All Fields marked with * are mandatory.

Change Request #:	287
Assigned OGC Document #:	13-035
Name:	*Panagiotis (Peter) A. Vretanos
Organization:	*CubeWerx Inc.
Email:	*pvretano@cubewerx.com
Document Name/Version:	*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 Revsion Number that you are revising here: [　　　　]

Title: 🔵	* [WFS 2.0] Allow client to control which features in a join query are presented ir
Source: 🔵	*dandrosevic@galdosinc.com
Work item code: 🔵	
Category: 🔵	* C (Functional modification of feature) ⬍

Reason for change: 🔵	* At the moment a join query always returns tuples of feature types satisfying the join predicate. The problem with this is that in many cases the client only wants one (or a subset) of the feature types being queried. For example, the query "Find all roads that cross Algonquin Park" is a join query but the client only really wants the roads.
Summary of change: 🔵	* I would like to suggest adding an optional element as a query projection clause. That element would specify which features will be included in the response to a query in case of a join query. We briefly discussed it during our last telecom. Currently all features are returned and it requires a client application to filter the response. That response can be quite big. Here is one option: `<xsd:element name="ResultTypeName" type="fes:ResultTypeNameType" substitutionGroup="fes:AbstractProjectionClause"/>` `<xsd:simpleType name="ResultTypeNameType">` `<xsd:union memberTypes="xsd:NCName xsd:QName"/>` `</xsd:simpleType>` NCName is needed for aliases and QName is for features without

```
aliases.

Example 1 (no aliases):
<Query typeNames="myns:Building myns:Road
myns:River">
<ResultTypeName>myns:Road</ResultTypeName>
<PropertyName>myns:numberOfLanes</PropertyName>
<fes:Filter>
....
</fes:Filter>
</Query>

This query would return only myns:Road.

Example 2 (with aliases):
<Query typeNames="myns:Building myns:Road myns:River"
aliases="a b c">
<ResultTypeName>b</ResultTypeName>
<ResultTypeName>c</ResultTypeName>
<PropertyName>b/myns:numberOfLanes</PropertyName>
<fes:Filter>
....
</fes:Filter>
</Query>

This query would return only myns:Road and myns:River.
```

Consequences if not approved: ⊙	1) Clients have to filter the response on the client side to get what they want. 2) Server may transmit more information that required to satisfy the query.

Clauses affected: ⊙	* Clause 11
Additional Documents affected: ⊙	
Supporting Documentation: ⊙	
Comments: ⊙	
Status: ⊙	Assigned ▲▼
Assigned To: ⊙	WFS/FES SWG ▲▼
Disposition: ⊙	Referred ▲▼