# DR 15-0006 — PML: ST\_TLBehaviourAdditiveType is not well-specified

Status: Further Consideration Required

Subject: PML: ST\_TLBehaviourAdditiveType is not well-specified

Qualifier: Technical defect

Submitter: Chris Rae Organization: Microsoft

Contact Information: <chris.rae@microsoft.com>

Submitter’s Defect Number: None

Supporting Document(s): None

Date Circulated by Secretariat: 2015-06-02

Deadline for Response from Editor: 2015-08-02

IS 29500 Reference(s): Part 1:2016, §19.7.28, “ST\_TLBehaviorAdditiveType (Behavior Additive Type)”, p. 2706

Related DR(s): xx

Nature of the Defect:

The specification of ST\_ TLBehaviorAdditiveType is unsatisfactory. It provides the enumerations for animation behaviors, but does not describe how they actually alter animations in practice.

Solution Proposed by the Submitter:

Add normative text detailing what effect this simple type has.

Schema Change(s) Needed: No

**Editor’s Response:**

**2016-02-29/03-02 Barcelona Meeting:**

Darrin proposed the following:

We (MS) need to create an implementer note, since we don’t think we should update the standard itself.

This enum specifies how the animated value is incorporated into the attribute value. PowerPoint only supports the value of Repl and Sum. Other enum values fall back to Base.

* Repl: the animated value replaces the attribute value as absolute value
* Sum: the animated value is added to the attribute value as relative value

Feedback from Caroline:

I believe the semantics of the enum values could usefully be made clearer in the standard (perhaps in the Description boxes or through an explanatory example).  I'm going to assume "we don’t think we should update the standard itself" means they don't want to see a schema modification just because PowerPoint doesn't support a couple of the values.  I also see no reason to modify the schema now, although we could do that later if there are other reasons for an amendment.  What are the intended semantics for Base?

Also, what is "the property" in "how to apply the animation values to the original value for the property"?  I assume it means some original static property?  Would it be an ancestor property or child property?

After some discussion, it was agreed that much of this behavior should be implementation-defined. However, at the very least, the Description column needed to say more than simply restating the left column entries.

**2016-08-08 Darrin House:**

Proposal:

**Part 1: §19.7.28, “ST\_TLBehaviorAdditiveType (Behavior Additive Type)”, p. xxx**

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| Enumeration Value | Description |
| base (Additive Enum ( Base )) | ~~Base~~Specifies that the animation will replace the underlying value and the base value of the attribute |
| mult (Additive Enum ( Multiply )) | ~~Multiply~~Specifies that the animation will multiply the underlying value of the attribute |
| none (None) | ~~None~~The default value, equivalent to repl |
| repl (Additive Enum ( Replace )) | ~~Replace~~Specifies that the animation will replace the underlying value of the attribute |
| sum (Additive Enum ( Sum )) | ~~Sum~~Specifies that the animation will add to the underlying value of the attribute |

**2016-08-08 Francis Cave:**

The proposed text uses the term “attribute” in the description column where I think that it would be clearer and more consistent to use “property”. If you look at the current text of both §19.7.28 and §19.5.22 (where the attribute ‘additive’ is defined for the element type ‘cBhvr’ – the only use of this simple type), it seems clear to me that “property” is the correct term.

**2016-08-08 Makoto Murata:**

Is the description of base correct?

specifies that the animation will replace the underlying value and the base value of the attribute

I assume that base means that the base value be used as is.

**2016-09-26/29 Seoul F2F Meeting:**

We agreed to Francis’ change from “attribute” to “property”.

Regarding Murata-san’s question, “What are the properties involved in animation? Are those spec’d anywhere? What are the differences between original, underlying, and base value?

Darrin will send these questions along to his experts. He did this, and got the following reply:

Underlying value is the value specified in the file format. Base is the underlying value but can be changed at runtime – base starts as the underlying but can then be changed to a new base value.

This is probably explained better in an example:

<rect width="20px" …>

   <animate attributeName="width" from="0px" to="10px" dur="10s"

      additive="sum"/>

</rect>

This is an example of a rectangle that is getting animated from width 20+0px to 20+10px so from 20px to 30px. The underlying value is 20px. The base value is also 20px but if you were to use the additiveType Base, it would replace the base value with whatever you specify.

Is “original” used in the definition? I thought in the table we only used base and underlying.

Rex will take ownership of this issue, and will come up with wording based on this feedback. However, the question, “What are the properties involved in animation?”, was not answered, and Darrin will ask for a response to that.

**2016-10-07 Rex Jaeschke:**

**Part 1: §19.7.28, “ST\_TLBehaviorAdditiveType (Behavior Additive Type)”, p. xxx**

This simple type specifies how to apply the animation values to the underlying~~original~~ value for the property~~.~~(§19.5.1); that is, the value of the property as stored in the file format. At runtime, the base value starts out the same as the underlying value, but the base value can change. [Example: Consider the following:

<rect width="20px" …>

   <animate attributeName="width" from="0px" to="10px" dur="10s"

      additive="sum"/>

</rect>

Here, a rectangle that is to be animated from width 20+0px to 20+10px; that is, from 20px to 30px. The underlying value is 20px. The base value is also 20px, but if base were applied, the base value would change to that specified. end example]

This simple type's contents are a restriction of the W3C XML Schema token datatype.

This simple type is restricted to the values listed in the following table:

|  |  |
| --- | --- |
| Enumeration Value | Description |
| base (Additive Enum ( Base )) | ~~Base~~Specifies that the animation will replace the underlying value and the base value of the property |
| mult (Additive Enum ( Multiply )) | ~~Multiply~~Specifies that the animation will multiply the underlying value of the property |
| none (None) | ~~None~~The default value, equivalent to repl |
| repl (Additive Enum ( Replace )) | ~~Replace~~Specifies that the animation will replace the underlying value of the property |
| sum (Additive Enum ( Sum )) | ~~Sum~~Specifies that the animation will add to the underlying value of the property |

**2016-10-27 Rex Jaeschke:**

From MS experts: It looks like the properties involved in animation are specified in table in 19.5.1 anim (Animate). Copied here: by, calcmode, from, to, valueType.

As such, I’ve added the reference “(§19.5.1)” to the first occurrence of “property” in the proposed solution above.

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