sqlserverandxml.com/image.jpg',
    'Welcome to XML Workshop',
    'http://www.sqlserverandxml.com/...central.html',
    144,
    22,
    'Jacob Sebastian. All rights reserved.',
    '2008-03-12 23:45:02',
    100


INSERT INTO Articles (
    Title,
    Link,
    Description,
    Guid,
    PubDate )
SELECT
    'XML Workshop I - Generating XML with FOR XML',
    'http://www.sqlservercentral.com/...2982.asp',
    'A short article that explains how to generate XML output
    with TSQL keyword FOR XML',
    'http://www.sqlservercentral.com/...2982.asp',
    '2008-03-12 23:45:02'
UNION ALL
SELECT
    'XML Workshop II - Reading values from XML variables',
    'http://www.sqlservercentral.com/...2996/',
    'This article explains how to read values from an XML variable
    using XQuery',
    'http://www.sqlservercentral.com/...2996/',
    '2008-03-12 23:45:02'
```

Generating the feed

Let us start writing the TSQL code to generate the *feed*. Let us break the task into different steps and attempt one step at a time.

Step 1

Let us generate the root element at this step. The root element of an RSS 2.0 feed is the *rss* element.

```
SELECT
    1 AS Tag,
    NULL AS Parent,
    '2.0' AS 'rss!1!version'
FOR XML EXPLICIT
```

```
<rss version="2.0" />
```

Step 2

Let us generate the *channel* element at this step. The channel element is little complicated because it has a number of child elements and some of the child elements have their children too. So at this step, let us just create a basic declaration of the *channel* element.

```
SELECT
    1 AS Tag,
```

```
    NULL AS Parent,
    '2.0' AS 'rss!1!version',
    NULL AS 'channel!2!title!element'
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL,
    Title
FROM channel

FOR XML EXPLICIT
```

```
<rss version="2.0">
  <channel>
    <title>Welcome to XML Workshop</title>
  </channel>
</rss>
```

Step 3

Let us enhance the code a little more so that it includes all the child elements of *channel*.

```
SELECT
    1 AS Tag,
    NULL AS Parent,
    '2.0' AS 'rss!1!version',
    NULL AS 'channel!2!title!element',
    NULL AS 'channel!2!link!element',
    NULL AS 'channel!2!description!element',
    NULL AS 'channel!2!webMaster!element',
    NULL AS 'channel!2!language!element',
    NULL AS 'channel!2!copyright!element',
    NULL AS 'channel!2!lastBuildDate!element',
    NULL AS 'channel!2!ttl!element'
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL,
    Title ,
    Link,
    Description,
    WebMaster,
    Language,
    CopyRight,
    LEFT(DATENAME(dw, LastBuildDate),3) + ', ' +
                STUFF(CONVERT(nvarchar,LastBuildDate,113),21,4,' GMT'),
    ttl
FROM channel

FOR XML EXPLICIT
```

```
<rss version="2.0">
  <channel>
    <title>Welcome to XML Workshop</title>
    <link>http://www.sqlserverandxml.com/...central.html</link>
    <description>A collection of short articles on SQL Server and XML</description>
    <webMaster>jacob@dotnetquest.com (Jacob Sebastian)</webMaster>
    <language>en-us</language>
    <copyright>Jacob Sebastian. All rights reserved.</copyright>
    <lastBuildDate>Wed, 12 Mar 2008 23:45:02 GMT</lastBuildDate>
    <ttl>100</ttl>
  </channel>
</rss>
```

Step 4

The structure of *channel* element is little complicated. One of its child element, *image* has other child elements too. This leads us to generate an additional level in the *XML* hierarchy. Lets us write the code to generate this structure.

```
SELECT
    1 AS Tag,
    NULL AS Parent,
    '2.0' AS 'rss!1!version',
    NULL AS 'channel!2!title!element',
    NULL AS 'channel!2!link!element',
    NULL AS 'channel!2!description!element',
    NULL AS 'channel!2!webMaster!element',
    NULL AS 'channel!2!language!element',
    NULL AS 'channel!2!copyright!element',
    NULL AS 'channel!2!lastBuildDate!element',
    NULL AS 'channel!2!ttl!element',
    NULL AS 'image!3!url!element',
    NULL AS 'image!3!title!element',
    NULL AS 'image!3!link!element',
    NULL AS 'image!3!width!element',
    NULL AS 'image!3!height!element'
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL,
    Title ,
    Link,
    Description,
    WebMaster,
    Language,
    CopyRight,
    LEFT(DATENAME(dw, LastBuildDate),3) + ', ' +
                STUFF(CONVERT(nvarchar,LastBuildDate,113),21,4,' GMT'),
    ttl,
    NULL, NULL, NULL, NULL, NULL
FROM channel
UNION ALL
SELECT
    3 AS Tag,
    2 AS Parent,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    ImageUrl,
    ImageTitle,
    ImageLink,
    ImageWidth,
    ImageHeight
FROM channel
FOR XML EXPLICIT
```

```
<rss version="2.0">
  <channel>
    <title>Welcome to XML Workshop</title>
    <link>http://www.sqlserverandxml.com/...central.html</link>
    <description>
      A collection of short articles on SQL Server and XML
    </description>
    <webMaster>jacob@dotnetquest.com (Jacob Sebastian)</webMaster>
    <language>en-us</language>
    <copyright>Jacob Sebastian. All rights reserved.</copyright>
    <lastBuildDate>Wed, 12 Mar 2008 23:45:02 GMT</lastBuildDate>
    <ttl>100</ttl>
    <image>
      <url>http://www.sqlserverandxml.com/image.jpg</url>
      <title>Welcome to XML Workshop</title>
```

```
      <link>http://www.sqlserverandxml.com/...central.html</link>
      <width>144</width>
      <height>22</height>
    </image>
  </channel>
</rss>
```

Step 5

We are done with the *channel* element. Let us move to the *item* element. Let us do it in two steps. First let us see if we can correctly generate the *item* elements with just the *title* information.

```
SELECT
    1 AS Tag,
    NULL AS Parent,
    '2.0' AS 'rss!1!version',
    NULL AS 'channel!2!title!element',
    NULL AS 'channel!2!link!element',
    NULL AS 'channel!2!description!element',
    NULL AS 'channel!2!webMaster!element',
    NULL AS 'channel!2!language!element',
    NULL AS 'channel!2!copyright!element',
    NULL AS 'channel!2!lastBuildDate!element',
    NULL AS 'channel!2!ttl!element',
    NULL AS 'image!3!url!element',
    NULL AS 'image!3!title!element',
    NULL AS 'image!3!link!element',
    NULL AS 'image!3!width!element',
    NULL AS 'image!3!height!element',
    NULL AS 'item!4!title!element'
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL,
    Title ,
    Link,
    Description,
    WebMaster,
    Language,
    CopyRight,
    LEFT(DATENAME(dw, LastBuildDate),3) + ', ' +
                STUFF(CONVERT(nvarchar,LastBuildDate,113),21,4,' GMT'),
    ttl,
    NULL, NULL, NULL, NULL, NULL,
    NULL
FROM channel
UNION ALL
SELECT
    3 AS Tag,
    2 AS Parent,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    ImageUrl,
    ImageTitle,
    ImageLink,
    ImageWidth,
    ImageHeight,
    NULL
FROM channel
UNION ALL
SELECT
    4 AS Tag,
    2 AS Parent,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    title
FROM Articles
```

```
FOR XML EXPLICIT
```

```
<rss version="2.0">
  <channel>
    <title>Welcome to XML Workshop</title>
    <link>http://www.sqlserverandxml.com/...central.html</link>
    <description>A collection of short articles on SQL Server and XML</description>
    <webMaster>jacob@dotnetquest.com (Jacob Sebastian)</webMaster>
    <language>en-us</language>
    <copyright>Jacob Sebastian. All rights reserved.</copyright>
    <lastBuildDate>Wed, 12 Mar 2008 23:45:02 GMT</lastBuildDate>
    <ttl>100</ttl>
    <image>
      <url>http://www.sqlserverandxml.com/image.jpg</url>
      <title>Welcome to XML Workshop</title>
      <link>http://www.sqlserverandxml.com/...central.html</link>
      <width>144</width>
      <height>22</height>
    </image>
    <item>
      <title>XML Workshop I - Generating XML with FOR XML</title>
    </item>
    <item>
      <title>XML Workshop II - Reading values from XML variables</title>
    </item>
  </channel>
</rss>
```

Step 6

It looks like we are getting there. Let us write the query to generate the other elements too.

```
SELECT
    1 AS Tag,
    NULL AS Parent,
    '2.0' AS 'rss!1!version',
    NULL AS 'channel!2!title!element',
    NULL AS 'channel!2!link!element',
    NULL AS 'channel!2!description!element',
    NULL AS 'channel!2!webMaster!element',
    NULL AS 'channel!2!language!element',
    NULL AS 'channel!2!copyright!element',
    NULL AS 'channel!2!lastBuildDate!element',
    NULL AS 'channel!2!ttl!element',
    NULL AS 'image!3!url!element',
    NULL AS 'image!3!title!element',
    NULL AS 'image!3!link!element',
    NULL AS 'image!3!width!element',
    NULL AS 'image!3!height!element',
    NULL AS 'item!4!title!element',
    NULL AS 'item!4!link!element',
    NULL AS 'item!4!description!element',
    NULL AS 'item!4!guid!element',
    NULL AS 'item!4!pubDate!element'
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL,
    Title ,
    Link,
    Description,
    WebMaster,
    Language,
    CopyRight,
    LEFT(DATENAME(dw, LastBuildDate),3) + ', ' +
```

```
                STUFF(CONVERT(nvarchar,LastBuildDate,113),21,4,' GMT'),
    ttl,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL
FROM channel
UNION ALL
SELECT
    3 AS Tag,
    2 AS Parent,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    ImageUrl,
    ImageTitle,
    ImageLink,
    ImageWidth,
    ImageHeight,
    NULL, NULL, NULL, NULL, NULL
FROM channel
UNION ALL
SELECT
    4 AS Tag,
    2 AS Parent,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    title,
    Link,
    Description,
    Guid,
    LEFT(DATENAME(dw, PubDate),3) + ', ' +
                        STUFF(CONVERT(nvarchar,PubDate,113),21,4,' GMT')
FROM Articles
FOR XML EXPLICIT
```

```
<rss version="2.0">
  <channel>
    <title>Welcome to XML Workshop</title>
    <link>http://www.sqlserverandxml.com/...central.html</link>
    <description>
      A collection of short articles on SQL Server and XML
    </description>
    <webMaster>jacob@dotnetquest.com (Jacob Sebastian)</webMaster>
    <language>en-us</language>
    <copyright>Jacob Sebastian. All rights reserved.</copyright>
    <lastBuildDate>Wed, 12 Mar 2008 23:45:02 GMT</lastBuildDate>
    <ttl>100</ttl>
    <image>
      <url>http://www.sqlserverandxml.com/image.jpg</url>
      <title>Welcome to XML Workshop</title>
      <link>http://www.sqlserverandxml.com/...central.html</link>
      <width>144</width>
      <height>22</height>
    </image>
    <item>
      <title>XML Workshop I - Generating XML with FOR XML</title>
      <link>http://www.sqlservercentral.com/...2982.asp</link>
      <description>
        A short article that explains how to generate XML output
        with TSQL keyword FOR XML
      </description>
      <guid>http://www.sqlservercentral.com/...2982.asp</guid>
      <pubDate>Wed, 12 Mar 2008 23:45:02 GMT</pubDate>
    </item>
    <item>
      <title>
        XML Workshop II - Reading values from XML variables
      </title>
      <link>http://www.sqlservercentral.com/...2996/</link>
      <description>
        This article explains how to read values from an XML
```

```
        variable using XQuery
      </description>
      <guid>http://www.sqlservercentral.com/...2996/</guid>
      <pubDate>Wed, 12 Mar 2008 23:45:02 GMT</pubDate>
    </item>
  </channel>
</rss>
```

Step 7

Well, we are almost done. The only remaining task is to add the attribute *isPermalink* with each item element. Let us try to add that.

```
SELECT
    1 AS Tag,
    NULL AS Parent,
    '2.0' AS 'rss!1!version',
    NULL AS 'channel!2!title!element',
    NULL AS 'channel!2!link!element',
    NULL AS 'channel!2!description!element',
    NULL AS 'channel!2!webMaster!element',
    NULL AS 'channel!2!language!element',
    NULL AS 'channel!2!copyright!element',
    NULL AS 'channel!2!lastBuildDate!element',
    NULL AS 'channel!2!ttl!element',
    NULL AS 'image!3!url!element',
    NULL AS 'image!3!title!element',
    NULL AS 'image!3!link!element',
    NULL AS 'image!3!width!element',
    NULL AS 'image!3!height!element',
    NULL AS 'item!4!title!element',
    NULL AS 'item!4!link!element',
    NULL AS 'item!4!description!element',
    NULL AS 'item!4!pubDate!element',
    NULL AS 'guid!5!isPermaLink',
    NULL AS 'guid!5!!element'
UNION ALL
SELECT
    2 AS Tag,
    1 AS Parent,
    NULL,
    Title ,
    Link,
    Description,
    WebMaster,
    Language,
    CopyRight,
    LEFT(DATENAME(dw, LastBuildDate),3) + ', ' +
                STUFF(CONVERT(nvarchar,LastBuildDate,113),21,4,' GMT'),
    ttl,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    NULL, NULL
FROM channel
UNION ALL
SELECT
    3 AS Tag,
    2 AS Parent,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    ImageUrl,
    ImageTitle,
    ImageLink,
    ImageWidth,
    ImageHeight,
    NULL, NULL, NULL, NULL,
    NULL, NULL
FROM channel
```

```
UNION ALL
SELECT
    4 AS Tag,
    2 AS Parent,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    title,
    Link,
    Description,
    LEFT(DATENAME(dw, PubDate),3) + ', ' +
                        STUFF(CONVERT(nvarchar,PubDate,113),21,4,' GMT'),
    NULL, NULL
FROM Articles
UNION ALL
SELECT
    5 AS Tag,
    4 AS Parent,
    NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL, NULL,
    NULL, NULL, NULL, NULL,
    'true',
    guid
FROM Articles
FOR XML EXPLICIT
```

```
<rss version="2.0">
  <channel>
    <title>Welcome to XML Workshop</title>
    <link>http://www.sqlserverandxml.com/...central.html</link>
    <description>
      A collection of short articles on SQL Server and XML
    </description>
    <webMaster>jacob@dotnetquest.com (Jacob Sebastian)</webMaster>
    <language>en-us</language>
    <copyright>Jacob Sebastian. All rights reserved.</copyright>
    <lastBuildDate>Wed, 12 Mar 2008 23:45:02 GMT</lastBuildDate>
    <ttl>100</ttl>
    <image>
      <url>http://www.sqlserverandxml.com/image.jpg</url>
      <title>Welcome to XML Workshop</title>
      <link>http://www.sqlserverandxml.com/...central.html</link>
      <width>144</width>
      <height>22</height>
    </image>
    <item>
      <title>XML Workshop I - Generating XML with FOR XML</title>
      <link>http://www.sqlservercentral.com/...2982.asp</link>
      <description>
        A short article that explains how to generate XML output
        with TSQL keyword FOR XML
      </description>
      <pubDate>Wed, 12 Mar 2008 23:45:02 GMT</pubDate>
    </item>
    <item>
      <title>
        XML Workshop II - Reading values from XML variables
      </title>
      <link>http://www.sqlservercentral.com/...2996/</link>
      <description>
        This article explains how to read values from an XML variable
        using XQuery
      </description>
      <pubDate>Wed, 12 Mar 2008 23:45:02 GMT</pubDate>
      <guid isPermaLink="true">http://www.sqlservercentral.com/...2982.asp</guid>
      <guid isPermaLink="true">http://www.sqlservercentral.com/...2996/</guid>
    </item>
  </channel>
</rss>
```

Step 8

We have a problem here. The *isPermalink* attribute should be generated for each item element. At present they appear with the last element only. The problem is with the physical order of the query result. We need to make sure that the *isPermalink* row appears along with the rows of each item. We need to add some sort of ordering logic to get this done. Here is the updated version of the query.

```
SELECT
    Tag,
    Parent,
    [rss!1!version],
    [channel!2!title!element],
    [channel!2!link!element],
    [channel!2!description!element],
    [channel!2!webMaster!element],
    [channel!2!language!element],
    [channel!2!copyright!element],
    [channel!2!lastBuildDate!element],
    [channel!2!ttl!element],
    [image!3!url!element],
    [image!3!title!element],
    [image!3!link!element],
    [image!3!width!element],
    [image!3!height!element],
    [item!4!title!element],
    [item!4!link!element],
    [item!4!description!element],
    [item!4!pubDate!element],
    [guid!5!isPermaLink],
    [guid!5!!element]
FROM (
    SELECT
        1 AS Tag,
        NULL AS Parent,
        '2.0' AS 'rss!1!version',
        NULL AS 'channel!2!title!element',
        NULL AS 'channel!2!link!element',
        NULL AS 'channel!2!description!element',
        NULL AS 'channel!2!webMaster!element',
        NULL AS 'channel!2!language!element',
        NULL AS 'channel!2!copyright!element',
        NULL AS 'channel!2!lastBuildDate!element',
        NULL AS 'channel!2!ttl!element',
        NULL AS 'image!3!url!element',
        NULL AS 'image!3!title!element',
        NULL AS 'image!3!link!element',
        NULL AS 'image!3!width!element',
        NULL AS 'image!3!height!element',
        NULL AS 'item!4!title!element',
        NULL AS 'item!4!link!element',
        NULL AS 'item!4!description!element',
        NULL AS 'item!4!pubDate!element',
        NULL AS 'guid!5!isPermaLink',
        NULL AS 'guid!5!!element',
        CAST(1 AS VARBINARY(4)) AS Sort
    UNION ALL
    SELECT
        2 AS Tag,
        1 AS Parent,
        NULL,
        Title ,
        Link,
        Description,
        WebMaster,
        Language,
        CopyRight,
        LEFT(DATENAME(dw, LastBuildDate),3) + ', ' +
```

```
                    STUFF(CONVERT(nvarchar,LastBuildDate,113),21,4,' GMT'),
        ttl,
        NULL, NULL, NULL, NULL, NULL,
        NULL, NULL, NULL, NULL,
        NULL, NULL,
        CAST(1 AS VARBINARY(4)) + CAST(2 AS VARBINARY(4))
    FROM channel
    UNION ALL
    SELECT
        3 AS Tag,
        2 AS Parent,
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
        ImageUrl,
        ImageTitle,
        ImageLink,
        ImageWidth,
        ImageHeight,
        NULL, NULL, NULL, NULL,
        NULL, NULL,
        CAST(1 AS VARBINARY(4)) + CAST(2 AS VARBINARY(4))
            + CAST(3 AS VARBINARY(4))
    FROM channel
    UNION ALL
    SELECT
        4 AS Tag,
        2 AS Parent,
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
        NULL, NULL, NULL, NULL, NULL,
        title,
        Link,
        Description,
        LEFT(DATENAME(dw, PubDate),3) + ', ' +
                            STUFF(CONVERT(nvarchar,PubDate,113),21,4,' GMT'),
        NULL, NULL,
        CAST(1 AS VARBINARY(4)) + CAST(2 AS VARBINARY(4))
            + CAST(3 AS VARBINARY(4))
            + CAST(ArticleID AS VARBINARY(4))
    FROM Articles
    UNION ALL
    SELECT
        5 AS Tag,
        4 AS Parent,
        NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
        NULL, NULL, NULL, NULL, NULL,
        NULL, NULL, NULL, NULL,
        'true',
        guid,
        CAST(1 AS VARBINARY(4)) + CAST(2 AS VARBINARY(4))
            + CAST(3 AS VARBINARY(4))
            + CAST(ArticleID AS VARBINARY(4))
            + CAST(ArticleID AS VARBINARY(4))
    FROM Articles
) a
ORDER BY SORT
FOR XML EXPLICIT
```

```
<rss version="2.0">
  <channel>
    <title>Welcome to XML Workshop</title>
    <link>http://www.sqlserverandxml.com/...central.html</link>
    <description>
      A collection of short articles on SQL Server and XML
    </description>
    <webMaster>jacob@dotnetquest.com (Jacob Sebastian)</webMaster>
    <language>en-us</language>
    <copyright>Jacob Sebastian. All rights reserved.</copyright>
    <lastBuildDate>Wed, 12 Mar 2008 23:45:02 GMT</lastBuildDate>
    <ttl>100</ttl>
```

```
    <image>
      <url>http://www.sqlserverandxml.com/image.jpg</url>
      <title>Welcome to XML Workshop</title>
      <link>http://www.sqlserverandxml.com/...central.html</link>
      <width>144</width>
      <height>22</height>
    </image>
    <item>
      <title>XML Workshop I - Generating XML with FOR XML</title>
      <link>http://www.sqlservercentral.com/...2982.asp</link>
      <description>
        A short article that explains how to generate XML output
        with TSQL keyword FOR XML
      </description>
      <pubDate>Wed, 12 Mar 2008 23:45:02 GMT</pubDate>
      <guid isPermaLink="true">
        http://www.sqlservercentral.com/...2982.asp
      </guid>
    </item>
    <item>
      <title>XML Workshop II - Reading values from XML variables</title>
      <link>http://www.sqlservercentral.com/...2996/</link>
      <description>
        This article explains how to read values from an XML variable
        using XQuery
      </description>
      <pubDate>Wed, 12 Mar 2008 23:45:02 GMT</pubDate>
      <guid isPermaLink="true">
        http://www.sqlservercentral.com/...2996/
      </guid>
    </item>
  </channel>
</rss>
```

Conclusions

This is yet another session that demonstrates an *XML* shaping example. We have seen different *XML* shaping requirements and their implementation in the previous sessions of the XML Workshop. This session explains the basics of generating an *RSS 2.0* feed using *TSQL* keyword *FOR XML EXPLICIT*.
