DR 16-0008 — SML: sortState

Status: Further Consideration Required

Subject: SML: sortState

Qualifier: Technical defect

Submitter: Charlie Clark Organization: Ecma/Clark Consulting & Research

Contact Information: [charlie.clark@clark-consulting.eu](file:///E:\Std\SC%2034\WG4\DRs\Log\2014\charlie.clark@clark-consulting.eu)

Submitter’s Defect Number: None

Supporting Document(s): None

Date Circulated by Secretariat: 2016-03-18

Deadline for Response from Editor: 2016-05-18

IS 29500 Reference(s): 29500:2016, Part 1, §18.3.1.92, pp. 1,704–1,705

Related DR(s): None

Nature of the Defect:

This collection preserves the AutoFilter sort state.

[Example: This example shows a sort which is case-sensitive, descending sort. While the range of data to sort is B4:E8, the range to sort by is B4:B8.

<sortState caseSensitive="1" ref="B4:E8"> <sortCondition descending="1" ref="B4:B8"/></sortState>

end example]

This is inaccurate inasmuch as sortState is also a valid child element the worksheet element. However, I'm not sure whether this should be allowed:

The function of the element in the file format is sort of unclear. Sorting is a one-time, in place operation that actually manipulates the data within the worksheet. As such, and unlike the autoFilter mini-database, there seems no point in persisting this information. Doing so without a clear definition of what applications should do complicates the implementation because expectations are unclear. I recently had this case with my library: someone was hoping to mark rows for sorting but nothing was happening with the rows neither in the library nor when the file was subsequently opened.

Unless someone can come up with a good reason, which can then be added to the docs, I suggest that sortState should not be allowed to be a child element of a worksheet.

The difference between filtering and sorting should be more explicit:

§18.3.2, “AutoFilter Settings” states, “The following subclause defines the settings which can be specified as part of an AutoFilter definition. An AutoFilter temporarily hides rows based on a filter criteria, which is applied column by column to a table of data in the worksheet.”

I suggest that this should be extended with something like "Except for sorting operations, an AutoFilter temporarily hides. Sorting operations are performed at in place and are permanent. The criteria are of the most recent sort are preserved as the SortState".

Solution Proposed by the Submitter:

None

Schema Change(s) Needed:

Yes

**Editor’s Response:**

**2016-12-07 Rex Jaeschke:**

MS experts responded: We interpret this question as wanting a way to better understand the runtime behavior of the sortState property. We believe that the spec is correct and clear, and that no change is needed. To directly answer Charlie’s question though, MS uses the sortState property to help repopulate the Sort & Filter options when a user wants to perform a custom sort a second time.

**2016-12-07 Charlie Clark:**

I don't think the answer necessarily applies. I am talking specifically about sortState as a child of the worksheet element. The description, however, specifically refers to sortState as a child of an autoFilter element and Aarti's response seems to mean this as well:

"This collection preserves the AutoFilter sort state." & "MS uses the sortState property to help repopulate the Sort & Filter options when a user wants to perform a custom sort a second time"

What is the purpose of a global sortState element within a worksheet?

**2017-01-25 Aarti Nankani:**

I received the following response from the Excel team: Sorting is not limited to auto-filters.  A custom sort can be performed directly against a range on the worksheet.  The sortState element written as a child of the worksheet is associated with those types of non-autofilter sort operations. Let us know if this is acceptable.

**2017-01-26 Charlie Clark:**

Well, quite obviously not: the documentation makes explicit reference to the AutoFilter:

"18.3.1.92 This collection preserves the AutoFilter sort state."

If there is a difference, then this must be documented including examples.

**2017-02-24 Murata-san:**

The specification makes explicit reference to the AutoFilter in §18.3.1.92, “sortState (Sort State)”: This collection preserves the AutoFilter sort state."

But sortState can occur under table, worksheet, and queryTableRefresh elements as well.

Charlie was talking specifically about sortState as a child of the worksheet element.  He argued that the semantics is not clear.

**2017-03-02 Seattle Meeting:**

After some discussion with MS experts, they agreed to review the latest feedback on this DR. In particular, they will provide information on the use of sortState with table, worksheet, and queryTableRefresh elements.

**2017-06-07 Aarti Nankani:**

Here are the results of MS Experts’ investigation:

* When used within an autofilter, this element defines the last sort operation performed via that autofilter
* When used within a sheet, this element defines the last non-autofilter sort operation the user performed on the sheet
* When used within a table, this element defines the last sort operation performed on that table
  + In the event that a table has an autofilter, and both the table and autofilter specify sortState fields then Excel gives preference to the autofilter
* When used within a queryTableRefresh, this element defines the sort operation to perform on the data returned from the refresh before entering it into the cell table

**2017-06-19/22 Tokyo F2F Meeting:**

Francis will work on this for consideration on Thursday. Here’s his reply:

Following the WG4 face-to-face session on 2017-06-20 I have reviewed DR 16-0008 and the current specification of sml:sortState in prose and in the SML schemas (Strict and Transitional).

Charlie Clark's original DR mentions two issues:

1. The prose in §18.3.1.92 only mentions the use of sml:sortState in one context: within AutoFilter Settings (see §18.3.1.2 autoFilter and §18.3.2 AutoFilter Settings). But the schema specifies sml:sortState as a child element of sml:Worksheet. There is no prose mentioning the use of sml:sortState in this latter context.
2. The function of the element sml:sortState is unclear. It is not clear what purpose is served by storing sml:sortState in a document. It is not clear as to what, if anything, a conformant consuming implementation is supposed to do with this information.

Following a series of exchanges, MS experts provided detailed feedback indicating that sml:sortState can be used in four contexts:

1. sml:autoFilter
2. sml:worksheet
3. sml:table
4. sml:queryTableRefresh

In each case the element except the last, the element sml:sortState defines the last sort operation performed in the given context. In the case of sml:queryTableRefresh, the element sml:sortState defines the sort operation that will be performed when cell table data is refreshed.

I have checked the Strict and Transitional schemas, and in both cases sml:sortState is mentioned as a child element within a complex type CT\_Macrosheet. This complex type is not used in the schemas and is not mentioned in prose. I propose that, as an adjunct to resolving this DR, these complex types should be removed from the schemas (XSD and RNG).

The remaining occurrences of sml:sortState in the schemas correspond to the four contexts described by MS experts; i.e., as a child element of four complex types: CT\_AutoFilter, CT\_Worksheet, CT\_Table and CT\_QueryTableRefresh.

To resolve the two issues raised by Charlie Clark in the DR, I propose the following new text for §18.3.1.92, to replace the first paragraph:

"This collection preserves information used to perform a sort operation on a specific range of data. The applicable range of data shall be interpreted according to the context provided by the parent element: a worksheet (§18.3.1), a filter applied to a worksheet (§18.3.1.2), a table within a worksheet (§18.5), or a query table (§18.12).  
[Note: A consuming implementation may choose to use the preserved information to configure its internal state for the next time a sort operation is performed, unless modified by the user. end note]"

Charlie’s response to this was: “Thanks very much for this. I did get the previous e-mail from Microsoft from you but was frankly largely baffled. It looks to me like Francis has done an excellent job of providing the necessary, though a global sortState for a worksheet looks vestigial as much as anything else.”

If WG4 members agree with Francis’ proposal, then the issue is closed. Schemas need to be changed.

**2017-07-15 Makoto Murata:**

I deleted CT\_Macrosheet from strict/transitional sml.{rnc,xsd}, as proposed by Francis in the DR 16-0008 discussions. This commit is available at [2]. but you first have to create a Visual Studio Team Services Account [1].

[1]

<https://www.visualstudio.com/en-us/docs/setup-admin/team-services/sign-up-for-visual-studio-team-services>

[2]

<https://muratamakoto24.visualstudio.com/_git/OOXMLSchemas/commit/bc0e3cb4de51e2d2f7c0663fcbb8bf421a5e79f8>

**2017-09-21 Rex Jaeschke:**

Here are the final edits as I understand them:

**Part 1: §18.3.1.92, “sortState (Sort State)”, p. 1704**

~~This collection preserves the AutoFilter sort state.~~

This collection preserves information used to perform a sort operation on a specific range of data. The applicable range of data shall be interpreted according to the context provided by the parent element: a worksheet (§18.3.1), a filter applied to a worksheet (§18.3.1.2), a table within a worksheet (§18.5), or a query table (§18.12). [Note: A consuming implementation may choose to use the preserved information to configure its internal state for the next time a sort operation is performed, unless modified by the user. end note]

**Part 1: §A.2, “SpreadsheetML”, p. 3911, lines 2130–2163**

~~<xsd:complexType name="CT\_Macrosheet">~~

~~<xsd:sequence>~~

~~<xsd:element name="sheetPr" type="CT\_SheetPr" minOccurs="0" maxOccurs="1"/>~~

~~…~~

~~<xsd:element name="extLst" minOccurs="0" type="CT\_ExtensionList"/>~~

~~</xsd:sequence>~~

~~</xsd:complexType>~~

**Part 1: §B.2, “SpreadsheetML”, pp. 4221–4222, lines 2294–2319**

~~sml\_CT\_Macrosheet =~~

~~element sheetPr { sml\_CT\_SheetPr }?,~~

~~…~~

~~element extLst { sml\_CT\_ExtensionList }?~~

**Part 4: §A.3, “SpreadsheetML”, p. 974–975, lines 2141–2176**

~~<xsd:complexType name="CT\_Macrosheet">~~

~~<xsd:sequence>~~

~~<xsd:element name="sheetPr" type="CT\_SheetPr" minOccurs="0" maxOccurs="1"/>~~

~~…~~

~~<xsd:element name="extLst" minOccurs="0" type="CT\_ExtensionList"/>~~

~~</xsd:sequence>~~

~~</xsd:complexType>~~

**Part 4: §B.2, “SpreadsheetML”, p. 1318, lines 2301–2328**

~~sml\_CT\_Macrosheet =~~

~~element sheetPr { sml\_CT\_SheetPr }?,~~

~~…~~

~~element extLst { sml\_CT\_ExtensionList }?~~

Changes to Part 1: Y Part 2: N Part 3: N Part 4: Y