

All Fields marked with * are mandatory.

Change Request #:	193
Assigned OGC Document #:	11-177
Name:	*Panagiotis (Peter) A. Vretanos
Organization:	*CubeWerx Inc.
Email:	*pvretano@cubewerx.com
Document Name/Version:	*OpenGIS Web Feature Service 2.0 Interface Standard (also ISO 19142) / 2.0
OGC Project Document:	*09-025r1

If this is a revision of a previous submission and you have a Change Request Number, then check here:

Enter the CR number here:

Enter the Revision Number that you are revising here:

Title:	*Implementation of required valueOf() function will reduce query/filter efficiency.
Source:	*schneider@lat-lon.de
Work item code:	
Category:	* C (Functional modification of feature)

Reason for change:	<p>* during the implementation of 2.0, I ran into a major problem with the wfs:valueOf() function, which may be a specification issue.</p> <p>If I understand correctly, wfs:valueOf() was introduced to have a defined way of addressing property values which can be both encoded inline or by reference (xlink).</p> <p>I would like to point out why the introduction of this function effectively reduces the possibility to map queries to SQL.</p> <p>Please consider the following fragment of a feature collection containing INSPIRE Address features and referenced subfeatures:</p> <pre>[...] <ad:Address [...] gml:id="NL.KAD.BAG.053220000000026"> [...] <ad:component xlink:href="#NL.KAD.AA.1102"/> <ad:component xlink:href="#NL.KAD.PD.1611BT"/> <ad:component xlink:href="#NL.KAD.TN.053230000000062"/> </ad:Address> [...] <ad:ThoroughfareName gml:id="NL.KAD.TN.053230000000062" [...]> [...] <ad:name> <gn:GeographicalName> <gn:language>nld</gn:language> <gn:nativeness>Endonym</gn:nativeness> <gn:nameStatus>Official</gn:nameStatus> <gn:sourceOfName>Het Kadaster, Nederland</gn:sourceOfName> <gn:pronunciation> <gn:PronunciationOfName /> </gn:pronunciation> <gn:spelling> <gn:SpellingOfName> <gn:text>Madame Curiestraat</gn:text> <gn:script>Latn</gn:script> </gn:SpellingOfName> </gn:spelling> </gn:GeographicalName> </ad:name> </ad:ThoroughfareName> [...]</pre> <p>Even before WFS/FE 2.0, PropertyNames with multiple steps were allowed, but there was no hint that the value of an xlink-property should be treated differently than an inlined subfeature. Therefore, we assumed that inline/by-reference shouldn't matter when evaluating an XPath (as</p>
---------------------------	---

it's only a representation aspect anyway). To query Address features that are located in street "Madame Curiestraat", we came up with the following WFS 1.1.0 query:

```
[...]
<wfs:Query typeName="ad:Address">
  <ogc:Filter>
    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>ad:component/ad:ThoroughfareName/ad:name/gn:GeographicalName/gn:spelling/gn:SpellingOfName/gn:text</ogc:Prop
      <ogc:Literal>Madame Curiestraat</ogc:Literal>
    </ogc:PropertyIsEqualTo>
  </ogc:Filter>
</wfs:Query>
[...]
```

This works well for us, and our WFS is able to map the query to an SQL SELECT clause which joins three tables:

```
AD_ADDRESS
AD_THOROUGHFARENAME
AD_ADDRESS_AD_COMPONENT (join table that stores an FK to AD_ADDRESS
and
an FK of *some* other table, e.g AD_ADDRESS_AD_COMPONENT)
```

Please excuse the long prologue. Now to the actual problem:

If I understand correctly, then the above query must be transformed to use the wfs:valueOf() function in order to be a correct WFS 2.0 request:

```
[...]
<wfs:Query typeName="ad:Address">
  <fes:Filter>
    <fes:PropertyIsEqualTo>
      <fes:ValueReference>wfs:valueOf(ad:component)/ad:name/gn:GeographicalName/gn:spelling/gn:SpellingOfName/gn:text</fes:ValueRef
      <fes:Literal>Madame Curiestraat</fes:Literal>
    </fes:PropertyIsEqualTo>
  </fes:Filter>
</wfs:Query>
[...]
```

Unfortunately, the WFS 2.0 XPath is missing the information of the sub feature type (ad:ThoroughfareName):

```
- - WFS 2.0: wfs:valueOf(ad:component)/ad:name/...
- - Before : ad:component/ad:ThoroughfareName/ad:name/...
```

And as the ad:component property may actually contain more than one value feature type (e.g. PostalDescriptor, AddressAreaName, ThoroughfareName), it's not possible any more to transform the query into an unambiguous SQL query. The table to be joined (AD_THOROUGHFARENAME) can not be determined safely, it could also be AD_POSTALDESCRIPTOR or something else.

To summarize: Compared to pre-2.0 XPath expressions, the wfs:valueOf() function removes the actual feature element from the path. And if a feature property allows multiple feature types as a values, this information can be vital in order to generate efficient SQL-WHERE clauses.

Please note that this also has some relation to the (optional) "ImplementsInheritance" conformance class. Implementing "schema-element(..)" is optional, but every 2.0 WFS, must implement "wfs:valueOf()". And "wfs:ValueOf" deals with the ambiguity caused by feature type hierarchies as well (when a substitutable feature type is used in the definition of a feature property).

In order not to loose any filtering possibilities, we would like to stay with the old syntax (at least as an option). Would this be acceptable for a 2.0 implementation? Alternatively, could there be a change to the function signature so that contains the element name of the property value, e.g.

```
wfs:valueOf(ad:component)/ad:ThoroughfareName/ad:name/
or
wfs:valueOf(ad:component,'ad:ThoroughfareName')/ad:name/
```

Summary of change:

* Deprecate valueOf() - i.e., it would still be in the standard in versions 2.x, but client developers would be advised not to use it - and adds a new conformance class based on the old 03-106 proposal.

Consequences if not approved:

Clauses affected: ⓘ	* Primarily 7.3.2 but there are others that reference/use valueOf() that would need to be updated as well.
Additional Documents affected: ⓘ	
Supporting Documentation: ⓘ	03-106
Comments: ⓘ	
Status: ⓘ	Assigned ⓘ
Assigned To: ⓘ	WFS/FES SWG ⓘ
Disposition: ⓘ	Referred ⓘ