

1 2.4 Document Subsets

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

Frederick Hirsch 11/5/07 2:52 PM

Deleted: some of the

Frederick Hirsch 11/5/07 2:53 PM

Deleted: ancestors are

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

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

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 *En...E1* (in reverse document
48 order) that are omitted and *E=En+1* is included. (It is
49 important to note that *En..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 ... En* had
52 an `xml:base` attribute. In that case let *X1 ... Xm* be the values
53 of the `xml:base` attributes on *E1 ... En+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 *Xm* with *Xm-1*, then the result with
57 *Xm-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.

Frederick Hirsch 11/5/07 5:23 PM

Deleted:

Frederick Hirsch 11/5/07 5:36 PM

Deleted: ,

Frederick Hirsch 11/5/07 5:36 PM

Deleted: which

Frederick Hirsch 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 base URI (Base). (i.e. Base.scheme may be null)
<#>Perform [RFC 3986](#) section 5.2.2. "Transform References" modified as follows to ignore the fragment part of *R*.
<#>After parsing *R* set *R.fragment* = null
<#>5.2.4. "Remove Dot Segments" is modified to keep leading "." segments and to prevent the erroneous creation of an output that looks like a net path. (seg/././pseudo-netpath/seg/file.ext)
<#>several changes as in "Remove Dot Segments" ... (see Appendix)

Frederick Hirsch 11/5/07 5:43 PM

Formatted: Font:Courier, 12 pt

Frederick Hirsch 11/5/07 5:35 PM

Deleted: This function may also be called with the URI to be fixed up (*R*) being null (i.e. when no `xml:base` attribute exists in *E*) or empty "" (`xml:base=""`). The base URI (Base) may also be unknown in which case the Algorithm is performed with Base.scheme = null, Base.authority = null, Base.path = "" and Base.query = nu[... [1]

Frederick Hirsch 11/5/07 5:21 PM

Formatted: Font:Bold, Italic

Frederick Hirsch 11/5/07 5:20 PM

Formatted: Font:Courier

Frederick Hirsch 11/5/07 5:39 PM

Deleted: n

Frederick Hirsch 11/5/07 5:39 PM

Deleted: has

Frederick Hirsch 11/5/07 5:23 PM

Deleted:

Frederick Hirsch 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 • Perform RFC 3986 section 5.2.2. "Transform
85 References" modified as follows to ignore the fragment
86 part of R
 - 87 ○ After parsing R set R.fragment = null
- 88
- 89 • The algorithm is modified to ensure that a combination
90 of two xml:base attribute values that include relative
91 path components (i.e., path components that do not
92 begin with a '/' character) results in an attribute value

Frederick Hirsch 11/5/07 5:42 PM
Formatted: Bullets and Numbering

Frederick Hirsch 11/6/07 10:54 AM
Formatted: Bulleted + Level: 1 + Aligned
at: 0.25" + Tab after: 0.5" + Indent at:
0.5"

Frederick Hirsch 11/6/07 10:50 AM
Formatted: Bullets and Numbering

Frederick Hirsch 11/6/07 10:54 AM
Formatted: Font:16 pt

Frederick Hirsch 11/6/07 10:54 AM
Formatted: Bulleted + Level: 2 + Aligned
at: 0.75" + Tab after: 1" + Indent at: 1"

Frederick Hirsch 11/6/07 10:54 AM
Formatted: Font:16 pt

Frederick Hirsch 11/6/07 10:54 AM
Formatted: Font:16 pt

Frederick Hirsch 11/6/07 10:54 AM
Formatted: Bullets and Numbering

Frederick Hirsch 11/6/07 11:06 AM
Formatted: Bullets and Numbering

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

Frederick Hirsch 11/12/07 8:32 AM
Formatted: Bullets and Numbering

that is a relative path component.

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.

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

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

3.4 Character Modifications and Character References

Input Document	<pre><!DOCTYPE doc [<!ATTLIST normNames attr NMTOKENS #IMPLIED]> <doc> <text>First line&#x0d;&#x10;Second <value>&#x32;</value> <compute><![CDATA[value="0" & va <compute expr='value="0" &am ?"valid":"error"'>valid</compute> <norm attr=' &apos; &#x20;&#13; <normNames attr=' A &#x20;&#1 <normId xml:id=' &apos; &#x20;& </doc></pre>
Canonical Form	<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>

Frederick Hirsch 11/12/07 12:40 PM
Deleted: <!ATTLIST normId id ID #IMPLIED>

Demonstrates:

- 106 • Character reference replacement
- 107 • Attribute value delimiters set to quotation marks (double
- 108 quotes)
- 109 • Attribute value normalization
- 110 • CDATA section replacement
- 111 • Encoding of special characters as character references
- 112 in attribute values (,
,
- 113)
- 114 • Encoding of special characters as character references
- 115 in text (,
,)

116

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

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

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

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

131

132 3.7 Document Subsets

Input Document

```
<!DOCTYPE doc [  
<!ATTLIST e2 xml:space (default|pres  
>]
```

Frederick Hirsch 11/12/07 12:41 PM

Deleted: <!ATTLIST e3 id ID
#IMPLIED>

	<pre><doc xmlns="http://www.ietf.org" xml <e1> <e2 xmlns=""> <e3 xml:id="E3"/> </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::]</pre>
Canonical Form	<pre><e1 xmlns="http://www.ietf.org" xmlns xml:id="E3" xml:space="preserve"></e</pre>

133 Demonstrates:

- 134 • Empty default namespace propagation from omitted
- 135 parent element
- 136 • Propagation of attributes in the `xml` namespace in
- 137 document subsets
- 138 • Persistence of omitted namespace declarations in
- 139 descendants

140

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

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

143 nodes in the input document, subjecting each to the

144 predicate expression in square brackets. The expression is

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

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

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

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

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

150

151

<p>Input Document</p>	<pre><!DOCTYPE doc [<!ATTLIST e2 xml:space (default preserve)]> <doc xmlns="http://www.ietf.org" xml:base="something/else"> <e1> <e2 xmlns="" xml:id="E2"> <e3 xml:id="E3" xmlns="http://www.example.com/"> </e2> </e1> </doc></pre>
<p>Document Subset Expression</p>	<pre><!-- Evaluate with declaration xmlns:ietf="http://www.ietf.org" --> (//. //@* //namespace::*) [self::ietf:e1 or (parent::ietf:e1 or count(id("E3") ancestor-or-self::node()) self::node()]</pre>
<p>Canonical Form</p>	<pre><e1 xmlns="http://www.ietf.org" xmlns:example="http://www.example.com/" xml:base="something/else"><e3 xmlns="http://www.example.com/" xml:base="something/bar/foobaz"></pre>

153 Demonstrates:

- 154 • xml:id not inherited.
- 155 • simple inheritable XML attribute inherited (xml:space)
- 156 • xml:base fixup performed

157

158 Appendix A

159

160 Remove text in Example A up to and including “Some
 161 Examples”, retain table. Add following text before table:

162

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

Frederick Hirsch 11/12/07 12:33 PM
 Deleted: <!ATTLIST e3 id ID #IMPLIED>
 Frederick Hirsch 11/5/07 5:12 PM
 Deleted: http://www.example.com/
 Frederick Hirsch 11/5/07 5:13 PM
 Deleted: ../

Frederick Hirsch 11/5/07 5:14 PM
 Deleted: http://www.example.com/
 Frederick Hirsch 11/5/07 5:15 PM
 Deleted: http://www.example.com

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

