

All Fields marked with * are mandatory.

Change Request #:	219
Assigned OGC Document #:	12-070
Name:	*Joshua Vote
Organization:	*CSIRO
Email:	*Josh.Vote@csiro.au
Document Name/Version:	*09-025r1 / 2.0
OGC Project Document:	*09-025
If this is a revision of a previous submission and you have a Change Request Number, then check here: <input type="checkbox"/>	
Enter the CR number here: <input type="text" value="206"/>	
Enter the Revision Number that you are revising here: <input type="text" value="2"/>	
<hr/>	
Title:	* <input type="text" value="WFS Add support for unit of measure conversions"/>
Source:	*WFS-FES.SWG
Work item code:	
Category:	* <input type="text" value="B (Addition of feature)"/>
<hr/>	
Reason for change:	* The current WFS standard has no support for unit of measure conversions. Driving usecases: * Get me everything that matches my filter where the literal has units of my choosing * Get some features and portray their properties in a unit of measure of my choosing
Summary of change:	* * Extensions to PropertyName to include units of measure to enable unit of measure 'projections' * Added a GetDomain operation (modelled from CS/W) to allow client identification of properties with UoM's * Added conformance classes wrapping all of the above.
Consequences if not approved:	
<hr/>	

Clauses affected:	*
	See attached modular specification
Additional Documents affected:	Filter Encoding 2.0 - 09-026r1
Supporting Documentation:	
Comments:	This is a revision of previous change request. It is written using the modular specification principles.
Status:	Assigned
Assigned To:	WFS/FES SWG
Disposition:	Referred

OGC 09-026r1 (Filter Encoding Standard 2.0) - UOM Support

N – Units of Measure support

N.1 Introduction

Numeric properties of features and other items delivered from OGC services are almost always scaled by a unit of measure. While basic services will expect to receive queries and deliver responses in terms of the 'native' units of measure, full support for requests filtered by numeric properties or delivering results containing numbers requires that both requests and responses are explicit about the units of measure or scale.

This clause provides for units of measure support by services that include filter expressions in their requests and responses.

N.2 Service requirements

This sub-clause describes a requirements class for units-of-measure support by services.

Requirements Class	
http://www.opengis.net/spec/filter/2.1/req/uomsupport	
Target type	Service
Dependency	http://www.opengis.net/doc/filter/2.0#clause-7
Dependency	http://www.opengis.net/doc/urn-def/1.3#table-25
Requirement	/req/uomsupport/supports-uom
Requirement	/req/uomsupport/select-uom

The first requirement is that a service supporting units of measure indicates this in the capabilities statement.

Requirement
/req/uomsupport/supports-uom
A service SHALL indicate its support for units of measure in getFeature requests in the FilterCapabilities section of the getCapabilities response using a constraint named 'ImplementsUomSupport'.

EXAMPLE The following XML fragment illustrates a conformance section with 'unit of measure support':

```
<fes:Conformance>
  <fes:Constraint name="ImplementsQuery">
    <ows:NoValues/>
    <ows:DefaultValue>TRUE</ows:DefaultValue>
  </fes:Constraint>
  <fes:Constraint name="ImplementsAdHocQuery">
    <ows:NoValues/>
    <ows:DefaultValue>TRUE</ows:DefaultValue>
  </fes:Constraint>
</fes:Conformance>
```

```

<fes:Constraint name="ImplementsMinStandardFilter">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsStandardFilter">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsMinSpatialFilter">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsSpatialFilter">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsMinTemporalFilter">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsVersionNav">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsSorting">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
</fes:Conformance>
<fes:Constraint name="ImplementsUomSupport">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
</fes:Conformance>

```

Filter expressions include actual values for the various operations using the fes:Literal element. Where a value is a scaled number with a unit of measure, this will be encoded using the relevant GML element, i.e. gml:Measure.

If a service defines itself as supporting units of measure then it is reasonable to expect that clients of that service will want to use filters which contain literals with differing units of measure. As part of supporting units of measure, a service is expected to identify when a literal's unit of measure differs from the property's native unit of measure and attempt to reconcile the differences between the units so an accurate evaluation of the predicate can occur.

Requirement
/req/uomsupport/select-uom
<p>A service receiving a filter expression including a predicate with a valid gml:Measure element in a Literal SHALL compare scaled values respecting the units of measure indicated.</p> <p>If a literal in a filter expression does not include a unit of measure then a service SHALL compare values assuming that they are scaled using a default unit of measure for each property.</p> <p>A service receiving a filter expression including predicate with a unit of measure that it is unable compare against the property referenced by the predicates ValueReference SHALL return an InvalidParameterValue exception.</p>

EXAMPLE – A service with a property <depth> whose values are stored in 'United States feet' is queried with:

```

<PropertyIsGreaterThan>
  <PropertyName>depth</PropertyName>

```

```

    <Literal>
      <gml:Measure uom="m">50</gml:Measure>
    </Literal>
  </PropertyIsGreaterThan>

```

Features whose depth property is greater than 164.042 feet (equivalent to 50 metres) will be selected by the filter.

Conformance Class	
http://www.opengis.net/spec/filter/2.1/conf/uomsupport	
Requirements	http://www.opengis.net/spec/filter/2.1/req/uomsupport
Dependency	http://www.opengis.net/doc/filter/2.0#clause-7.13
Test	/conf/uomsupport/supports-uom

Test	
/conf/uomsupport/supports-uom	
Requirement	/req/uomsupport/supports-uom
Test purpose	To verify that a service supporting units of measure correctly reports this fact in its FilterCapabilities.
Test method	The service FilterCapabilities document SHALL indicate support for units of measure if and only if the service supports it. The service FilterCapabilities document SHALL be schema valid. Pass if the conditions above are all met, Fail otherwise.
Test type	Basic

N.3 Client Requirements

Requirements Class	
http://www.opengis.net/spec/filter/2.1/req/uomfilters	
Target type	Client
Dependency	http://www.opengis.net/doc/filter/2.0#clause-7
Dependency	http://www.opengis.net/spec/filter/2.1/req/uomsupport
Requirement	/req/uomfilters/literal-encoding

A client interacts with the unit of measure functionality in a unit of measure enabled service through the creation of filter expressions that include literals with an explicit unit of measure defined.

Requirement
/req/uomfilters/literal-encoding
A client wishing to encode the units of measure for a literal SHALL encode a gml:Measure element inside of the fes:Literal element.

EXAMPLE: The following XML fragment defines a literal representing one hundred kilograms.

```
<Literal>
  <gml:Measure uom="kg">100</gml:Measure>
</Literal>
```

A service is expected to reconcile any differences between units of measure in the request and units of the data sources before performing a comparison.

Conformance Class	
http://www.opengis.net/spec/filter/2.1/conf/uomfilters	
Requirements	http://www.opengis.net/spec/filter/2.1/req/uomfilters
Dependency	http://www.opengis.net/spec/filter/2.0/conf/...
Test	/conf/uomfilters/literal-encoding

Test	
/conf/uomfilters/literal-encoding	
Requirement	/req/uomfilters/literal-encoding
Test purpose	To verify that a service supporting units of measure accepts literals with a unit of measure.
Test method	The service SHALL accept <Literal> elements encoded with a <gml:Measure>. Pass if <gml:Measure> is accepted as part of a <Literal>, Fail otherwise.
Test type	Basic

09-025r1 (Web Feature Service Standard 2.0) GetDomain Support

P – GetDomain support

P.1 Introduction

NOTE: GetDomain support for WFS has been adapted from OGC-CSW v2.0.2-6 (OGC 07-006r1). In future this functionality might be moved to OWS-Common.

The GetDomain operation is used to obtain runtime information about the unit of measures and/or range of values of one or more properties in a given feature type. The range of values available from a service for a property is typically much smaller than the value space for that property based on its static type definition. For example, a property or request parameter defined as a 16bit positive integer in a database may have a value space of 65535 distinct integers but the actual number of distinct values existing in the database may be much smaller. Numeric properties of features delivered from OGC WFS services are almost always scaled by a unit of measure. While basic WFS services will expect to receive queries and deliver responses in terms of the 'native' units of measure, full support for requests filtered by numeric properties or delivering results containing numbers requires that both requests and responses are explicit about the units of measure or scale.

The runtime information about the range of values of a property is useful for configuring user interfaces with accurate selectable options, or for constructing query predicates that have a higher chance of actually identifying a result set. The runtime information about the unit of measure of a property is also useful in identifying which numerical fields have a defined unit of measure.

In order to generate an accurate filter for a numerical property it is necessary to know the units of measure used for that property. This information needs to be available in advance of a GetFeature request. Furthermore, if a service has a numerical property with a defined unit of measure and the service also supports <http://www.opengis.net/spec/filter/2.1/req/uomsupport>, then properties can be filtered using a variety of equivalent units of measure. Finally, the units of measure of a property to be reported in a GetFeature response can be specified.

It should be noted that the GetDomain operation is a "best-effort" operation. A service tries to generate useful information about the specified properties if it can. However, a service may not know anything about the values of a property beyond the basic type, in which case only a type reference or a type description will be returned.

This clause provides for GetDomain support by WFS services.

P.2 Service requirements

This sub-clause describes a requirements class for services to specify units of measure support via the getdomain operation.

Requirements Class	
http://www.opengis.net/spec/wfs/2.1/req/getdomain	
Target type	Service
Dependency	http://www.opengis.net/doc/WFS/2.0#clause-8
Dependency	http://www.opengis.net/doc/WFS/2.0#clause-9

Dependency	http://www.opengis.net/doc/WFS/2.0#clause-11
Dependency	http://www.opengis.net/spec/filter/2.1/req/uomssupport
Requirement	/req/getdomain/service-supports-getdomain
Requirement	/req/getdomain/request-kvp
Requirement	/req/getdomain/request
Requirement	/req/getdomain/response

The first requirement is that a service supporting the GetDomain request indicate this in the capabilities statement. See also OGC 09-025r1 Clause 8.3.3.

Requirement
/req/getdomain/service-supports-getdomain
A service SHALL indicate its support for the GetDomain operation in the WSDL section of the getCapabilities response.

Example: GetDomain operation expressed as a WSDL operation

```
<wsdl:operation name="wfs.getDomain">
  <wsdl:input message="wfs-req:GetDomainRequest"/>
  <wsdl:output message="wfs-resp:GetDomainResponse"/>
  <wsdl:fault name="ServiceExceptionReport" message="wfs-resp:ServiceExceptionReport"/>
</wsdl:operation>
```

The next two requirements relate to how a request is validly encoded. A service that supports the getDomain operation will provide a suitable response to either an XML- or KVP-encoded request.

Requirement
/req/getdomain/request
Upon receiving a POST request including a <GetDomain> element, for each GetDomainProperty element the service SHALL return a description of the domain indicated by the <PropertyName>, or if <PropertyName> is omitted, of all properties of the specified <TypeName>.

The following XML-Schema fragment defines the XML encoding for the GetDomain operation request:

```
<xsd:element name="GetDomain" type="wfs:GetDomainType"/>
<xsd:complexType name="GetDomainType">
  <xsd:complexContent>
    <xsd:extension base="wfs:BaseRequestType">
      <xsd:sequence>
        <xsd:element ref="wfs:GetDomainProperty" minOccurs="1" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```



```

        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:element name="GetDomainProperty" type="wfs:GetDomainPropertyType"/>

<xsd:complexType name="GetDomainPropertyType">
    <xsd:sequence>
        <xsd:element name="TypeName" type="xsd:string" minOccurs="1"/>
        <xsd:element name="PropertyName" type="xsd:anyURI" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>

```

TypeName parameter

The TypeName refers to the feature type whose properties the client wishes to query. See also clause 7.9.2.4.1 of 09-025r1 OGC WFS version 2.0

PropertyName parameter

The PropertyName parameter is used to specify the name of a property that is defined in the information model for the specified type name. An example of a property name might be an XPath describing the property to be queried.

The property name is an optional element, should it be omitted then the service is expected to return domain information about all properties for the specified typeName.

EXAMPLE – A service receiving the following request shall interpret it as a request for domain information for both the 'length' property of feature type 'Road' and all properties of feature type 'City'.

```

<GetDomain>
  <GetDomainProperty>
    <TypeName>Road</ParameterName>
    <PropertyName>length</ParameterName>
  </GetDomainProperty>
  <GetDomainProperty>
    <TypeName>City</ParameterName>
  </GetDomainProperty>
</GetDomain>

```

Requirement
/req/getdomain/request-kvp
Upon receiving a GET request where REQUEST=GetDomain, a service SHALL return a description of the property domains specified according to the parameters defined in Table 2.

Table 2

URL Component	Description
Common Keywords (REQUEST=GetDomain)	See Table 7 from 09-025r1 OGC WFS version 2.0 for additional parameters that may be used in a KVP-encoded GetDomain request.
typeName	See clause 7.9.2.4.1 of 09-025r1 OGC WFS version 2.0
propertyName	This parameter is optional and SHALL only be used if typeName refers to a single feature type. This parameters SHALL be a comma separated list of property XPaths appropriate for the specified type name. The

	<p>GetDomain response SHALL only include properties included in this list.</p> <p>If this value is omitted, domain information about all properties for the specified type names SHALL be returned.</p>
--	---

Upon receiving a GetDomain request for a set of features and their properties, a service is expected to return domain information about requested properties. The domain information may include ranges of valid values as well as the native unit of measure (if any) for numerical properties.

Requirement
/req/getdomain/response
A service SHALL respond to a GetDomain request with a <GetDomainResponse> element.

NOTE: This information has been adapted from OGC 07-006r1 Section 10.7.5

The following XML-Schema fragment defines the response to a **GetDomain** operation.

```

<xsd:element name="GetDomainResponse" type="wfs:GetDomainResponseType"/>
<xsd:complexType name="GetDomainResponseType">
  <xsd:sequence>
    <xsd:element name="DomainValues" type="wfs:DomainValuesType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DomainValuesType">
  <xsd:sequence>
    <xsd:sequence>
      <xsd:element name="PropertyName" type="xsd:QName"/>
    </xsd:sequence>
    <xsd:choice minOccurs="0">
      <xsd:element name="ListOfValues" type="wfs:ListOfValuesType"/>
      <xsd:element name="ConceptualScheme" type="wfs:ConceptualSchemeType" />
      <xsd:element name="RangeOfValues" type="wfs:RangeOfValuesType" />
      <xsd:element name="UnitOfMeasure" type="gml:UomIdentifier"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="type" type="xsd:QName" use="required" />
</xsd:complexType>
<xsd:complexType name="ListOfValuesType">
  <xsd:sequence>
    <xsd:element name="Value" type="xsd:anyType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ConceptualSchemeType">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:string"/>
    <xsd:element name="Document" type="xsd:anyURI"/>
    <xsd:element name="Authority" type="xsd:anyURI"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RangeOfValuesType">
  <xsd:sequence>
    <xsd:element name="MinValue" type="xsd:anyType"/>
    <xsd:element name="MaxValue" type="xsd:anyType"/>
  </xsd:sequence>
</xsd:complexType>

```

The response is composed of one or more <DomainValues> elements. The domain values may be a list of enumerated values (i.e. <ListOfValues>), one or more ranges of values (i.e. <RangeOfValues>), or a reference to some authoritative vocabulary (i.e. <ConceptualScheme>). An example of an authoritative vocabulary might be a standard list of animal and plant species names.

If the only child element of the <DomainValue> element is the <PropertyName> element, this shall be taken to mean that the service was unable to determine anything about the specified property or parameter.

EXAMPLE: The following is an example response for a feature 'Road' with properties 'length', a number, and 'surface', a string. From the response it is possible to determine that the service treats length as being stored in metres and that it doesn't have any domain information about surface.

```
<GetDomainResponse>
  <DomainValues type="myns:Road">
    <PropertyName>myns:length</PropertyName>
    <RangeOfValues>
      <MinValue>0</MinValue>
      <MaxValue>4294967295</MaxValue>
    </RangeOfValues>
    <UnitOfMeasure>m</UnitOfMeasure>
  </DomainValues>
  <DomainValues type="myns:Road">
    <PropertyName>myns:surface</PropertyName>
  </DomainValues>
</GetDomainResponse>
```

Conformance Class	
http://www.opengis.net/spec/wfs/2.1/conf/getdomain	
Requirements	http://www.opengis.net/spec/wfs/2.1/req/getdomain
Dependency	http://www.opengis.net/spec/wfs/2.0/conf/basic
Dependency	http://www.opengis.net/spec/filter/2.1/conf/uomssupport
Test	/conf/getdomain/service-supports-getdomain
Test	/conf/getdomain/request-kvp
Test	/conf/getdomain/request

Test	
/conf/getdomain/service-supports-getdomain	
Requirement	/req/getdomain/service-supports-getdomain
Test purpose	Verify that a service supporting GetDomain correctly reports this fact in its GetCapabilities document.
Test method	The service GetCapabilities document SHALL indicate support for GetDomain if and only if the service supports it. Pass if the document correctly indicates support, Fail otherwise.
Test type	Basic

Test	
/conf/getdomain/request-kvp	
Requirement	/req/getdomain/request-kvp
Test purpose	Verify that a service correctly supports the KVP encoding of a GetDomain request.
Test method	<p>A GetDomain KVP request with multiple PropertyName's and a single TypeName should be made to the service. The response SHALL be a (possibly empty) list of DomainValues for requested properties and type name. Fail if the response is an exception or a list containing DomainValues for properties or typeNames outside the original request.</p> <p>A GetDomain KVP request with no PropertyName's and a single TypeName should be made to the service. The response SHALL be a (possibly empty) list of DomainValues listed for the specified TypeName. Fail if the response is an exception or a list containing DomainValues for typeNames not specified in the original request.</p> <p>For both tests above, the response GetDomainResponse element must validate according to the schema listed in this document.</p> <p>Pass otherwise.</p>
Test type	Basic

Test	
/conf/getdomain/request	
Requirement	/req/getdomain/request
Test purpose	Verify that a service correctly supports the XML encoding of a GetDomain request.
Test method	<p>A GetDomain XML request with multiple <GetDomainProperty> elements, each with a defined typeName and propertyName, should be made to the service. The response SHALL be a (possibly empty) list of DomainValues for requested properties and type names. Fail if the response is an exception or a list containing DomainValues for properties or typeNames outside the original request.</p> <p>A GetDomain XML request with a single <GetDomainProperty> element with a defined typeName but no propertyName should be made to the service. The response SHALL be a (possibly empty) list of DomainValues for requested type names. Fail if the response is an exception or a list containing DomainValues for typeNames outside the original request.</p> <p>For both tests above, the response GetDomainResponse element must validate according to the schema listed in this document.</p> <p>Pass otherwise.</p>
Test type	Basic

09-025r1 (Web Feature Service Standard 2.0) UoM Projection Support

P – Unit of measure projection support

P.1 Introduction

When a service supports units of measure this allows clients to specify units of measure in the selection clause of a GetFeature request. This clause extends the support to allow units of measure to be specified in the projection clause to guide the encoding of property values in the response. This is accomplished by extending the projection clause with UomProjection elements, each of which defines a preferred unit of measure for a particular property. This gives a client the flexibility to use filter expressions and consume responses in their preferred units of measure.

Unit of measure projection support is limited to properties whose units of measure are explicitly defined by a response from the getDomain request.

P.2 Service requirements

This sub-clause describes a requirements class for services to specify units of measure projection via a concrete specialization of the AbstractProjectionClause.

Requirements Class	
http://www.opengis.net/spec/wfs/2.1/req/uomprojection	
Target type	Server
Dependency	http://www.opengis.net/doc/wfs/2.0#clause-11
Dependency	http://www.opengis.net/spec/filter/2.1/req/uomsupport
Dependency	http://www.opengis.net/spec/wfs/2.1/req/getdomain
Requirement	/req/uomprojection/projection
Requirement	/req/uomprojection/projection-kvp
Requirement	/req/uomprojection/uom-properties

A unit of measure projection request is a list of properties, each with a preferred unit of measure. A service receiving such a request is expected to encode the response using the units of measure indicated for the specified properties.

A service supporting unit of measure projections may received requests through HTTP POST encoded using XML, or through HTTP GET using KVP parameters.

XML Encoding

Requirement
/req/uomprojection/projection

A server SHALL treat an <AbstractProjectionClause> encoded with a <PropertyWithUnits> as a unit of measure projection request.

NOTE: The original use of the projection clause was to choose which optional properties are reported in addition to all mandatory properties for the feature type. In this extension the projection clause is used to indicate the unit of measure for reported properties, which may include both mandatory and optional properties.

```
<xsd:element name="PropertyWithUnits" substitutionGroup="fes:AbstractProjectionClause">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:QName">
        <xsd:attributeGroup ref="wfs:StandardResolveParameters"/>
        <xsd:attribute name="resolvePath" type="xsd:string"/>
        <xsd:attribute name="uom" type="gml:UomIdentifier" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

Comment [vot002 1]: Is this the right type to use?

EXAMPLE: a client of a service supporting units of measure can request that a numerical property 'depth' of a feature type 'Lake' be described using metres. Such a request could be modelled with the following XML:

```
<Query typeNames="myns:Lake">
  <PropertyWithUnits uom="m">myns:depth</PropertyWithUnits>
</Query>
```

KVP Encoding

The KVP encoding described here is an alternative to the method described above. An Ad hoc query expression SHALL include the values in the Table 1 when encoded in the KVP style, or instead use the elements defined by the schema at /req/uomsupport/portrayal.

Requirement
/req/uomprojection/projection-kvp
A service SHALL accept the parameters listed in Table 1 in addition to the KVP parameters defined for Ad Hoc query expressions.

Table 1

URL Component	O/M	Description
UOM_PROPERTY	O*	The value of this parameter SHALL be a comma-separated list of property name XPath expressions representing the properties which shall be portrayed using a particular unit of measure. There must be a 1:1 correspondence between the values in this list and the values of the UOM_ PROJECTION.
UOM_PROJECTION	O*	The value of this parameter shall be a comma-separated list of unit of measure symbols. The values SHOULD conform to the symbology defined by UCUM [1]. Each item in the list SHALL correspond 1:1 with an appropriate property to portray in UOM_PROPERTY.

A service providing support for unit of measure projection is not required to be able to reproject every numerical property, instead it is only expected that the service provide projection capabilities to properties with a unit of measure listed in the response to a getDomain request.

Requirement
/req/uomprojection/uom-properties
A service SHALL only accept unit of measure projection requests for properties listed in the service's getDomain response as having a unit of measure.
A service SHALL return an InvalidParameterValue exception in response to a request for projecting a property's unit of measure if that property is not listed in the service's getDomain response as having a unit of measure.

EXAMPLE: A service with a feature type 'Road' with two properties 'length' and 'maxElevation' returns the following XML in response to a GetDomain request:

```
<GetDomainResponse>
  <DomainValues type="Road" uom="mi_i">
    <PropertyName>length</PropertyName>
    <RangeOfValues>
      <MinValue>0</MinValue>
      <MaxValue>4294967295</MaxValue>
    </RangeOfValues>
  </DomainValues>
  <DomainValues type="Road">
    <PropertyName>maxElevation</PropertyName>
    <RangeOfValues>
      <MinValue>0</MinValue>
      <MaxValue>4294967295</MaxValue>
    </RangeOfValues>
  </DomainValues>
</GetDomainResponse>
```

A client attempt to reproject 'length' to use metres 'm' instead of statute miles 'mi_i' would need to be treated by the service as a valid request. There is no guarantee the service will be able to perform the reprojection but the fact that the GetDomainResponse lists a unit of measure for that property indicates that client may request a reprojection.

Alternatively, a client request to reproject 'maxElevation' to use 'metres' would require the service return an InvalidParameterValue exception as the service has indicated that there is no unit of measure for that property.

Conformance Class	
http://www.opengis.net/spec/wfs/2.1/conf/uomprojection	
Requirements	http://www.opengis.net/spec/wfs/2.1/req/uomprojection
Requirements	http://www.opengis.net/spec/wfs/2.1/req/getdomain
Dependency	http://www.opengis.net/spec/wfs/2.0/conf/basic
Dependency	http://www.opengis.net/spec/filter/2.1/conf/uomsupport
Test	/conf/uomprojection/projection

Test	/conf/uomprojection/uom-properties
-------------	------------------------------------

Test	
/conf/uomprojection/projection	
Requirement	/req/uomprojection/projection
Test purpose	To verify that a service supporting unit of measure projections can correctly project the units of measure used by particular properties.
Test method	Verify that a <AbstractAdhocQuery> request with a single <PropertyWithUnits> element will return a schema valid a set of features. Pass if every feature with the requested property is encoded with the requested unit of measure, Fail otherwise.
Test type	Basic

Test	
/conf/uomprojection/uom-properties	
Requirement	/req/uomprojection/uom-properties
Test purpose	To verify that a service supporting units of measure projections will correctly fail when requested to reproject an invalid property.
Test method	Make a getDomain request for a feature type, identify a property in the GetDomainResponse that does NOT have a unit of measure. Make a subsequent unit of measure projection request for the property identified in the previous step. Pass if an InvalidParameterValue exception is returned, Fail otherwise.
Test type	Basic

Bibliography

[1] The Unified Code for Units of Measure - Version 1.8.2 - <http://unitsofmeasure.org/>