Title: Fixing STRIP WHITESPACE

Author: Jan-Eike Michels
Source: U.S.A.
Status: SQL:2003 TC and SQL:200x WD change proposal
Date: March 8, 2004

Abstract

This paper fixes the behavior of the STRIP WHITESPACE option on XMLParse to restore the original intent. TC changes are needed but not yet included.

References

[DRS-071] Fred Zemke et. al., “Further discussion of Parse, Validate and Serialize”, ISO/IEC JTC1/SC32 WG3:DRS-071 = ANSI INCITS H2-2002-452

[SQL/XML:2003] Jim Melton (ed), “ISO International Standard (IS) Database Language SQL - Part 14: SQL/XML”, ISO/IEC 9075-14:2003

[SQL:2003 TC] Stephen Cannan (ed), to be published

[SQL/XML WD] Jim Melton (ed), “Working Draft (WD) Database Language SQL - Part 14: SQL/XML”, ISO/IEC JTC1/SC32 WG3:SIA-010 = ANSI INCITS H2-2003-427

[ZSH-077r2] Fred Zemke, “Moving to the Infoset data model”, ISO/IEC JTC1/SC32 WG3:ZSH-077r2 = ANSI INCITS H2-2003-052r2

1. Introduction

A discussion in the H2 ad hoc revealed that the current rules of the XMLParse pseudo-function do not behave as intended when the STRIP WHITESPACE is explicitly or implicitly specified.

1.1 Rationale

During the discussion of this topic in the H2 ad hoc it became apparent that the example in [DRS-071] on page 9 expressed the intent of the STRIP WHITESPACE option (which is quite different from the actual behavior as summarized in the table following the example).

The example was:

```
<?xml              version='1.0'           ?>

<well/>

Hello

<a attr=' '>
   <c>    </c>
   Dolly
</a>

You’re looking swell
```

It was decided that there was one error in [DRS-071]: all newlines between the XML declaration and the first element should be stripped, placing <well/> on the same line with the XML declaration. The corrected result of stripping whitespace is then:

```
<?xml version='1.0' ?><well/>

Hello

<a attr=' '><c/>
   Dolly
</a>

You’re looking swell
```

However, the actual rules proposed by [ZSH-077r2] (also present in [SQL/XML:2003]) failed to implement the above behavior. The following table summarizes the different options (STRIP WHITESPACE, PRESERVE WHITESPACE) of XMLParse and the effect they have on stripping white space in [SQL/XML:2003]:

	STRIP WHITESPACE	PRESERVE WHITSPACE
no in-line DTD or XML schema present	no character info item is removed.	no character info item is removed.
in-line DTD or XML schema indicate mixed content		
in-line DTD or XML schema indicate element content and xml:space=’preserve’	no character info item is removed (even if parser sets [element content whitespace] property to ‘true’).	
in-line DTD or XML schema indicate element content and (xml:space attribute is absent or xml:space=’default’)	those character info items are removed whose [element content whitespace] property is ‘true’.	

As can be seen, the STRIP WHITESPACE option behaves the same as the PRESERVE WHITESPACE option except for one case. This makes this option today rather useless and as stated earlier, this behavior was not what was originally intended as expressed in [DRS-071].

The following algorithm summarizes what the actual behavior of the whitespace option in XMLParse should be (with annotations in **red** how this is implemented by either existing GRs or by the newly proposed GR 6) below):

1. Any defaults for the xml:space attribute specified in an internal DTD are applied. If an implementation supports external DTDs, then it may also apply defaults specified there. **(This is covered by the existing GRs 5)c), d), and e))**
2. A top-level element that does not contain an explicit or defaulted xml:space attribute is treated as if it contained “xml:space='default'”. **(For STRIP WHITESPACE, this is covered by the new GRs 6)a)i) below. For PRESERVE WHITESPACE, it does not matter.)**
3. A nested element that does not contain an explicit or defaulted xml:space attribute is treated as if it contained the same xml:space attribute as its parent. **(For STRIP WHITESPACE, this is covered by the new GRs 6)a)iii) below. For PRESERVE WHITESPACE, it does not matter.)**
4. An element with an explicit, defaulted or implicit “xml:space='preserve'” attribute has no whitespace stripping. **(For STRIP WHITESPACE, this is covered by GRs 6)a)i)-iii) since those rules would never make such an**

element potentially whitespace-strippable and consequently GR 6)b) would not remove any white spaces from such an element. For PRESERVE WHITESPACE, it does not matter.)

5. An element with an explicit, defaulted or implicit “xml:space='default'” is handled as follows:

   a) If the PRESERVE WHITESPACE option was specified, then there is no whitespace stripping. **(This is covered by GR 6) being not applicable at all. So no white space stripping takes place.)**

   b) If STRIP WHITESPACE was specified, then any whitespace-only text nodes that are immediate children of the element are dropped. **(The new GR 6)b) below handles this.)**

6. Top-level text nodes (immediate children of the root information item) are handled as follows:

   a) If the PRESERVE WHITESPACE option was specified, then there is no whitespace stripping. **(This is covered by GR 6) being not applicable at all. So, no white space stripping takes place.)**

   b) If STRIP WHITESPACE was specified, then a top-level whitespace-only text node is dropped. **(The new GR 6)c) below handles this.)**

1.2 Proposed Solution

We propose to change GR 6) of Subclause 10.15, Parsing a character string as an XML value, to reflect this algorithm. GR 6) will then read:

6) If *WO* is STRIP WHITESPACE, then:

   a) An XML element information item *EII* contained in *C* is potentially whitespace-strippable if any of the following is true:

      i) *EII* is contained in the [children] property of *XRII* and *EII* does not have an [attributes] property that contains an XML attribute information item for which all of the following are true:

         1) [local name] property is “space”.

         2) [namespace name] property is “http://www.w3.org/XML/1998/namespace”.

         3) [normalized value] property is “preserve”.

      ii) *EII* has an [attributes] property that contains an XML attribute information item for which all of the following are true:

         1) [local name] property is “space”.

2) [namespace name] property is “http://www.w3.org/XML/1998/namespace”.

3) [normalized value] property is “default”.

iii) *EII* is contained in the [children property] of a potentially whitespace-strippable XML element information item and *EII* does not have an [attributes] property that contains an XML attribute information item for which all of the following is true:

1) [local name] property is “space”.

2) [namespace name] property is “http://www.w3.org/XML/1998/namespace”.

3) [normalized value] property is “preserve”.

b) For every potentially whitespace-strippable XML element information item *PWSEII* contained in *C*:

i) Let *N* be the cardinality of the [children] property of PWSEII. Let *XIIi*, 1 (one) <= *i* <= N, be the list of XML information items that is the [children] property of *PWSEII*.

ii) For every *i* between 1 (one) and *N*, if *XIIi* is an XML character information item, then:

1) Let *j* be the least subscript less than or equal to *i* such that for all *q* between *j* and *i*, *XIIq* is an XML character information item.

2) Let *k* be the greatest subscript greater than or equal to *i* such that for all *q* between *i* and *k*, *XIIq* is an XML character information item.

3) If for all *q* between *j* and *k*, *XIIq* is an XML character information item whose [character code] property is a whitespace character, then *XIIq* is marked for removal from *C*.

iii) For every *i* between 1 (one) and *N*, if *XIIi* is marked for removal from *C*, then *XIIi* is removed from *C*.

c) Let *N* be the cardinality of the [children] property of *XRII*. Let *XIIi*, 1 (one) <= *i* <= *N*, be the list of XML information items that is the [children] property of *XRII*.

i) For every *i* between 1 (one) and *N*, if *XIIi* is an XML character information item, then:

1) Let *j* be the least subscript less than or equal to *i* such that for all *q* between *j* and *i*, *XIIq* is an XML character information item.

2) Let *k* be the greatest subscript greater than or equal to *i* such that for all *q* between *i* and *k*, *XIIq* is an XML character information item.

3) If for all *q* between *j* and *k*, *XIIq* is an XML character information item whose [character code] property is a whitespace character, then *XIIq* is marked for removal from *C*.

ii) For every *i* between 1 (one) and *N*, if *XIIi* is marked for removal from *C*, then *XIIi* is removed from *C*.

Consider now the following example:

```
<h><b>database</b> <u>management</u> <i>system</i></h>
```

(there is a space between and <u>, and between </u> and <i>).

Assume the use has not explicitly specified either STRIP WHITESPACE or PRESERVE WHITESPACE, then according to the current rules in [SQL/XML:2003], STRIP WHITESPACE is implicit in this case. An XMLParse with the (implicit) STRIP WHITESPACE option and the new rules would drop the XML character information items corresponding to the spaces and a serialization of this value would produce:

```
<h><b>database</b><u>management</u><i>system</i></h>
```

which, assuming an HTML rendering, would be printed as:

databasemanagement*system*

However, this might not be the user's intent. The user might want the serialization to produce:

database management *system*

which could be achieved by explicitly specifying PRESERVE WHITESPACE.

Therefore, it seems inappropriate to retain the current default (STRIP WHITESPACE) when no option is specified as it could lead to many unwanted/unexpected results. Hence, we propose to require the user to explicitly specify which whitespace handling option he requires; *i.e.*, the user has to specify either STRIP WHITESPACE or PRESERVE WHITESPACE explicitly. An implementation is then free to choose either of those options as a default as a product extension.

Additionally, we create two separate conformance rules, one for STRIP WHITESPACE and one for PRESERVE WHITESPACE.

Since this is solution addresses a bug in [SQL/XML:2003] as well as [SQL/XML WD], we propose equivalent changes to [SQL:2003 TC] and [SQL/XML WD].

1.3 Issues not addressed

The changes in this paper do have some implications on the host language bindings of the XML type and on return values of type XML of external routines, which under the covers use the rules of XMLParse. These implications are not addressed by the present paper, but are left for a follow-on paper.

1.4 Editorial bug fixes in passing

During the research for this paper, the author saw some inconsistency in use of terminology. In some places the term “SQL/XML information item” is used, in others the term “XML information item” (without “SQL/” in front) is used. The latter term seems to be more appropriate, since it is consistent with terms like “XML element information item”, “XML attribute information item”, etc. We therefore instruct the Editor to globally replace all occurrences of “SQL/XML information item” with “XML information item”.

2. Proposal conventions

This proposal uses the following conventions:

1. SMALLCAPS	denote numbered editorial instructions;
~~strikeout~~	denotes existing text to be deleted;
boldface	denotes new text to be inserted;
plain	denotes existing text to be retained,
[Note:...]	brackets enclose italicized notes to the proposal reader
⌊‾‾‾‾⌋	boxes surround “editing tags,” which are part of the document (not instructions to the editor) and may be deleted, inserted, modified or retained, depending on the typeface within the box

3. Proposal for [SQL/XML WD]

[Note to the Editor and Reader: The author of the present paper is aware of at least one other paper (but maybe more) that also touches the same Subclauses and potentially the same rules as the present one. To help the Editor in applying those proposals it might be necessary to produce one paper that takes all those changes into account.]

3.1 Changes to [SQL/XML WD] as a whole.

1. GLOBALLY REPLACE “SQL/XML INFORMATION ITEM” WITH “XML INFORMATION ITEM” (24 TIMES - INCLUDING MAYBE AUTOMATICALLY PRODUCED REFERENCES).

3.2 Changes to Subclause 6.13, <XML parse>.

1. MODIFY THE FORMAT AS SHOWN HERE:

```
<XML parse> ::=
     XMLPARSE <left paren> <document or content> <string
       value expression> ~~[~~ <XML whitespace option> ~~]~~ <right
       paren>
```

2. DELETE SYNTAX RULE 3):

   ~~3) If <XML whitespace option> is not specified, then STRIP WHITESPACE is implicit.~~

3. DELETE CONFORMANCE RULE 3):

   ~~3) Without Feature X062, "XMLParse: explicit WHITESPACE option", in conforming SQL language, <XML parse> shall not contain <XML whitespace option>.~~

4. INSERT TWO NEW CONFORMANCE RULES AS SHOWN HERE:

   x) Without Feature X063, "XMLParse: STRIP WHITESPACE option", in conforming SQL language, <XML parse> shall not contain an <XML whitespace option> that is STRIP WHITESPACE.

   y) Without Feature X064, "XMLParse: PRESERVE WHITESPACE option", in conforming SQL language, <XML parse> shall not contain an <XML whitespace option> that is PRESERVE WHITESPACE.

3.3 Changes to Subclause 10.15, Parsing a character string as an XML value

1. MODIFY THE LEAD-IN OF GENERAL RULE 5)"*V* IS PARSED..." A) AS SHOWN HERE:

   a) Instead of an XML document information item, an XML root information item ***XRII*** is produced, as follows:

2. MODIFY GENERAL RULE 6) AS SHOWN HERE:

   6) If *WO* is STRIP WHITESPACE, then**:**

      a) An XML element information item *EII* contained in *C* is *potentially whitespace-strippable* if any of the following is true:

         i) *EII* is contained in the [children] property of *XRII* and *EII* does not have an [attributes] property that contains an XML attribute information item for which all of the following are true:

            1) [local name] property is "space".

            2) [namespace name] property is "http://www.w3.org/XML/1998/namespace".

            3) [normalized value] property is "preserve".

         ***ii)* *EII* has an [attributes] property that contains an XML attribute information item for which all of the following are true:**

            1) [local name] property is "space".

2) [namespace name] property is "http://www.w3.org/XML/1998/namespace".

3) [normalized value] property is "default".

iii) *EII* is contained in the [children property] of a potentially whitespace-strippable XML element information item and *EII* does not have an [attributes] property that contains an XML attribute information item for which all of the following is true:

1) [local name] property is "space".

2) [namespace name] property is "http://www.w3.org/XML/1998/namespace".

3) [normalized value] property is "preserve".

b) For every potentially whitespace-strippable XML element information item *PWSEII* contained in *C*:

i) Let *N* be the cardinality of the [children] property of *PWSEII*. Let XII_i, 1 (one) <= *i* <= *N*, be the list of XML information items that is the [children] property of *PWSEII*.

ii) For every *i* between 1 (one) and *N*, if XII_i is an XML character information item, then:

1) Let *j* be the least subscript less than or equal to *i* such that for all *q* between *j* and *i*, XII_q is an XML character information item.

2) Let *k* be the greatest subscript greater than or equal to *i* such that for all *q* between *i* and *k*, XII_q is an XML character information item.

NOTE x — Thus the list XII_j, ..., XII_k is the maximal sublist of XII_1, ..., XII_N containing XII_i and consisting entirely of XML character information items. Such a maximal list of XML character information items is commonly called a "text node".

[Note to the Editor: please assign the appropriate number to x.]

3) If for all *q* between *j* and *k*, XII_q is an XML character information item whose [character code] property is a whitespace character, then XII_q is marked for removal from *C*.

iii) For every *i* between 1 (one) and *N*, if XII_i is marked for removal from *C*, then XII_i is removed from *C*.

c) Let *N* be the cardinality of the [children] property of *XRII*. Let XII_i, 1 (one) <= *i* <= *N*, be the list of XML information items that is the [children] property of *XRII*.

i) For every *i* between 1 (one) and *N*, if XII_i is an XML character information item, then:

1) Let *j* be the least subscript less than or equal to *i* such that for all *q* between *j* and *i*, XII_q is an XML character information item.

2) Let *k* be the greatest subscript greater than or equal to *i* such that for all *q* between *i* and *k*, XII_q is an XML character information item.

3) If for all *q* between *j* and *k*, XII_q is an XML character information item whose [character code] property is a whitespace character, then XII_q is marked for removal from *C*.

ii) For every *i* between 1 (one) and *N*, if XII_i is marked for removal from *C*, then XII_i is removed from *C*.

~~any XML character information item *CII* contained in *C* whose [element content whitespace] property is "true" and that is contained in an XML element information item *EII1* that does not have an attribute `xml:space='preserve'` without an intervening XML element information item *EII2* that has an attribute `xml:space='default'` is removed from *C* and from the [children] property of the XML element information item that contains *CII*.~~

3.4 Changes to Annex A, SQL Conformance Summary.

1. AUTOMATICALLY GENERATE THE CHANGES RESULTING FROM THIS PAPER FOR ANNEX A.

3.5 Changes to Annex E, SQL feature taxonomy.

1. MODIFY TABLE 15 - "FEATURE TAXONOMY FOR OPTIONAL FEATURES" AS SHOWN HERE:

	Feature ID	Feature Name
...		
~~34~~	~~X062~~	~~XMLParse: explicit WHITESPACE option~~
...		
a	**X063**	**XMLParse: STRIP WHITESPACE option**
b	**X064**	**XMLParse: PRESERVE WHITESPACE option**

[Note to the Editor: please assign appropriate values to a and b.]

4. Proposal for [SQL:2003 TC]

4.1 TBD.

5. Checklist

Concepts	no
Access Rules	no
Conformance Rules	yes
Lists of SQL-statements by category	no
Table of identifiers used by diagnostics statements	no
Collation coercibility for character strings	no
Closing Possible Problems	no
Any new Possible Problems clearly identified	yes, see Section 1.3, “Issues not addressed”, on page 6 (but this will be addressed in a different paper).
Reserved and non-reserved keywords	no
SQLSTATE tables and Ada package	no
Information and Definition Schemas	no
Implementation-defined and –dependent Annexes	no
Incompatibilities Annex	no
Embedded SQL and host language implications	no (dealt with in a different paper)
Dynamic SQL issues: including descriptor areas	no
CLI issues	no
MED issues	no
SQL/XML issues	yes

- End of paper -