XSLT Ingest Example: Appendix D

Appendix D

Start Previous Next

This section is devoted to the recursive template used to create a sequence of partial sums of a sequence of integers. The code for this template is shown in Figure 24.

```
<xsl:template name='cumulativeSum'>

<xsl:param name='vals'/>                                          0

<xsl:param name='seq' />                                          1

<xsl:param name='nxtval' select='0'/>                             2

<xsl:choose>
<xsl:when test='not(empty($vals))'>                               3

<xsl:call-template name='cumulativeSum'>                          4

<xsl:with-param name='vals' select='$vals[position()>1]'/>         a

<xsl:with-param name='seq' select='($seq,$nxtval)'/>              b

<xsl:with-param name='nxtval' select='$nxtval+$vals[1]'/>         c

</xsl:call-template>

</xsl:when>
<xsl:otherwise>

<xsl:sequence select='$seq'/>                                     5

</xsl:otherwise>
</xsl:choose>

</xsl:template>
```

makeURPs.xsl Recursive Template - Figure 24

[F24H0] The **vals**argument should contain a sequence of integers. In recursive calls, **vals** refers to the tail subsequence formed by dropping the first term.

[F24H1] The **seq** argument should start as the empty sequence **()**. In recursive calls, **seq** will refer to the sequence of partial sums.

[F24H2] The **nxtval** argument starts with an initial value of 0 and so need not be present in the first call. In recursive calls it refers to the sum of its last value and the first term in the **vals** sequence.

[F24H3] This test allows the recursion to continue while there are terms left in the sequence **vals** (see [F24H5]).

[F24H4] This is the recursive call where

- [F24H4a] The XPATH expression is the tail of the **vals** sequence with the first term dropped.
- [F24H4b] This parameter uses the sequence constructor **()** to add the value of **nxtval** to the end of the sequence **seq**.
- [F24H4c] The new value of **nxtval** is the old value plus the first term of the **vals** sequence.

[F24H5] This declares the sequence **seq** as the result when **vals** is exhausted (see [F24H3]).

Start Previous Next