

2.4 Document Subsets

Some applications require the ability to create a physical representation for an XML document subset (other than the one generated by default, which can be a proper subset of the document if the comments are omitted). Implementations of XML canonicalization that are based on XPath can provide this functionality with little additional overhead by accepting a node-set as input rather than an octet stream. The processing of an element node E MUST be modified slightly when an XPath node-set is given as input and **the element's parent is** omitted from the node-set. This is necessary because omitted nodes SHALL not break the inheritance rules of inheritable attributes [C14N-Issues] defined in the xml namespace.

[Definition:] **Simple inheritable attributes** are attributes that have a value that requires at most a simple redeclaration. This redeclaration is done by supplying a new value in the child axis. The redeclaration of a simple inheritable attribute A contained in one of E 's ancestors is done by supplying a value to an attribute Ae inside E with the same name. Simple inheritable attributes are `xml:lang` and `xml:space`.

The method for processing the attribute axis of an element E in the node-set is hence enhanced. All element nodes along E 's ancestor axis are examined for the nearest occurrences of simple inheritable attributes in the xml namespace, such as `xml:lang` and `xml:space` (whether or not they are in the node-set). From this list of attributes, any simple inheritable attributes that are already in E 's attribute axis (whether or not they are in the node-set) are removed. Then,

..., 11/5/07 2:52 PM

Deleted: some of the

..., 11/5/07 2:53 PM

Deleted: ancestors are

31 lexicographically merge this attribute list with the nodes of
32 *E*'s attribute axis that are in the node-set. The result of
33 visiting the attribute axis is computed by processing the
34 attribute nodes in this merged attribute list.

35 The `xml:id` attribute is not a simple inheritable attribute and
36 no processing of these attributes is performed.

37 The `xml:base` attribute is not a simple inheritable attribute
38 and requires special processing beyond a simple
39 redeclaration. Hence the processing of *E*'s attribute axis
40 needs to be enhanced further. A "join-URI-References"
41 function is used for `xml:base` fix up. It incorporates `xml:base`
42 attribute values from omitted `xml:base` attributes and
43 updates the `xml:base` attribute value of the element being
44 fixed up, as follows.

45 An `xml:base` fixup is performed on an element *E* as follows.
46 Let *E* be an element in the node set whose ancestor axis
47 contains successive elements *E_n...E₁* (in reverse document
48 order) that are omitted and *E*=*E_{n+1}* is included. (It is
49 important to note that *E_n..E₁* is for contiguously omitted
50 elements, for example only *e₂* in the example in section 3.8.)
51 The fix-up is only performed if at least one of *E₁ ... E_n* had
52 an `xml:base` attribute. In that case let *X₁ ... X_m* be the values
53 of the `xml:base` attributes on *E₁ ... E_{n+1}* (in document order,
54 from outermost to innermost, *m* <= *n+1*). The sequence of
55 values is reduced in reverse document order to a single
56 value by first combining *X_m* with *X_{m-1}*, then the result with
57 *X_{m-2}*, and so on by calling the "join-URI-References"
58 function until the new value for *E*'s `xml:base` attribute
59 remains. The result may also be null or empty (`xml:base=""`)
60 in which case `xml:base` MUST NOT be rendered.

..., 11/5/07 5:23 PM
Deleted:

..., 11/5/07 5:36 PM
Deleted: ,

..., 11/5/07 5:36 PM
Deleted: which

..., 11/5/07 5:42 PM
Deleted: takes any URI (Base) from an ancestor and joins a relative URI of *E* (*R*) (in most cases after the last slash) of the former and then normalizes the result. We describe here a simple method for providing this functionality similar to that found in sections 5.2.1, 5.2.2. and 5.2.4. of [RFC 3986](#) with the following modifications: .
<#>Perform [RFC 3986](#) section 5.2.1. "Pre-parse the Base URI" modified as follows. .
<#>The scheme component is not required in the (... [1])

..., 11/5/07 5:43 PM
Formatted:
Font:Courier, 12 pt

..., 11/5/07 5:35 PM
Deleted: This function may also be called with (... [2])

..., 11/5/07 5:21 PM
Formatted:
Font:Bold, Italic

..., 11/5/07 5:20 PM
Formatted:
Font:Courier

..., 11/5/07 5:39 PM
Deleted: n

..., 11/5/07 5:39 PM
Deleted: has

..., 11/5/07 5:23 PM
Deleted:

..., 11/5/07 5:40 PM
Deleted: described previously

61 Note that this xml:base fixup is only performed if an element
62 with an xml:base attribute is removed. Specifically, it is not
63 performed if the element is present but the attribute is
64 removed.

65 The join-URI-References function takes an xml:base
66 attribute value from an omitted element and combines it with
67 other contiguously omitted values to create a value for an
68 updated xml:base attribute. A simple method for doing this is
69 similar to that found in sections 5.2.1, 5.2.2 and 5.2.4 of RFC
70 3986 with the following modifications:

- 71 • Perform RFC 3986 section 5.2.1. "Pre-parse the Base
72 URI" modified as follows.
 - 73 ○ The scheme component is not required in the base
74 URI (Base). (i.e. Base.scheme may be null)
 - 75 ○ Replace a trailing ".." segment with "../" segment
76 before processing.
- 77
- 78 • 5.2.4. "Remove Dot Segments" is modified as follows:
 - 79 ○ Keep leading "../" segments
 - 80 ○ Replace multiple consecutive "/" characters with a
81 single "/" character.
 - 82 ○ Append a "/" character to a trailing ".." segment
- 83
- 84 • The "Remove Dot Segments" algorithm is modified to
85 ensure that a combination of two xml:base attribute
86 values that include relative path components (i.e., path
87 components that do not begin with a '/' character)
88 results in an attribute value that is a relative path
89 component.
- 90
- 91 • Perform RFC 3986 section 5.2.2. "Transform
92 References" modified as follows to ignore the fragment

..., 11/5/07 5:42 PM

Formatted: Bullets and Numbering

11/6/07 10:54 AM

Formatted:
Bulleted + Level: 1
+ Aligned at: 0.25"
+ Tab after: 0.5" +
Indent at: 0.5"

11/6/07 10:50 AM

Formatted: Bullets and Numbering

11/6/07 10:54 AM

Formatted: Font:16 pt

11/6/07 10:54 AM

Formatted:
Bulleted + Level: 2
+ Aligned at: 0.75"
+ Tab after: 1" +
Indent at: 1"

11/6/07 10:54 AM

Formatted: Font:16 pt

11/6/07 10:54 AM

Formatted: Font:16 pt

11/6/07 10:54 AM

Formatted: Bullets and Numbering

11/19/07 6:12 PM

Formatted:
Bulleted + Level: 1
+ Aligned at: 0.25"
+ Tab after: 0.5" +
Indent at: 0.5",
Tabs: 0.65", Left +
Not at 0.15"

11/6/07 11:06 AM

Formatted: Bullets and Numbering

11/12/07 8:32 AM
Formatted: Indent: Left: 0.75"

11/6/07 11:07 AM
Formatted: Space After: 0 pt, Tabs: 0.15", Left

11/19/07 6:20 PM
Formatted: Font:16 pt

11/19/07 6:20 PM
Formatted: Bulleted + Level: 1 + Aligned at: 0.25" + Tab after: 0.5" + Indent at: 0.5", No widow/orphan control, Don't adjust space between Latin and Asian text, Don't adjust space between Asian text and numbers, Tabs: 0.39", Left + 0.78", Left + 1.17", Left + 1.56", Left + 1.94", Left + 2.33", Left + 2.72", Left + 3.11", Left + 3.5", Left + 3.89", Left + 4.28", Left + 4.67", Left

11/19/07 6:20 PM
Formatted: Font color: Auto

11/19/07 6:17 PM
Formatted: Font:16 pt

11/19/07 6:14 PM
Formatted: Bullets and Numbering

11/19/07 6:17 PM
Formatted: Font color: Auto

11/19/07 6:17 PM
Formatted: Font:16 pt

11/19/07 6:18 PM
Formatted: Indent: Left: 0.25"

11/19/07 6:17 PM
Formatted: Font:16 pt

93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124

part of R

- o After parsing R set R.fragment = null

Then, lexicographically merge this fixed up attribute with the nodes of *E*'s attribute axis that are in the node-set. The result of visiting the attribute axis is computed by processing the attribute nodes in this merged attribute list.

Attributes in the XML namespace other than `xml:base`, `xml:id`, `xml:lang`, and `xml:space` MUST be processed as ordinary attributes.

The following examples illustrate the modification of the "Remove Dot Segments" algorithm:

- "abc/" and "../" should result in ""
- "../" and "../" are combined as "../../" and the result is "../../"
- ".." and ".." are combined as "../../" and the result is "../../"

To illustrate the last example, when the elements b and c are removed from the following sample XML document, the correct result for the xml:base attribute on element d would be "../..x":

```
<a xml:base="foo/bar">  
<b xml:base="..">  
<c xml:base="..">  
<d xml:base="x">  
</d>  
</c>
```

125

~~~~

126

~~~~

127

3.4 Character Modifications and Character

11/19/07 6:17 PM

Deleted: .

128

References

<p>Input Document</p>	<pre><!DOCTYPE doc [<!ATTLIST normNames attr NMTOKENS #I]> <doc> <text>First line&#x0d;&#x32; <value>&#x32;</value> <compute><![CDATA[value>"0" & va <compute expr='value>"0" & amp; & am ?"valid": "error" ' >valid</compute> <norm attr=' & apos; &#x20;&#13; <normNames attr=' A &#x20;&#1 <normId xml:id=' & apos; &#x20;& </doc></pre>
<p>Canonical Form</p>	<pre><doc> <text>First line&#xD; Second line</text> <value>2</value> <compute>value&gt;"0" & amp; & amp; <compute expr="value&quot;0&quot; ?&quot;valid&quot;:&quot;error&quot; <norm attr=" ' &#xD;&#xA;&#x9; <normNames attr="A &#xD;&#xA;&#x9 <normId xml:id=" ' &#xD;&#xA;&#x9; </doc></pre>

11/12/07 12:40 PM

Deleted: <!ATTLI
ST normId id ID
#IMPLIED> .

129

Demonstrates:

130

- Character reference replacement

131

- Attribute value delimiters set to quotation marks (double quotes)

132

133

- Attribute value normalization

134

- CDATA section replacement

135

- Encoding of special characters as character references in attribute values (& amp;, & lt;, & quot;, ,
,)

136

137

- Encoding of special characters as character references in text (&, <, >, )

140

141 **Note:** The last element, `normId`, is well-formed but violates a
 142 validity constraint for attributes of type ID. For testing
 143 canonical XML implementations based on validating
 144 processors, remove the line containing this element from the
 145 input and canonical form. In general, XML consumers should
 146 be discouraged from using this feature of XML.

147 **Note:** Whitespace character references other than ` `
 148 are not affected by attribute value normalization [\[XML\]](#).

149 **Note:** In the canonical form, the value of the attribute named
 150 `attr` in the element `norm` begins with a space, an apostrophe
 151 (single quote), then *four* spaces before the first character
 152 reference.

153 **Note:** The `expr` attribute of the second `compute` element
 154 contains no line breaks.

155

156 3.7 Document Subsets

<p>Input Document</p>	<pre> <!DOCTYPE doc [<!ATTLIST e2 xml:space (default pres]> <doc xmlns="http://www.ietf.org/rfc/rfc3026.xml" > <e1> <e2 xmlns=""> <e3 <u>xml:id</u>="E3"/> </e2> </e1> </doc> </pre> <div data-bbox="1339 1323 1555 1432" style="border: 1px solid red; padding: 2px; font-size: small;"> 11/12/07 12:41 PM Deleted: <!ATTLI ST e3 id ID #IMPLIED> . </div>
<p>Document Subset Expression</p>	<pre> <!-- Evaluate with declaration xmlns (//. //@* //namespace::*) [</pre>

	self::ietf:e1 or (parent::ietf:e1 or count(id("E3") ancestor-or-self::])
Canonical Form	<e1 xmlns="http://www.ietf.org" xmlns: xml:id="E3" xml:space="preserve"></e

157 Demonstrates:

- 158 • Empty default namespace propagation from omitted
- 159 parent element
- 160 • Propagation of attributes in the `xml` namespace in
- 161 document subsets
- 162 • Persistence of omitted namespace declarations in
- 163 descendants

164

165 **Note:** In the document subset expression, the

166 subexpression (`///.` | `//@*` | `//namespace::*`) selects all

167 nodes in the input document, subjecting each to the

168 predicate expression in square brackets. The expression is

169 true for `e1` and its implicit namespace nodes, and it is true if

170 the element identified by `E3` is in the `ancestor-or-self` path

171 of the context node (such that `ancestor-or-self` stays the

172 same size under union with the element identified by `E3`).

173 **Note:** The canonical form contains no line delimiters.

174

175

176 3.8 Document Subsets and XML Attributes

Input Document	<pre><!DOCTYPE doc [<!ATTLIST e2 xml:space (def]> <doc xmlns="http://www.ietf xml:base="something/else"> <e1> <e2 xmlns="" xml:id=" <e3 xml:id="E3" xm</pre>	<p>11/12/07 12:33 PM Deleted: <!ATTLI ST e3 id ID #IMPLIED></p> <p>... 11/5/07 5:12 PM Deleted: http:// www.example.com /</p> <p>... 11/5/07 5:13 PM Deleted: ../</p>
-----------------------	---	---

	<pre> </e2> </e1> </doc> </pre>
Document Subset Expression	<pre> <!-- Evaluate with declaration xmlns (//. //@* //namespace::*) [self::ietf:e1 or (parent::ietf:e1 or count(id("E3") ancestor-or-self:: self::node())] </pre>
Canonical Form	<pre> <e1 xmlns="http://www.ietf.org" xmlns xml:base="something/else"><e3 xmlns= xml:base="something/bar/fo </pre>

... 11/5/07 5:14 PM
Deleted: http://www.example.com /

... 11/5/07 5:15 PM
Deleted: http://www.example.com

11/20/07 10:33 AM
Formatted: Font:20 pt, Bold

11/20/07 10:33 AM
Formatted: Font:Italic

11/6/07 11:04 AM
Formatted: Font:Helvetica, 16 pt, Italic

177 Demonstrates:

- 178 • xml:id not inherited.
- 179 • simple inheritable XML attribute inherited (xml:space)
- 180 • xml:base fixup performed

181
182
183
184
185
186
187
188
189
190
191
192
193
194
195

Appendix A

Use the material in the table in Appendix A following "Some Examples" as given in http://lists.w3.org/Archives/Public/public-xml-core-wg/2007Jun/att-0050/Apendix_20060625.html

Add the following text before table:

The following informative table outlines example results of the modified Remove Dot Segments algorithm described in Section 2.4.