

## W3C QT Specifications - Bug #4236

### Note regarding streamability of fn:filter

2019-06-14 17:16 - Michael Kay

<b>Status:</b>	New	<b>Start date:</b>	2019-06-14
<b>Priority:</b>	Low	<b>Due date:</b>	
<b>Assignee:</b>	Michael Kay	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0:00 hour
<b>Sprint/Milestone:</b>			
<b>Description</b>			
Section §19.8.9.10 discusses streamability of the fn:for-each function but the Note contains a reference to fn:filter, making it unclear whether this is a typo for fn:for-each. There's no separate section for streamability of fn:filter, which might be useful, even if the default rules say it all already.			

#### History

##### #1 - 2019-06-14 17:53 - Michael Kay

Furthermore, the streamability profile for fn:filter is given as fn:filter(N, I) but I think it should be fn:filter(T, I). The distinction doesn't really matter unless we get smarter about streamable higher-order functions.

##### #2 - 2019-06-14 18:20 - Michael Kay

Furthermore, the table in 19.8.9 lists fn:for-each and fn:for-each-pair without a reference to the sections in which they are explained more fully (19.8.9.10, 19.8.9.11).

I don't think that higher-order functions in general are correctly handled simply by applying the general streamability rules to the proforma. The supplied function argument is generally treated as an operand with usage=inspection, and supplying an argument value like function(\$x){\$x/../\*} works just fine for an inspection argument (inline function declarations, as well as user function references, are grounded and motionless). It's when we invoke the function that we get a problem. The rule for fn:filter() should be that the first argument can be consuming/striding provided that the second argument is a call on a function with streamability="inspection".

We should really be using the function annotation mechanism for this. The streamability of a user-declared function should translate into an annotation on the function, and higher order functions should define constraints on the streamability annotation of the actual function supplied.

##### #3 - 2019-06-18 18:00 - Michael Kay

Note that for Saxon, I have now made fn:filter() streamable under particular conditions: the supplied predicate function must be statically known, and must be either (a) a reference to a user-declared stylesheet function with streamability="inspection", or (b) an anonymous inline function that satisfies the rules for streamability="inspection", that is, the function body must be grounded and motionless. The operand usage for the first argument is transmission (so if the first argument is striding and consuming, then so is the result).