Open Geospatial Consortium Inc.

Date: 2006-02-12

Reference number of this document: OGC 06-027r1

Version: 1.1.0

Category: OpenGIS® IS Corrigendum

Editor: Panagiotis (Peter) A. Vretanos

Corrigendum for the OpenGIS® Web Feature Service (WFS) implementation specification 04-095

Document type:	OpenGIS® IS
Document subtype:	Implementation Specification Corrigendum
Document stage:	Proposed
Document language:	English

Contents

Page

i. Preface

This document is a corrigendum for OGC Document 04-094. Specifically, this document corrects the files referenced in ANNEX A and found in the OGC schema repository.

ii. Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008]. In particular, the word "shall" (not "must") is the verb form used to indicate a requirement to be strictly followed to conform to this specification.

iii. Document contributor contact points

All questions regarding this submission should be directed to the editor or the contributors.

Editor:

Panagiotis (Peter) A. Vretanos (Editor)
CubeWerx Inc.
pvretano {at} cubewerx.com

Contributors:

Name	Organization
Kammersell, Willian	BBNT Solutions LLC
Lansing, Jeff	SYS Technologies Inc.

iv. Revision history

Date	Release	Editor	Primary clauses modified	Description
2006-02-12	1.0.0	Panagiotis (Peter) A. Vretanos	wfs.xsd	The copy of wfs.xsd currently available in the OGC schema repository does not validate and this is causing WFS requests that reference it to not validate as well. This corrigendum lists the changes required to fix wfs.xsd so that it validates.

v. Changes to OGC Specifications

The previously approved OGC™ Specifications do not need changes to accommodate the technical contents of this document.

Foreword

This document provides the details for a corrigendum for the files referenced in ANNEX A of the Web Feature Service implementation specification version 1.1.0 and does not modify that implementation specification.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The OGC shall not be held responsible for identifying any or all such patent rights.

Introduction

This document contains revision notes for 04-094. The OGC document 04-094 was approved by the OGC membership on 2004-12-22. As a result of the RWG process, there were a number of edits and enhancements made to this specification. This document provides the details of those edits, deficiency corrections, and enhancements. It also documents those items that have been deprecated.

Corrigendum for the OpenGIS® WFS implementation specification 04-094

1 Scope

The Web Feature Service implementation specification (04-094) defines a set of operations that allow clients to query, insert, update and delete feature instances for web-accessible feature repositories. ANNEX A of that specification references a set of files in the OGC schema repository (http://schemas.opengis.net/wfs/1.1.0) that define the XML encoding of those operations as well as documents that describe the API using the Web Services Description Language (WSDL). Those files either contain errors that prevent them from validating or do not exist at all.

This document provides the details for a corrigendum that corrects the referenced files that have errors and adds the files that are referenced in ANNEX A but do not currently exist in the OGC schema repository.

2 Normative references

The following is a list of any normative document references that have changed for this Corrigendum. A good example might be that this revision of the specification references the latest OWS Common Specification. For undated references, the latest edition of the normative document referred to applies.

[1] [OGC 04-094], OpenGIS Web Feature Service Implementation Specification version 1.1

3 Corrigendum Description

3.1 Changes to ANNEX A of 04-094

ANNEX A of 04-094 should be changed to read:

> In order to keep this document to a reasonable length, the normative schemas are not included inline but are attached to the archive package that includes this document. Optionally, the schemas can be obtained at http://schemas.opengis.net/wfs.
>
> The files that make up the WFS schemas, WSDL documents and example are:
>
> ```
> 1.1.0/wfs.xsd
> 1.1.0/wsdl/wfs-http-bindings.wsdl
> 1.1.0/wsdl/wfs-kvp-bindings.wsdl
> ```



```
1.1.0/wsdl/wfs-kvp-interfaces.wsdl
1.1.0/wsdl/wfs-kvp.xsd
1.1.0/wsdl/wfs-responses.wsdl
1.1.0/wsdl/wfs-util.xsd
1.1.0/wsdl/wfs-xml-interfaces.wsdl
1.1.0/wsdl/WSDL2Java.bat
1.1.0/examples/WFS_Capabilities_Sample.xml
```

3.2 Changes to the XML Schema *wfs.xsd*

ANNEX A contains a copy of the *wfs.xsd* XML Schema document that was distributed with 04-094. The schema in ANNEX A has been annotated with line numbers, which are used in Table 1 to indicate where the specific changes listed should be made.

The columns in Table 1 contain the following information:

Line #: The line number where a change has been made is show here. The line number refers to the line numbers in the annotated schema in ANNEX A.

O: The letter contained in the column denotes the operation performed at that indicated line number; **A**=add, **C**=change, **D**=delete.

Original Text: The XML Schema fragment that currently exists in the schema in ANNEX A is shown here. If an addition is being made then this column can be empty.

Replacement Text: The XML Schema fragment that should replace the existing fragment is shown here. If a deletion is being made then this column can be empty

Reason: A brief explanation about the change.

Validity of the resulting schema has been checked using the following validating XML parsers: XSV Web Form version 2.10-1, XML Spy version 2006 sp1 and Xerces version 2.7.1 (Java).

Table 1 – Changes to wfs.xsd

Line #	O	Original Text	Replacement Text	Reason
1795	C	`<xsd:complexType name="InsertResultType">`	`<xsd:complexType name="InsertResultsType">`	The element is named "InsertResults" so the type should be renamed to match.
1692	C	`Type="wfs:InsertResultType">`	`type="wfs:InsertResultsType">`	The element is named "InsertResults" so the type should be renamed to match.

Line #	O	Original Text	Replacement Test	Reason
1342	C	<xsd:choice> <xsd:element ref="gml:_FeatureCollection"/> <xsd:sequence> <xsd:element ref="gml:_Feature" maxOccurs="unbounded"/> </xsd:sequence> </xsd:choice>	<xsd:sequence> <xsd:element ref="gml:_Feature" maxOccurs="unbounded"/> </xsd:sequence>	_FeatureCollection and _Feature are already substitutable for each other so you don't need to reference them twice.
1253	C	<xsd:extension base="ows:GetCapabilitiesType">	<xsd:extension base="wfs:GetBaseRequestType">	Derived from wrong base type.
695	A		<xsd:element ref="XlinkPropertyName"/>	See line 88.
344	C	<xsd:restriction base="xsd:NMTOKEN">	<xsd:restriction base="xsd:string">	MIME types, such as text/xml, are not valid NMTOKEN types.
334	C	<xsd:restriction base="xsd:NMTOKEN">	<xsd:restriction base="xsd:string">	MIME types, such as text/xml, are not valid NMTOKEN types.
304	C	<xsd:enumeration value="Unsert"/>	<xsd:enumeration value="Update"/>	
164	D	substitutionGroup="ows:Capabilities"		This element does not exist so you cannot substitute for it.
129	A		</xsd:extension> </xsd:simpleContent>	See line 99
99	A		<xsd:simpleContent> <xsd:extension base="xsd:string">	See line 87
88	D	substitutionGroup="wfs:PropertyName">		a choice group can be used to encode either PropertyName or XlinkPropertyName
87	C	<xsd:element name="XlinkPropertyName" type="xsd:QName">	<xsd:element name="XlinkPropertyName>	XlinkPropertyName can be an XPath expression so QName is not the correct type.
86	C	<xsd:element name="ProperyName" type="xsd:QName"/>	<xsd:element name="PropertyName" type="xsd:string"> <xsd:annotation> <xsd:documentation> The Property element … n the first place </xsd:documentation> </xsd:annotation> </xsd:element>	Fix spelling error ProperyName should be PropertyName. PropertyName can be an XPath expression so it must be 'string' not Qname Add docco from spec
20	C	owsGetCapabilities.xsd	owsAll.xsd	Fixes unresolved reference to ows:ExceptionReport

ANNEX A
Annotated XML Schema wfs.xsd distributed with 04-094

```
 1 <?xml version="1.0"?>
 2 <xsd:schema
 3    targetNamespace="http://www.opengis.net/wfs"
 4    xmlns:wfs="http://www.opengis.net/wfs"
 5    xmlns:ogc="http://www.opengis.net/ogc"
 6    xmlns:ows="http://www.opengis.net/ows"
 7    xmlns:gml="http://www.opengis.net/gml"
 8    xmlns:xlink="http://www.w3.org/1999/xlink"
 9    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
10    elementFormDefault="qualified" version="1.1.0">
11
12    <!-- ============================================================
13         Includes and Imports
14         ============================================================ -->
15    <xsd:import namespace="http://www.opengis.net/gml"
16                schemaLocation="../../gml/3.1.1/base/gml.xsd"/>
17    <xsd:import namespace="http://www.opengis.net/ogc"
18                schemaLocation="../../filter/1.1.0/filter.xsd"/>
19    <xsd:import namespace="http://www.opengis.net/ows"
20                schemaLocation="../../ows/1.0.0/owsGetCapabilities.xsd"/>
21
22    <!-- ================================================================== -->
23    <!-- =   BASE REQUEST TYPE                                           = -->
24    <!-- ================================================================== -->
25    <xsd:complexType name="BaseRequestType" abstract="true">
26       <xsd:annotation>
27          <xsd:documentation>
28             XML encoded WFS operation request base, for all operations
29             except GetCapabilities.
30          </xsd:documentation>
31       </xsd:annotation>
32       <xsd:attribute name="service" type="ows:ServiceType"
33                      use="optional" default="WFS">
34         <xsd:annotation>
35            <xsd:documentation>
36               The service attribute is included to support service
37               endpoints that implement more than one OGC service.
38               For example, a single CGI that implements WMS, WFS
39               and WCS services.
40               The endpoint can inspect the value of this attribute
41               to figure out which service should process the request.
42               The value WFS indicates that a web feature service should
43               process the request.
44               This parameter is somewhat redundant in the XML encoding
45               since the request namespace can be used to determine
46               which service should process any give request.  For example,
47               wfs:GetCapabilities and easily be distinguished from
48               wcs:GetCapabilities using the namespaces.
49            </xsd:documentation>
50         </xsd:annotation>
51       </xsd:attribute>
52       <xsd:attribute name="version" type="xsd:string"
53                      use="optional" default="1.1.0">
54         <xsd:annotation>
55            <xsd:documentation>
56                The version attribute is used to indicate the version of the
57                WFS specification that a request conforms to.  All requests in
58                this schema conform to V1.1 of the WFS specification.
59                For WFS implementations that support more than one version of
60                a WFS sepcification ... if the version attribute is not
61                specified then the service should assume that the request
62                conforms to greatest available specification version.
63            </xsd:documentation>
64         </xsd:annotation>
65       </xsd:attribute>
```

```
      <xsd:attribute name="handle"
                     type="xsd:string" use="optional">
         <xsd:annotation>
            <xsd:documentation>
               The handle attribute allows a client application
               to assign a client-generated request identifier
               to a WFS request.  The handle is included to
               facilitate error reporting.  A WFS may report the
               handle in an exception report to identify the
               offending request or action.  If the handle is not
               present, then the WFS may employ other means to
               localize the error (e.g. line numbers).
            </xsd:documentation>
         </xsd:annotation>
      </xsd:attribute>
   </xsd:complexType>

   <!-- ============================================================= -->
   <!-- = PROPERTY NAME ELEMENT                                       = -->
   <!-- ============================================================= -->
   <xsd:element name="ProperyName" type="xsd:QName"/>
   <xsd:element name="XlinkPropertyName" type="xsd:QName"
                substitutionGroup="wfs:PropertyName">
      <xsd:annotation>
         <xsd:documentation>
            This element may be used in place of an wfs:PropertyName element
            in a wfs:Query element in a wfs:GetFeature element to selectively
            request the traversal of nested XLinks in the returned element for
            the named property. This element may not be used in other requests
            -- GetFeatureWithLock, LockFeature, Insert, Update, Delete -- in
            this version of the WFS specification.
         </xsd:documentation>
      </xsd:annotation>
      <xsd:complexType>
         <xsd:attribute name="traverseXlinkDepth"
                        type="xsd:string" use="required">
            <xsd:annotation>
               <xsd:documentation>
                  This attribute indicates the depth to which nested property
                  XLink linking element locator attribute (href) XLinks are
                  traversed and resolved if possible.  A value of "1" indicates
                  that one linking element locator attribute (href) Xlink
                  will be traversed and the referenced element returned if
                  possible, but nested property XLink linking element locator
                  attribute (href) XLinks in the returned element are not
                  traversed.  A value of  "*" indicates that all nested property
                  XLink linking element locator attribute (href) XLinks will be
                  traversed and the referenced elements returned if possible.
                  The range of valid values for this attribute consists of
                  positive integers plus "*".
               </xsd:documentation>
            </xsd:annotation>
         </xsd:attribute>
         <xsd:attribute name="traverseXlinkExpiry"
                        type="xsd:positiveInteger"
                        use="optional">
            <xsd:annotation>
               <xsd:documentation>
                  The traverseXlinkExpiry attribute value is specified in
                  minutes It indicates how long a Web Feature Service should
                  wait to receive a response to a nested GetGmlObject request.
               </xsd:documentation>
            </xsd:annotation>
         </xsd:attribute>
      </xsd:complexType>
   </xsd:element>

   <!-- ============================================================= -->
   <!-- =   GETCAPABILITIES Request and Response                      = -->
   <!-- ============================================================= -->
   <!-- REQUEST -->
   <xsd:element name="GetCapabilities" type="wfs:GetCapabilitiesType"/>
```



```
138   <xsd:complexType name="GetCapabilitiesType">
139     <xsd:annotation>
140       <xsd:documentation>
141          Request to a WFS to perform the GetCapabilities operation.
142          This operation allows a client to retrieve a Capabilities
143          XML document providing metadata for the specific WFS server.
144
145          The GetCapapbilities element is used to request that a Web Feature
146          Service generate an XML document describing the organization
147          providing the service, the WFS operations that the service
148          supports, a list of feature types that the service can operate
149          on and list of filtering capabilities that the service support.
150          Such an XML document is called a capabilities document.
151       </xsd:documentation>
152     </xsd:annotation>
153     <xsd:complexContent>
154       <xsd:extension base="ows:GetCapabilitiesType">
155         <xsd:attribute name="service" type="ows:ServiceType"
156                        use="optional" default="WFS"/>
157
158       </xsd:extension>
159     </xsd:complexContent>
160   </xsd:complexType>
161   <!-- RESPONSE -->
162   <xsd:element name="WFS_Capabilities"
163                type="wfs:WFS_CapabilitiesType"
164                substitutionGroup="ows:Capabilities"/>
165   <xsd:complexType name="WFS_CapabilitiesType">
166      <xsd:annotation>
167         <xsd:documentation>
168            XML encoded WFS GetCapabilities operation response. This
169            document provides clients with service metadata about a
170            specific service instance, including metadata about the
171            tightly-coupled data served. If the server does not implement
172            the updateSequence parameter, the server shall always return
173            the complete Capabilities document, without the updateSequence
174            parameter. When the server implements the updateSequence
175            parameter and the GetCapabilities operation request included
176            the updateSequence parameter with the current value, the server
177            shall return this element with only the "version" and
178            "updateSequence" attributes. Otherwise, all optional elements
179            shall be included or not depending on the actual value of the
180            Contents parameter in the GetCapabilities operation request.
181         </xsd:documentation>
182      </xsd:annotation>
183      <xsd:complexContent>
184         <xsd:extension base="ows:CapabilitiesBaseType">
185            <xsd:sequence>
186               <xsd:element ref="wfs:FeatureTypeList" minOccurs="0"/>
187               <xsd:element ref="wfs:ServesGMLObjectTypeList" minOccurs="0"/>
188               <xsd:element ref="wfs:SupportsGMLObjectTypeList" minOccurs="0"/>
189               <xsd:element ref="ogc:Filter_Capabilities"/>
190            </xsd:sequence>
191         </xsd:extension>
192      </xsd:complexContent>
193   </xsd:complexType>
194   <xsd:element name="FeatureTypeList" type="wfs:FeatureTypeListType"/>
195   <xsd:complexType name="FeatureTypeListType">
196      <xsd:annotation>
197         <xsd:documentation>
198            A list of feature types available from  this server.
199         </xsd:documentation>
200      </xsd:annotation>
201      <xsd:sequence>
202         <xsd:element name="Operations"
203                      type="wfs:OperationsType"
204                      minOccurs="0"/>
205         <xsd:element name="FeatureType"
206                      type="wfs:FeatureTypeType"
207                      maxOccurs="unbounded"/>
208      </xsd:sequence>
209   </xsd:complexType>
```

```
<xsd:complexType name="FeatureTypeType">
   <xsd:annotation>
      <xsd:documentation>
         An element of this type that describes a feature in an application
         namespace shall have an xml xmlns specifier, e.g.
         xmlns:bo="http://www.BlueOx.org/BlueOx"
      </xsd:documentation>
   </xsd:annotation>
   <xsd:sequence>
      <xsd:element name="Name" type="xsd:QName">
         <xsd:annotation>
            <xsd:documentation>
               Name of this feature type, including any namespace prefix
            </xsd:documentation>
         </xsd:annotation>
      </xsd:element>
      <xsd:element name="Title" type="xsd:string">
         <xsd:annotation>
            <xsd:documentation>
               Title of this feature type, normally used for display
               to a human.
            </xsd:documentation>
         </xsd:annotation>
      </xsd:element>
      <xsd:element name="Abstract" type="xsd:string" minOccurs="0">
         <xsd:annotation>
            <xsd:documentation>
               Brief narrative description of this feature type, normally
               used for display to a human.
            </xsd:documentation>
         </xsd:annotation>
      </xsd:element>
      <xsd:element ref="ows:Keywords" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:choice>
         <xsd:sequence>
            <xsd:element name="DefaultSRS"
                         type="xsd:anyURI">
               <xsd:annotation>
                  <xsd:documentation>
                     The DefaultSRS element indicated which spatial
                     reference system shall be used by a WFS to
                     express the state of a spatial feature if not
                     otherwise explicitly identified within a query
                     or transaction request.  The SRS may be indicated
                     using either the EPSG form (EPSG:posc code) or
                     the URL form defined in subclause 4.3.2 of
                     refernce[2].
                  </xsd:documentation>
               </xsd:annotation>
            </xsd:element>
            <xsd:element name="OtherSRS"
                         type="xsd:anyURI"
                         minOccurs="0" maxOccurs="unbounded">
               <xsd:annotation>
                  <xsd:documentation>
                     The OtherSRS element is used to indicate other
                     supported SRSs within query and transaction
                     operations.  A supported SRS means that the
                     WFS supports the transformation of spatial
                     properties between the OtherSRS and the internal
                     storage SRS.  The effects of such transformations
                     must be considered when determining and declaring
                     the guaranteed data accuracy.
                  </xsd:documentation>
               </xsd:annotation>
            </xsd:element>
         </xsd:sequence>
         <xsd:element name="NoSRS">
           <xsd:complexType/>
         </xsd:element>
      </xsd:choice>
      <xsd:element name="Operations"
```

```
282                     type="wfs:OperationsType"
283                     minOccurs="0"/>
284         <xsd:element name="OutputFormats"
285                     type="wfs:OutputFormatListType"
286                     minOccurs="0"/>
287         <xsd:element ref="ows:WGS84BoundingBox"
288                     minOccurs="1" maxOccurs="unbounded"/>
289         <xsd:element name="MetadataURL"
290                     type="wfs:MetadataURLType"
291                     minOccurs="0" maxOccurs="unbounded"/>
292      </xsd:sequence>
293   </xsd:complexType>
294   <xsd:complexType name="OperationsType">
295      <xsd:sequence>
296         <xsd:element name="Operation"
297                     type="wfs:OperationType"
298                     maxOccurs="unbounded"/>
299      </xsd:sequence>
300   </xsd:complexType>
301   <xsd:simpleType name="OperationType">
302      <xsd:restriction base="xsd:string">
303         <xsd:enumeration value="Insert"/>
304         <xsd:enumeration value="Unsert"/>
305         <xsd:enumeration value="Delete"/>
306         <xsd:enumeration value="Query"/>
307         <xsd:enumeration value="Lock"/>
308         <xsd:enumeration value="GetGmlObject"/>
309      </xsd:restriction>
310   </xsd:simpleType>
311   <xsd:complexType name="OutputFormatListType">
312      <xsd:sequence maxOccurs="unbounded">
313         <xsd:element name="Format" type="xsd:string"/>
314      </xsd:sequence>
315   </xsd:complexType>
316   <xsd:complexType name="MetadataURLType">
317      <xsd:annotation>
318         <xsd:documentation>
319            A Web Feature Server MAY use zero or more MetadataURL
320            elements to offer detailed, standardized metadata about
321            the data underneath a particular feature type.  The type
322            attribute indicates the standard to which the metadata
323            complies; the format attribute indicates how the metadata is
324            structured.  Two types are defined at present:
325            'TC211' or 'ISO19115' = ISO TC211 19115;
326            'FGDC'                = FGDC CSDGM.
327            'ISO19139'            = ISO 19139
328         </xsd:documentation>
329      </xsd:annotation>
330      <xsd:simpleContent>
331         <xsd:extension base="xsd:string">
332            <xsd:attribute name="type" use="required">
333               <xsd:simpleType>
334                  <xsd:restriction base="xsd:NMTOKEN">
335                     <xsd:enumeration value="TC211"/>
336                     <xsd:enumeration value="FGDC"/>
337                     <xsd:enumeration value="19115"/>
338                     <xsd:enumeration value="19139"/>
339                  </xsd:restriction>
340               </xsd:simpleType>
341            </xsd:attribute>
342            <xsd:attribute name="format" use="required">
343               <xsd:simpleType>
344                  <xsd:restriction base="xsd:NMTOKEN">
345                     <xsd:enumeration value="text/xml"/>
346                     <xsd:enumeration value="text/html"/>
347                     <xsd:enumeration value="text/sgml"/>
348                     <xsd:enumeration value="text/plain"/>
349                  </xsd:restriction>
350               </xsd:simpleType>
351            </xsd:attribute>
352         </xsd:extension>
353      </xsd:simpleContent>
```

```
354    </xsd:complexType>
355    <xsd:element name="ServesGMLObjectTypeList"
356                 type="wfs:GMLObjectTypeListType">
357       <xsd:annotation>
358          <xsd:documentation>
359             List of GML Object types available for GetGmlObject requests
360          </xsd:documentation>
361       </xsd:annotation>
362    </xsd:element>
363    <xsd:element name="SupportsGMLObjectTypeList"
364                 type="wfs:GMLObjectTypeListType">
365       <xsd:annotation>
366          <xsd:documentation>
367             List of GML Object types that WFS is capable of serving, either
368             directly, or as validly derived types defined in a GML application
369             schema.
370          </xsd:documentation>
371       </xsd:annotation>
372    </xsd:element>
373    <xsd:complexType name="GMLObjectTypeListType">
374       <xsd:sequence>
375          <xsd:element name="GMLObjectType" type="wfs:GMLObjectTypeType"
376             maxOccurs="unbounded">
377             <xsd:annotation>
378                <xsd:documentation>
379                   Name of this GML object type, including any namespace prefix
380                </xsd:documentation>
381             </xsd:annotation>
382          </xsd:element>
383       </xsd:sequence>
384    </xsd:complexType>
385    <xsd:complexType name="GMLObjectTypeType">
386       <xsd:annotation>
387          <xsd:documentation>
388             An element of this type that describes a GML object in an
389             application namespace shall have an xml xmlns specifier,
390             e.g. xmlns:bo="http://www.BlueOx.org/BlueOx"
391          </xsd:documentation>
392       </xsd:annotation>
393       <xsd:sequence>
394          <xsd:element name="Name" type="xsd:QName">
395             <xsd:annotation>
396                <xsd:documentation>
397                   Name of this GML Object type, including any namespace prefix.
398                </xsd:documentation>
399             </xsd:annotation>
400          </xsd:element>
401          <xsd:element name="Title" type="xsd:string" minOccurs="0">
402             <xsd:annotation>
403                <xsd:documentation>
404                   Title of this GML Object type, normally used for display
405                   to a human.
406                </xsd:documentation>
407             </xsd:annotation>
408          </xsd:element>
409          <xsd:element name="Abstract" type="xsd:string" minOccurs="0">
410             <xsd:annotation>
411                <xsd:documentation>
412                   Brief narrative description of this GML Object type, normally
413                   used for display to a human.
414                </xsd:documentation>
415             </xsd:annotation>
416          </xsd:element>
417          <xsd:element ref="ows:Keywords"
418                       minOccurs="0" maxOccurs="unbounded"/>
419          <xsd:element name="OutputFormats"
420                       type="wfs:OutputFormatListType" minOccurs="0"/>
421       </xsd:sequence>
422    </xsd:complexType>
423
424    <!-- ================================================================== -->
425    <!-- =   DESCRIBEFEATURETYPE Request and Response                     = -->
```



```
426   <!-- ==================================================================== -->
427   <!-- REQUEST -->
428   <xsd:element name="DescribeFeatureType" type="wfs:DescribeFeatureTypeType">
429      <xsd:annotation>
430         <xsd:documentation>
431            The DescribeFeatureType element is used to request that a Web
432            Feature Service generate a document describing one or more
433            feature types.
434         </xsd:documentation>
435      </xsd:annotation>
436   </xsd:element>
437   <xsd:complexType name="DescribeFeatureTypeType">
438      <xsd:annotation>
439         <xsd:documentation>
440            The DescribeFeatureType operation allows a client application
441            to request that a Web Feature Service describe one or more
442            feature types.   A Web Feature Service must be able to generate
443            feature descriptions as valid GML3 application schemas.
444
445            The schemas generated by the DescribeFeatureType operation can
446            be used by a client application to validate the output.
447
448            Feature instances within the WFS interface must be specified
449            using GML3.  The schema of feature instances specified within
450            the WFS interface must validate against the feature schemas
451            generated by the DescribeFeatureType request.
452         </xsd:documentation>
453      </xsd:annotation>
454      <xsd:complexContent>
455         <xsd:extension base="wfs:BaseRequestType">
456            <xsd:sequence>
457               <xsd:element name="TypeName" type="xsd:QName"
458                            minOccurs="0" maxOccurs="unbounded">
459                  <xsd:annotation>
460                     <xsd:documentation>
461                        The TypeName element is used to enumerate the
462                        feature types to be described.  If no TypeName
463                        elements are specified then all features should
464                        be described.  The name must be a valid type
465                        that belongs to the feature content as defined
466                        by the GML Application Schema.
467                     </xsd:documentation>
468                  </xsd:annotation>
469               </xsd:element>
470            </xsd:sequence>
471            <xsd:attribute name="outputFormat"
472                           type="xsd:string" use="optional"
473                           default="text/xml; subtype=gml/3.1.1">
474               <xsd:annotation>
475                  <xsd:documentation>
476                     The outputFormat attribute is used to specify what schema
477                     description language should be used to describe features.
478                     The default value of 'text/xml; subtype=3.1.1' means that
479                     the WFS must generate a GML3 application schema that can
480                     be used to validate the GML3 output of a GetFeature
481                     request or feature instances specified in Transaction
482                     operations.
483                     For the purposes of experimentation, vendor extension,
484                     or even extensions that serve a specific community of
485                     interest, other acceptable output format values may be
486                     advertised by a WFS service in the capabilities document.
487                     The meaning of such values in not defined in the WFS
488                     specification.  The only proviso is such cases is that
489                     clients may safely ignore outputFormat values that do
490                     not recognize.
491                  </xsd:documentation>
492               </xsd:annotation>
493            </xsd:attribute>
494         </xsd:extension>
495      </xsd:complexContent>
496   </xsd:complexType>
497   <!-- RESPONSE -->
```



```
498    <!-- ==================================================================== -->
499    <!-- For the outputFormat value of 'text/xml; subtype=3.1.1' a WFS      -->
500    <!-- must generate a valid XML-Schema/GML3 application schema that      -->
501    <!-- describes that requested feature type(s).                          -->
502    <!-- ==================================================================== -->
503
504    <!-- ==================================================================== -->
505    <!-- =   GETFEATURES Request and Response                             = -->
506    <!-- ==================================================================== -->
507    <xsd:element name="GetFeature" type="wfs:GetFeatureType">
508       <xsd:annotation>
509          <xsd:documentation>
510             The GetFeature element is used to request that a Web Feature
511             Service return feature type instances of one or more feature
512             types.
513          </xsd:documentation>
514       </xsd:annotation>
515    </xsd:element>
516    <xsd:complexType name="GetFeatureType">
517       <xsd:annotation>
518          <xsd:documentation>
519             A GetFeature element contains one or more Query elements
520             that describe a query operation on one feature type.  In
521             response to a GetFeature request, a Web Feature Service
522             must be able to generate a GML3 response that validates
523             using a schema generated by the DescribeFeatureType request.
524             A Web Feature Service may support other possibly non-XML
525             (and even binary) output formats as long as those formats
526             are advertised in the capabilities document.
527          </xsd:documentation>
528       </xsd:annotation>
529       <xsd:complexContent>
530          <xsd:extension base="wfs:BaseRequestType">
531             <xsd:sequence>
532                <xsd:element ref="wfs:Query" maxOccurs="unbounded"/>
533             </xsd:sequence>
534             <xsd:attribute name="resultType"
535                            type="wfs:ResultTypeType" use="optional"
536                            default="results">
537                <xsd:annotation>
538                   <xsd:documentation>
539                      The resultType attribute is used to indicate
540                      what response a WFS should return to user once
541                      a GetFeature request is processed.
542                      Possible values are:
543                         results - meaning that the full response set
544                                   (i.e. all the feature instances)
545                                   should be returned.
546                         hits    - meaning that an empty response set
547                                   should be returned (i.e. no feature
548                                   instances should be returned) but
549                                   the "numberOfFeatures" attribute
550                                   should be set to the number of feature
551                                   instances that would be returned.
552                   </xsd:documentation>
553                </xsd:annotation>
554             </xsd:attribute>
555             <xsd:attribute name="outputFormat"
556                            type="xsd:string" use="optional"
557                            default="text/xml; subtype=gml/3.1.1">
558                <xsd:annotation>
559                   <xsd:documentation>
560                      The outputFormat attribute is used to specify the output
561                      format that the Web Feature Service should generate in
562                      response to a GetFeature or GetFeatureWithLock element.
563                      The default value of 'text/xml; subtype=gml/3.1.1'
564                      indicates that the output is an XML document that
565                      conforms to the Geography Markup Language (GML)
566                      Implementation Specification V3.1.1.
567                      For the purposes of experimentation, vendor extension,
568                      or even extensions that serve a specific community of
569                      interest, other acceptable output format values may be
```



```
570                     used to specify other formats as long as those values
571                     are advertised in the capabilities document.
572                     For example, the value WKB may be used to indicate that a
573                     Well Known Binary format be used to encode the output.
574                  </xsd:documentation>
575               </xsd:annotation>
576            </xsd:attribute>
577            <xsd:attribute name="maxFeatures"
578                           type="xsd:positiveInteger" use="optional">
579               <xsd:annotation>
580                  <xsd:documentation>
581                     The maxFeatures attribute is used to specify the maximum
582                     number of features that a GetFeature operation should
583                     generate (regardless of the actual number of query hits).
584                  </xsd:documentation>
585               </xsd:annotation>
586            </xsd:attribute>
587            <xsd:attribute name="traverseXlinkDepth"
588                           type="xsd:string" use="optional">
589               <xsd:annotation>
590                  <xsd:documentation>
591                     This attribute indicates the depth to which nested property
592                     XLink linking element locator attribute (href) XLinks are
593                     traversed and resolved if possible.  A value of "1"
594                     indicates that one linking element locator attribute
595                     (href) Xlink will be traversed and the referenced element
596                     returned if possible, but nested property XLink linking
597                     element locator attribute (href) XLinks in the returned
598                     element are not traversed.  A value of "*" indicates that
599                     all nested property XLink linking element locator attribute
600                     (href) XLinks will be traversed and the referenced elements
601                     returned if possible.  The range of valid values for this
602                     attribute consists of positive integers plus "*".
603                     If this attribute is not specified then no xlinks shall be
604                     resolved and the value of traverseXlinkExpiry attribute (if
605                     it specified) may be ignored.
606                  </xsd:documentation>
607               </xsd:annotation>
608            </xsd:attribute>
609            <xsd:attribute name="traverseXlinkExpiry"
610                           type="xsd:positiveInteger"
611                           use="optional">
612               <xsd:annotation>
613                  <xsd:documentation>
614                     The traverseXlinkExpiry attribute value is specified in
615                     minutes.  It indicates how long a Web Feature Service
616                     should wait to receive a response to a nested GetGmlObject
617                     request.
618                     This attribute is only relevant if a value is specified
619                     for the traverseXlinkDepth attribute.
620                  </xsd:documentation>
621               </xsd:annotation>
622            </xsd:attribute>
623         </xsd:extension>
624      </xsd:complexContent>
625   </xsd:complexType>
626   <xsd:simpleType name="ResultTypeType">
627      <xsd:restriction base="xsd:string">
628         <xsd:enumeration value="results">
629            <xsd:annotation>
630               <xsd:documentation>
631                  Indicates that a complete response should be generated
632                  by the WFS.  That is, all response feature instances
633                  should be returned to the client.
634               </xsd:documentation>
635            </xsd:annotation>
636         </xsd:enumeration>
637         <xsd:enumeration value="hits">
638            <xsd:annotation>
639               <xsd:documentation>
640                  Indicates that an empty response should be generated with
641                  the "numberOfFeatures" attribute set (i.e. no feature
```



```
642                  instances should be returned).  In this manner a client may
643                  determine the number of feature instances that a GetFeature
644                  request will return without having to actually get the
645                  entire result set back.
646               </xsd:documentation>
647            </xsd:annotation>
648         </xsd:enumeration>
649      </xsd:restriction>
650   </xsd:simpleType>
651   <xsd:element name="Query" type="wfs:QueryType">
652      <xsd:annotation>
653         <xsd:documentation>
654            The Query element is used to describe a single query.
655            One or more Query elements can be specified inside a
656            GetFeature element so that multiple queries can be
657            executed in one request.  The output from the various
658            queries are combined in a wfs:FeatureCollection element
659            to form the response document.
660         </xsd:documentation>
661      </xsd:annotation>
662   </xsd:element>
663   <xsd:complexType name="QueryType">
664      <xsd:annotation>
665         <xsd:documentation>
666            The Query element is of type QueryType.
667         </xsd:documentation>
668      </xsd:annotation>
669     <xsd:sequence>
670       <xsd:choice minOccurs="0" maxOccurs="unbounded">
671          <xsd:element ref="wfs:PropertyName">
672             <xsd:annotation>
673                <xsd:documentation>
674                   The Property element is used to specify one or more
675                   properties of a feature whose values are to be retrieved
676                   by a Web Feature Service.
677
678                   While a Web Feature Service should endeavour to satisfy
679                   the exact request specified, in some instance this may
680                   not be possible.  Specifically, a Web Feature Service
681                   must generate a valid GML3 response to a Query operation.
682                   The schema used to generate the output may include
683                   properties that are mandatory.  In order that the output
684                   validates, these mandatory properties must be specified
685                   in the request.  If they are not, a Web Feature Service
686                   may add them automatically to the Query before processing
687                   it.  Thus a client application should, in general, be
688                   prepared to receive more properties than it requested.
689
690                   Of course, using the DescribeFeatureType request, a client
691                   application can determine which properties are mandatory
692                   and request them in the first place.
693                </xsd:documentation>
694             </xsd:annotation>
695          </xsd:element>
696          <xsd:element ref="ogc:Function">
697             <xsd:annotation>
698                <xsd:documentation>
699                   A function may be used as a select item in a query.
700                   However, if a function is used, care must be taken
701                   to ensure that the result type matches the type in the
702
703                </xsd:documentation>
704             </xsd:annotation>
705          </xsd:element>
706       </xsd:choice>
707       <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1">
708          <xsd:annotation>
709             <xsd:documentation>
710                The Filter element is used to define spatial and/or non-spatial
711                constraints on query.  Spatial constrains use GML3 to specify
712                the constraining geometry.  A full description of the Filter
713                element can be found in the Filter Encoding Implementation
```



```
714                Specification.
715              </xsd:documentation>
716            </xsd:annotation>
717          </xsd:element>
718          <xsd:element ref="ogc:SortBy" minOccurs="0" maxOccurs="1">
719             <xsd:annotation>
720                <xsd:documentation>
721                   The SortBy element is used specify property names whose
722                   values should be used to order (upon presentation) the
723                   set of feature instances that satisfy the query.
724                </xsd:documentation>
725             </xsd:annotation>
726          </xsd:element>
727       </xsd:sequence>
728       <xsd:attribute name="handle"
729                      type="xsd:string" use="optional">
730          <xsd:annotation>
731             <xsd:documentation>
732                The handle attribute allows a client application
733                to assign a client-generated identifier for the
734                Query.  The handle is included to facilitate error
735                reporting.  If one Query in a GetFeature request
736                causes an exception, a WFS may report the handle
737                to indicate which query element failed.  If the a
738                handle is not present, the WFS may use other means
739                to localize the error (e.g. line numbers).
740             </xsd:documentation>
741          </xsd:annotation>
742       </xsd:attribute>
743       <xsd:attribute name="typeName"
744                      type="wfs:TypeNameListType" use="required">
745          <xsd:annotation>
746             <xsd:documentation>
747                The typeName attribute is a list of one or more
748                feature type names that indicate which types
749                of feature instances should be included in the
750                reponse set.  Specifying more than one typename
751                indicates that a join operation is being performed.
752                All the names in the typeName list must be valid
753                types that belong to this query's feature content
754                as defined by the GML Application Schema.
755             </xsd:documentation>
756          </xsd:annotation>
757       </xsd:attribute>
758       <xsd:attribute name="featureVersion"
759                      type="xsd:string" use="optional">
760          <xsd:annotation>
761             <xsd:documentation>
762                For systems that implement versioning, the featureVersion
763                attribute is used to specify which version of a particular
764                feature instance is to be retrieved.  A value of ALL means
765                that all versions should be retrieved.  An integer value
766                'i', means that the ith version should be retrieve if it
767                exists or the most recent version otherwise.
768             </xsd:documentation>
769          </xsd:annotation>
770       </xsd:attribute>
771       <xsd:attribute name="srsName" type="xsd:anyURI" use="optional">
772          <xsd:annotation>
773             <xsd:documentation>
774                This attribute is used to specify a specific WFS-supported SRS
775                that should be used for returned feature geometries.  The value
776                may be the WFS StorageSRS value, DefaultRetrievalSRS value, or
777                one of AdditionalSRS values.  If no srsName value is supplied,
778                then the features will be returned using either the
779                DefaultRetrievalSRS, if specified, and StorageSRS otherwise.
780                For feature types with no spatial properties, this attribute
781                must not be specified or ignored if it is specified.
782             </xsd:documentation>
783          </xsd:annotation>
784       </xsd:attribute>
785    </xsd:complexType>
```



```
<xsd:simpleType name="Base_TypeNameListType">
   <xsd:list itemType="xsd:QName"/>
</xsd:simpleType>
<xsd:simpleType name="TypeNameListType">
   <xsd:restriction base="wfs:Base_TypeNameListType">
      <xsd:pattern value="((\w:)?\w(=\w)?){1,}">
         <xsd:annotation>
            <xsd:documentation>
               Example typeName attribute value might be:

                  typeName="ns1:Inwatera_1m=A, ns2:CoastL_1M=B"

               In this example, A is an alias for ns1:Inwatera_1m
               and B is an alias for ns2:CoastL_1M.
            </xsd:documentation>
         </xsd:annotation>
      </xsd:pattern>
   </xsd:restriction>
</xsd:simpleType>
<!-- RESPONSE -->
<xsd:element name="FeatureCollection"
             type="wfs:FeatureCollectionType"
             substitutionGroup="gml:_FeatureCollection">
   <xsd:annotation>
      <xsd:documentation>
         This element is a container for the response to a GetFeature
         or GetFeatureWithLock (WFS-transaction.xsd) request.
      </xsd:documentation>
   </xsd:annotation>
</xsd:element>
<xsd:complexType name="FeatureCollectionType">
   <xsd:annotation>
      <xsd:documentation>
         This type defines a container for the response to a
         GetFeature or GetFeatureWithLock request.  If the
         request is GetFeatureWithLock, the lockId attribute
         must be populated.  The lockId attribute can otherwise
         be safely ignored.
      </xsd:documentation>
   </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureCollectionType">
      <xsd:attribute name="lockId" type="xsd:string" use="optional">
         <xsd:annotation>
            <xsd:documentation>
               The value of the lockId attribute is an identifier
               that a Web Feature Service generates when responding
               to a GetFeatureWithLock request.  A client application
               can use this value in subsequent operations (such as a
               Transaction request) to reference the set of locked
               features.
            </xsd:documentation>
         </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="timeStamp" type="xsd:dateTime" use="optional">
         <xsd:annotation>
            <xsd:documentation>
               The timeStamp attribute should contain the date and time
               that the response was generated.
            </xsd:documentation>
         </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="numberOfFeatures"
                     type="xsd:nonNegativeInteger"
                     use="optional">
         <xsd:annotation>
            <xsd:documentation>
               The numberOfFeatures attribute should contain a
               count of the number of features in the response.
               That is a count of all features elements dervied
               from gml:AbstractFeatureType.
            </xsd:documentation>
```

```
858            </xsd:annotation>
859          </xsd:attribute>
860        </xsd:extension>
861      </xsd:complexContent>
862    </xsd:complexType>
863
864    <!-- ================================================================= -->
865    <!-- =   GETGMLOBJECT Request and Response                           = -->
866    <!-- ================================================================= -->
867    <xsd:element name="GetGmlObject" type="wfs:GetGmlObjectType">
868       <xsd:annotation>
869          <xsd:documentation>
870             The GetGmlObject element is used to request that a Web Feature
871             Service return an element with a gml:id attribute value specified
872             by an ogc:GmlObjectId.
873          </xsd:documentation>
874       </xsd:annotation>
875    </xsd:element>
876    <xsd:complexType name="GetGmlObjectType">
877       <xsd:annotation>
878          <xsd:documentation>
879             A GetGmlObjectType element contains exactly one GmlObjectId.
880             The value of the gml:id attribute on that GmlObjectId is used
881             as a unique key to retrieve the complex element with a
882             gml:id attribute with the same value.
883          </xsd:documentation>
884       </xsd:annotation>
885       <xsd:complexContent>
886          <xsd:extension base="wfs:BaseRequestType">
887             <xsd:sequence>
888                <xsd:element ref="ogc:GmlObjectId"/>
889             </xsd:sequence>
890             <xsd:attribute name="outputFormat"
891                            type="xsd:string" use="optional" default="GML3"/>
892             <xsd:attribute name="traverseXlinkDepth"
893                            type="xsd:string" use="required">
894                <xsd:annotation>
895                   <xsd:documentation>
896                      This attribute indicates the depth to which nested
897                      property XLink linking element locator attribute
898                      (href) XLinks are traversed and resolved if possible.
899                      A value of "1" indicates that one linking element
900                      locator attribute (href) XLink will be traversed
901                      and the referenced element returned if possible, but
902                      nested property XLink linking element locator attribute
903                      (href) XLinks in the returned element are not traversed.
904                      A value of "*" indicates that all nested property XLink
905                      linking element locator attribute (href) XLinks will be
906                      traversed and the referenced elements returned if
907                      possible.  The range of valid values for this attribute
908                      consists of positive integers plus "*".
909                   </xsd:documentation>
910                </xsd:annotation>
911             </xsd:attribute>
912             <xsd:attribute name="traverseXlinkExpiry"
913                            type="xsd:positiveInteger"
914                            use="optional">
915                <xsd:annotation>
916                   <xsd:documentation>
917                      The traverseXlinkExpiry attribute value is specified
918                      in minutes.  It indicates how long a Web Feature Service
919                      should wait to receive a response to a nested GetGmlObject
920                      request.
921                   </xsd:documentation>
922                </xsd:annotation>
923             </xsd:attribute>
924          </xsd:extension>
925       </xsd:complexContent>
926    </xsd:complexType>
927    <!-- RESPONSE -->
928    <!-- ================================================================= -->
929    <!-- The response to a GetGMLObject request is a GML3 fragment(s) that -->
```

```
930    <!-- has (have) the gml:id('s) specified in the request.               -->
931    <!-- ================================================================= -->
932
933    <!-- ================================================================= -->
934    <!-- =   GETFEATUREWITHLOCK Request and Response                     = -->
935    <!-- ================================================================= -->
936    <!-- REQUEST -->
937    <xsd:element name="GetFeatureWithLock" type="wfs:GetFeatureWithLockType">
938       <xsd:annotation>
939          <xsd:documentation>
940             This is the root element for the GetFeatureWithLock request.
941             The GetFeatureWithLock operation performs identically to a
942             GetFeature request except that the GetFeatureWithLock request
943             locks all the feature instances in the result set and returns
944             a lock identifier to a client application in the response.
945             The lock identifier is returned to the client application
946             using the lockId attribute define on the wfs:FeatureCollection
947             element.
948          </xsd:documentation>
949       </xsd:annotation>
950    </xsd:element>
951    <xsd:complexType name="GetFeatureWithLockType">
952       <xsd:annotation>
953          <xsd:documentation>
954             A GetFeatureWithLock request operates identically to a
955             GetFeature request expect that it attempts to lock the
956             feature instances in the result set and includes a lock
957             identifier in its response to a client.  A lock identifier
958             is an identifier generated by a Web Feature Service that
959             a client application can use, in subsequent operations,
960             to reference the locked set of feature instances.
961          </xsd:documentation>
962       </xsd:annotation>
963       <xsd:complexContent>
964          <xsd:extension base="wfs:BaseRequestType">
965             <xsd:sequence>
966                <xsd:element ref="wfs:Query" maxOccurs="unbounded"/>
967             </xsd:sequence>
968             <xsd:attribute name="expiry"
969                            type="xsd:positiveInteger"
970                            use="optional" default="5">
971                <xsd:annotation>
972                   <xsd:documentation>
973                      The expiry attribute is used to set the length
974                      of time (expressed in minutes) that features will
975                      remain locked as a result of a GetFeatureWithLock
976                      request.  After the expiry period elapses, the
977                      locked resources must be released.  If the
978                      expiry attribute is not set, then the default
979                      value of 5 minutes will be enforced.
980                   </xsd:documentation>
981                </xsd:annotation>
982             </xsd:attribute>
983             <xsd:attribute name="resultType"
984                            type="wfs:ResultTypeType" use="optional"
985                            default="results">
986                <xsd:annotation>
987                   <xsd:documentation>
988                      See definition of wfs:GetFeatureType.
989                   </xsd:documentation>
990                </xsd:annotation>
991             </xsd:attribute>
992             <xsd:attribute name="outputFormat"
993                            type="xsd:string" use="optional"
994                            default="text/xml; subtype=gml/3.1.1">
995                <xsd:annotation>
996                   <xsd:documentation>
997                      See definition of wfs:GetFeatureType.
998                   </xsd:documentation>
999                </xsd:annotation>
1000            </xsd:attribute>
1001            <xsd:attribute name="maxFeatures"
```



```
                        type="xsd:positiveInteger" use="optional">
               <xsd:annotation>
                  <xsd:documentation>
                     See definition of wfs:GetFeatureType.
                  </xsd:documentation>
               </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="traverseXlinkDepth"
                        type="xsd:string" use="optional">
               <xsd:annotation>
                  <xsd:documentation>
                     See definition of wfs:GetFeatureType.
                  </xsd:documentation>
               </xsd:annotation>
            </xsd:attribute>
            <xsd:attribute name="traverseXlinkExpiry"
                        type="xsd:positiveInteger" use="optional">
               <xsd:annotation>
                  <xsd:documentation>
                     See definition of wfs:GetFeatureType.
                  </xsd:documentation>
               </xsd:annotation>
            </xsd:attribute>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>

   <!-- ==================================================================== -->
   <!-- =   LOCKFEATURE Request and Response                               = -->
   <!-- ==================================================================== -->
   <!-- REQUEST -->
   <xsd:element name="LockFeature" type="wfs:LockFeatureType">
      <xsd:annotation>
         <xsd:documentation>
            This is the root element for a LockFeature request.
            The LockFeature request can be used to lock one or
            more feature instances.
         </xsd:documentation>
      </xsd:annotation>
   </xsd:element>
   <xsd:complexType name="LockFeatureType">
      <xsd:annotation>
         <xsd:documentation>
            This type defines the LockFeature operation.  The LockFeature
            element contains one or more Lock elements that define which
            features of a particular type should be locked.  A lock
            identifier (lockId) is returned to the client application which
            can be used by subsequent operations to reference the locked
            features.
         </xsd:documentation>
      </xsd:annotation>
      <xsd:complexContent>
         <xsd:extension base="wfs:BaseRequestType">
            <xsd:sequence>
               <xsd:element name="Lock" type="wfs:LockType"
                            maxOccurs="unbounded">
                  <xsd:annotation>
                     <xsd:documentation>
                        The lock element is used to indicate which feature
                        instances of particular type are to be locked.
                     </xsd:documentation>
                  </xsd:annotation>
               </xsd:element>
            </xsd:sequence>
            <xsd:attribute name="expiry"
                        type="xsd:positiveInteger"
                        use="optional" default="5">
               <xsd:annotation>
                  <xsd:documentation>
                     The expiry attribute is used to set the length
                     of time (expressed in minutes) that features will
                     remain locked as a result of a LockFeature
```



```
1074                     request.  After the expiry period elapses, the
1075                     locked resources must be released.  If the
1076                     expiry attribute is not set, then the default
1077                     value of 5 minutes will be enforced.
1078                  </xsd:documentation>
1079               </xsd:annotation>
1080            </xsd:attribute>
1081            <xsd:attribute name="lockAction"
1082                           type="wfs:AllSomeType"
1083                           use="optional" default="ALL">
1084               <xsd:annotation>
1085                  <xsd:documentation>
1086                     The lockAction attribute is used to indicate what
1087                     a Web Feature Service should do when it encounters
1088                     a feature instance that has already been locked by
1089                     another client application.
1090
1091                     Valid values are ALL or SOME.
1092
1093                     ALL means that the Web Feature Service must acquire
1094                     locks on all the requested feature instances.  If it
1095                     cannot acquire those locks then the request should
1096                     fail.  In this instance, all locks acquired by the
1097                     operation should be released.
1098
1099                     SOME means that the Web Feature Service should lock
1100                     as many of the requested features as it can.
1101                  </xsd:documentation>
1102               </xsd:annotation>
1103            </xsd:attribute>
1104         </xsd:extension>
1105      </xsd:complexContent>
1106   </xsd:complexType>
1107   <xsd:simpleType name="AllSomeType">
1108      <xsd:restriction base="xsd:string">
1109         <xsd:enumeration value="ALL"/>
1110         <xsd:enumeration value="SOME"/>
1111      </xsd:restriction>
1112   </xsd:simpleType>
1113   <xsd:complexType name="LockType">
1114      <xsd:annotation>
1115         <xsd:documentation>
1116            This type defines the Lock element.  The Lock element
1117            defines a locking operation on feature instances of
1118            a single type. An OGC Filter is used to constrain the
1119            scope of the operation.  Features to be locked can be
1120            identified individually by using their feature identifier
1121            or they can be locked by satisfying the spatial and
1122            non-spatial constraints defined in the filter.
1123         </xsd:documentation>
1124      </xsd:annotation>
1125      <xsd:sequence>
1126         <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1"/>
1127      </xsd:sequence>
1128      <xsd:attribute name="handle" type="xsd:string" use="optional">
1129         <xsd:annotation>
1130            <xsd:documentation>
1131               The handle attribute allows a client application
1132               to assign a client-generated request identifier
1133               to a Lock action.  The handle is included to
1134               facilitate error reporting.  If one of a set of
1135               Lock actions failed while processing a LockFeature
1136               request, a WFS may report the handle in an exception
1137               report to localize the error.  If a handle is not
1138               present then a WFS may employ some other means of
1139               localizing the error (e.g. line number).
1140            </xsd:documentation>
1141         </xsd:annotation>
1142      </xsd:attribute>
1143      <xsd:attribute name="typeName" type="xsd:QName" use="required">
1144        <xsd:annotation>
1145           <xsd:documentation>
```



```
1146              The value of the typeName attribute is the name
1147              of the feature type to be updated. The name
1148              specified must be a valid type that belongs to
1149              the feature content as defined by the GML
1150              Application Schema.
1151           </xsd:documentation>
1152        </xsd:annotation>
1153     </xsd:attribute>
1154  </xsd:complexType>
1155  <!-- RESPONSE -->
1156  <xsd:element name="LockFeatureResponse"
1157               type="wfs:LockFeatureResponseType">
1158     <xsd:annotation>
1159        <xsd:documentation>
1160           The LockFeatureResponse element contains a report
1161           about the completion status of a LockFeature request.
1162        </xsd:documentation>
1163     </xsd:annotation>
1164  </xsd:element>
1165  <xsd:complexType name="LockFeatureResponseType">
1166     <xsd:annotation>
1167        <xsd:documentation>
1168           The LockFeatureResponseType is used to define an
1169           element to contains the response to a LockFeature
1170           operation.
1171        </xsd:documentation>
1172     </xsd:annotation>
1173     <xsd:sequence>
1174        <xsd:element ref="wfs:LockId">
1175           <xsd:annotation>
1176              <xsd:documentation>
1177                 The LockFeatureResponse includes a LockId element
1178                 that contains a lock identifier.  The lock identifier
1179                 can be used by a client, in subsequent operations, to
1180                 operate upon the locked feature instances.
1181              </xsd:documentation>
1182           </xsd:annotation>
1183        </xsd:element>
1184        <xsd:element name="FeaturesLocked"
1185                     type="wfs:FeaturesLockedType" minOccurs="0">
1186           <xsd:annotation>
1187              <xsd:documentation>
1188                 The LockFeature or GetFeatureWithLock operations
1189                 identify and attempt to lock a set of feature
1190                 instances that satisfy the constraints specified
1191                 in the request.  In the event that the lockAction
1192                 attribute (on the LockFeature or GetFeatureWithLock
1193                 elements) is set to SOME, a Web Feature Service will
1194                 attempt to lock as many of the feature instances from
1195                 the result set as possible.
1196
1197                 The FeaturesLocked element contains list of ogc:FeatureId
1198                 elements enumerating the feature instances that a WFS
1199                 actually managed to lock.
1200              </xsd:documentation>
1201           </xsd:annotation>
1202        </xsd:element>
1203        <xsd:element name="FeaturesNotLocked"
1204                     type="wfs:FeaturesNotLockedType" minOccurs="0">
1205           <xsd:annotation>
1206              <xsd:documentation>
1207                 In contrast to the FeaturesLocked element, the
1208                 FeaturesNotLocked element contains a list of
1209                 ogc:Filter elements identifying feature instances
1210                 that a WFS did not manage to lock because they were
1211                 already locked by another process.
1212              </xsd:documentation>
1213           </xsd:annotation>
1214        </xsd:element>
1215     </xsd:sequence>
1216  </xsd:complexType>
1217  <xsd:complexType name="FeaturesLockedType">
```



```
1218      <xsd:sequence maxOccurs="unbounded">
1219        <xsd:element ref="ogc:FeatureId"/>
1220      </xsd:sequence>
1221    </xsd:complexType>
1222    <xsd:complexType name="FeaturesNotLockedType">
1223      <xsd:sequence maxOccurs="unbounded">
1224        <xsd:element ref="ogc:FeatureId"/>
1225      </xsd:sequence>
1226    </xsd:complexType>
1227
1228    <!-- ==================================================================== -->
1229    <!-- =   TRANSACTION Request and Response                               = -->
1230    <!-- ==================================================================== -->
1231    <!-- REQUEST -->
1232    <xsd:element name="Transaction" type="wfs:TransactionType">
1233       <xsd:annotation>
1234          <xsd:documentation>
1235             This is the root element for a Transaction request.
1236             A transaction request allows insert, update and
1237             delete operations to be performed to create, change
1238             or remove feature instances.
1239          </xsd:documentation>
1240       </xsd:annotation>
1241    </xsd:element>
1242    <xsd:complexType name="TransactionType">
1243       <xsd:annotation>
1244          <xsd:documentation>
1245             The TransactionType defines the Transaction operation.  A
1246             Transaction element contains one or more Insert, Update
1247             Delete and Native elements that allow a client application
1248             to create, modify or remove feature instances from the
1249             feature repository that a Web Feature Service controls.
1250          </xsd:documentation>
1251       </xsd:annotation>
1252       <xsd:complexContent>
1253          <xsd:extension base="ows:GetCapabilitiesType">
1254             <xsd:sequence>
1255                <xsd:element ref="wfs:LockId" minOccurs="0">
1256                   <xsd:annotation>
1257                      <xsd:documentation>
1258                         In order for a client application to operate upon
1259                         locked feature instances, the Transaction request
1260                         must include the LockId element.  The content of
1261                         this element must be the lock identifier the client
1262                         application obtained from a previous
1263                         GetFeatureWithLock or LockFeature operation.
1264
1265                         If the correct lock identifier is specified the Web
1266                         Feature Service knows that the client application may
1267                         operate upon the locked feature instances.
1268
1269                         No LockId element needs to be specified to operate upon
1270                         unlocked features.
1271                      </xsd:documentation>
1272                   </xsd:annotation>
1273                </xsd:element>
1274                <xsd:choice minOccurs="0" maxOccurs="unbounded">
1275                   <xsd:element ref="wfs:Insert"/>
1276                   <xsd:element ref="wfs:Update"/>
1277                   <xsd:element ref="wfs:Delete"/>
1278                   <xsd:element ref="wfs:Native"/>
1279                </xsd:choice>
1280             </xsd:sequence>
1281             <xsd:attribute name="releaseAction"
1282                            type="wfs:AllSomeType" use="optional">
1283                <xsd:annotation>
1284                   <xsd:documentation>
1285                      The releaseAction attribute is used to control how a Web
1286                      Feature service releases locks on feature instances after
1287                      a Transaction request has been processed.
1288
1289                      Valid values are ALL or SOME.
```



```
1290
1291                     A value of ALL means that the Web Feature Service should
1292                     release the locks of all feature instances locked with the
1293                     specified lockId regardless or whether or not the features
1294                     were actually modified.
1295
1296                     A value of SOME means that the Web Feature Service will
1297                     only release the locks held on feature instances that
1298                     were actually operated upon by the transaction.  The
1299                     lockId that the client application obtained shall remain
1300                     valid and the other, unmodified, feature instances shall
1301                     remain locked.
1302
1303                     If the expiry attribute was specified in the original
1304                     operation that locked the feature instances, then the
1305                     expiry counter will be reset to give the client
1306                     application that same amount of time to post subsequent
1307                     transactions against the locked features.
1308                  </xsd:documentation>
1309               </xsd:annotation>
1310            </xsd:attribute>
1311         </xsd:extension>
1312      </xsd:complexContent>
1313   </xsd:complexType>
1314   <xsd:element name="LockId" type="xsd:string">
1315      <xsd:annotation>
1316         <xsd:documentation>
1317            The LockId element contains the value of the lock identifier
1318            obtained by a client application from a previous GetFeatureWithLock
1319            or LockFeature request.
1320         </xsd:documentation>
1321      </xsd:annotation>
1322   </xsd:element>
1323   <xsd:element name="Insert" type="wfs:InsertElementType">
1324      <xsd:annotation>
1325         <xsd:documentation>
1326            The Insert element is used to indicate that the Web Feature
1327            Service should create a new instance of a feature type.  The
1328            feature instance is specified using GML3 and one or more
1329            feature instances to be created can be contained inside the
1330            Insert element.
1331         </xsd:documentation>
1332      </xsd:annotation>
1333   </xsd:element>
1334   <xsd:complexType name="InsertElementType">
1335      <xsd:annotation>
1336         <xsd:documentation>
1337            An Insert element may contain a feature collection or one
1338            or more feature instances to be inserted into the
1339            repository.
1340         </xsd:documentation>
1341      </xsd:annotation>
1342      <xsd:choice>
1343         <xsd:element ref="gml:_FeatureCollection" />
1344         <xsd:sequence>
1345            <xsd:element ref="gml:_Feature" maxOccurs="unbounded"/>
1346         </xsd:sequence>
1347      </xsd:choice>
1348      <xsd:attribute name="idgen"
1349                     type="wfs:IdentifierGenerationOptionType"
1350                     use="optional" default="GenerateNew">
1351         <xsd:annotation>
1352            <xsd:documentation>
1353               The idgen attribute control how a WFS generates identifiers
1354               from newly created feature instances using the Insert action.
1355               The default action is to have the WFS generate a new id for
1356               the features.  This is also backward compatible with WFS 1.0
1357               where the only action was for the WFS to generate an new id.
1358            </xsd:documentation>
1359         </xsd:annotation>
1360      </xsd:attribute>
1361      <xsd:attribute name="handle" type="xsd:string" use="optional">
```



```
1362          <xsd:annotation>
1363             <xsd:documentation>
1364                The handle attribute allows a client application
1365                to assign a client-generated request identifier
1366                to an Insert action.  The handle is included to
1367                facilitate error reporting.  If an Insert action
1368                in a Transaction request fails, then a WFS may
1369                include the handle in an exception report to localize
1370                the error.  If no handle is included of the offending
1371                Insert element then a WFS may employee other means of
1372                localizing the error (e.g. line number).
1373             </xsd:documentation>
1374          </xsd:annotation>
1375       </xsd:attribute>
1376       <xsd:attribute name="inputFormat" type="xsd:string"
1377                      use="optional" default="text/xml; subtype=gml/3.1.1">
1378          <xsd:annotation>
1379             <xsd:documentation>
1380                This inputFormat attribute is used to indicate
1381                the format used to encode a feature instance in
1382                an Insert element.  The default value of
1383                'text/xml; subtype=gml/3.1.1' is used to indicate
1384                that feature encoding is GML3.  Another example
1385                might be 'text/xml; subtype=gml/2.1.2' indicating
1386                that the feature us encoded in GML2.  A WFS must
1387                declare in the capabilities document, using a
1388                Parameter element, which version of GML it supports.
1389             </xsd:documentation>
1390          </xsd:annotation>
1391       </xsd:attribute>
1392       <xsd:attribute name="srsName" type="xsd:anyURI" use="optional">
1393         <xsd:annotation>
1394            <xsd:documentation>
1395               ===== PAV 12NOV2004 ====
1396               WHY IS THIS HERE? WOULDN'T WE KNOW THE INCOMING SRS FROM THE
1397               GML GEOMETRY ELEMENTS?   I ASSUME THAT IF THE INCOMING SRS
1398               DOES NOT MATCH ONE OF THE STORAGE SRS(s) THEN THE WFS WOULD
1399               EITHER PROJECT INTO THE STORAGE SRS OR RAISE AN EXCEPTION.
1400            </xsd:documentation>
1401         </xsd:annotation>
1402       </xsd:attribute>
1403    </xsd:complexType>
1404    <xsd:simpleType name="IdentifierGenerationOptionType">
1405       <xsd:restriction base="xsd:string">
1406          <xsd:enumeration value="UseExisting">
1407             <xsd:annotation>
1408                <xsd:documentation>
1409                   The UseExsiting value indicates that WFS should not
1410                   generate a new feature identifier for the feature
1411                   being inserted into the repositry.  Instead, the WFS
1412                   should use the identifier encoded if the feature.
1413                   If a duplicate exists then the WFS should raise an
1414                   exception.
1415                </xsd:documentation>
1416             </xsd:annotation>
1417          </xsd:enumeration>
1418          <xsd:enumeration value="ReplaceDuplicate">
1419             <xsd:annotation>
1420                <xsd:documentation>
1421                   The ReplaceDuplicate value indicates that WFS should
1422                   not generate a new feature identifier for the feature
1423                   being inserted into the repositry.  Instead, the WFS
1424                   should use the identifier encoded if the feature.
1425                   If a duplicate exists then the WFS should replace the
1426                   existing feature instance with the one encoded in the
1427                   Insert action.
1428                </xsd:documentation>
1429             </xsd:annotation>
1430          </xsd:enumeration>
1431          <xsd:enumeration value="GenerateNew">
1432             <xsd:annotation>
1433                <xsd:documentation>
```

```
1434                  The GenerateNew value indicates that WFS should
1435                  generate a new unique feature identifier for the
1436                  feature being inserted into the repositry.
1437               </xsd:documentation>
1438            </xsd:annotation>
1439         </xsd:enumeration>
1440      </xsd:restriction>
1441   </xsd:simpleType>
1442   <xsd:element name="Update" type="wfs:UpdateElementType">
1443      <xsd:annotation>
1444         <xsd:documentation>
1445            One or more existing feature instances can be changed by
1446            using the Update element.
1447         </xsd:documentation>
1448      </xsd:annotation>
1449   </xsd:element>
1450   <xsd:complexType name="UpdateElementType">
1451      <xsd:sequence>
1452         <xsd:element ref="wfs:Property" maxOccurs="unbounded">
1453            <xsd:annotation>
1454               <xsd:documentation>
1455                  Changing or updating a feature instance means that
1456                  the current value of one or more properties of
1457                  the feature are replaced with new values.  The Update
1458                  element contains  one or more Property elements.  A
1459                  Property element contains the name or a feature property
1460                  who's value is to be changed and the replacement value
1461                  for that property.
1462               </xsd:documentation>
1463            </xsd:annotation>
1464         </xsd:element>
1465         <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1">
1466            <xsd:annotation>
1467               <xsd:documentation>
1468                  The Filter element is used to constrain the scope
1469                  of the update operation to those features identified
1470                  by the filter.  Feature instances can be specified
1471                  explicitly and individually using the identifier of
1472                  each feature instance OR a set of features to be
1473                  operated on can be identified by specifying spatial
1474                  and non-spatial constraints in the filter.
1475                  If no filter is specified then update operation
1476                  applies to all feature instances.
1477               </xsd:documentation>
1478            </xsd:annotation>
1479         </xsd:element>
1480      </xsd:sequence>
1481      <xsd:attribute name="handle" type="xsd:string" use="optional">
1482         <xsd:annotation>
1483            <xsd:documentation>
1484               The handle attribute allows a client application
1485               to assign a client-generated request identifier
1486               to an Insert action.  The handle is included to
1487               facilitate error reporting.  If an Update action
1488               in a Transaction request fails, then a WFS may
1489               include the handle in an exception report to localize
1490               the error.  If no handle is included of the offending
1491               Insert element then a WFS may employee other means of
1492               localizing the error (e.g. line number).
1493            </xsd:documentation>
1494         </xsd:annotation>
1495      </xsd:attribute>
1496      <xsd:attribute name="typeName" type="xsd:QName" use="required">
1497        <xsd:annotation>
1498           <xsd:documentation>
1499              The value of the typeName attribute is the name
1500              of the feature type to be updated. The name
1501              specified must be a valid type that belongs to
1502              the feature content as defined by the GML
1503              Application Schema.
1504           </xsd:documentation>
1505        </xsd:annotation>
```



```
      </xsd:attribute>
      <xsd:attribute name="inputFormat" type="xsd:string"
                     use="optional" default="x-application/gml:3">
         <xsd:annotation>
            <xsd:documentation>
               This inputFormat attribute is used to indicate
               the format used to encode a feature instance in
               an Insert element.  The default value of
               'text/xml; subtype=gml/3.1.1' is used to indicate
               that feature encoding is GML3.  Another example
               might be 'text/xml; subtype=gml/2.1.2' indicating
               that the feature us encoded in GML2.  A WFS must
               declare in the capabilities document, using a
               Parameter element, which version of GML it supports.
            </xsd:documentation>
         </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="srsName" type="xsd:anyURI" use="optional">
        <xsd:annotation>
           <xsd:documentation>
               DO WE NEED THIS HERE?
           </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
   </xsd:complexType>
   <xsd:element name="Property" type="wfs:PropertyType">
      <xsd:annotation>
         <xsd:documentation>
            The Property element is used to specify the new
            value of a feature property inside an Update
            element.
         </xsd:documentation>
      </xsd:annotation>
   </xsd:element>
   <xsd:complexType name="PropertyType">
      <xsd:sequence>
         <xsd:element name="Name" type="xsd:QName">
            <xsd:annotation>
               <xsd:documentation>
                  The Name element contains the name of a feature property
                  to be updated.
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
         <xsd:element name="Value" minOccurs="0">
            <xsd:annotation>
               <xsd:documentation>
                  The Value element contains the replacement value for the
                  named property.
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:element name="Delete" type="wfs:DeleteElementType">
      <xsd:annotation>
         <xsd:documentation>
            The Delete element is used to indicate that one or more
            feature instances should be removed from the feature
            repository.
         </xsd:documentation>
      </xsd:annotation>
   </xsd:element>
   <xsd:complexType name="DeleteElementType">
      <xsd:sequence>
         <xsd:element ref="ogc:Filter" minOccurs="1" maxOccurs="1">
            <xsd:annotation>
               <xsd:documentation>
                  The Filter element is used to constrain the scope
                  of the delete operation to those features identified
                  by the filter.  Feature instances can be specified
                  explicitly and individually using the identifier of
```



```
1578                  each feature instance OR a set of features to be
1579                  operated on can be identified by specifying spatial
1580                  and non-spatial constraints in the filter.
1581                  If no filter is specified then an exception should
1582                  be raised since it is unlikely that a client application
1583                  intends to delete all feature instances.
1584               </xsd:documentation>
1585            </xsd:annotation>
1586         </xsd:element>
1587      </xsd:sequence>
1588      <xsd:attribute name="handle" type="xsd:string" use="optional">
1589         <xsd:annotation>
1590            <xsd:documentation>
1591               The handle attribute allows a client application
1592               to assign a client-generated request identifier
1593               to an Insert action.  The handle is included to
1594               facilitate error reporting.  If a Delete action
1595               in a Transaction request fails, then a WFS may
1596               include the handle in an exception report to localize
1597               the error.  If no handle is included of the offending
1598               Insert element then a WFS may employee other means of
1599               localizing the error (e.g. line number).
1600            </xsd:documentation>
1601         </xsd:annotation>
1602      </xsd:attribute>
1603      <xsd:attribute name="typeName" type="xsd:QName" use="required">
1604        <xsd:annotation>
1605           <xsd:documentation>
1606              The value of the typeName attribute is the name
1607              of the feature type to be updated. The name
1608              specified must be a valid type that belongs to
1609              the feature content as defined by the GML
1610              Application Schema.
1611           </xsd:documentation>
1612        </xsd:annotation>
1613      </xsd:attribute>
1614   </xsd:complexType>
1615   <xsd:element name="Native" type="wfs:NativeType">
1616      <xsd:annotation>
1617         <xsd:documentation>
1618            Many times, a Web Feature Service interacts with a repository
1619            that may have special vendor specific capabilities.  The native
1620            element allows vendor specific command to be passed to the
1621            repository via the Web Feature Service.
1622         </xsd:documentation>
1623      </xsd:annotation>
1624   </xsd:element>
1625   <xsd:complexType name="NativeType">
1626      <xsd:attribute name="vendorId" type="xsd:string" use="required">
1627         <xsd:annotation>
1628            <xsd:documentation>
1629               The vendorId attribute is used to specify the name of
1630               vendor who's vendor specific command the client
1631               application wishes to execute.
1632            </xsd:documentation>
1633         </xsd:annotation>
1634      </xsd:attribute>
1635      <xsd:attribute name="safeToIgnore" type="xsd:boolean" use="required">
1636         <xsd:annotation>
1637            <xsd:documentation>
1638               In the event that a Web Feature Service does not recognize
1639               the vendorId or does not recognize the vendor specific command,
1640               the safeToIgnore attribute is used to indicate whether the
1641               exception can be safely ignored.  A value of TRUE means that
1642               the Web Feature Service may ignore the command.  A value of
1643               FALSE means that a Web Feature Service cannot ignore the
1644               command and an exception should be raised if a problem is
1645               encountered.
1646            </xsd:documentation>
1647         </xsd:annotation>
1648      </xsd:attribute>
1649   </xsd:complexType>
```



```
<!-- REPONSE -->
<xsd:element name="TransactionResponse"
             type="wfs:TransactionResponseType">
   <xsd:annotation>
      <xsd:documentation>
         The TransactionResponse element contains a report
         about the completion status of a Transaction operation.
      </xsd:documentation>
   </xsd:annotation>
</xsd:element>
<xsd:complexType name="TransactionResponseType">
   <xsd:annotation>
      <xsd:documentation xml:lang="en">
         The response for a transaction request that was successfully
         completed. If the transaction failed for any reason, an
         exception report is returned instead.
      </xsd:documentation>
   </xsd:annotation>
   <xsd:sequence>
      <xsd:element name="TransactionSummary"
                   type="wfs:TransactionSummaryType">
         <xsd:annotation>
            <xsd:documentation xml:lang="en">
               The TransactionSummary element is used to summarize
               the number of feature instances affected by the
               transaction.
            </xsd:documentation>
         </xsd:annotation>
      </xsd:element>
      <xsd:element name="TransactionResults"
                   type="wfs:TransactionResultsType"
                   minOccurs="0">
         <xsd:annotation>
            <xsd:documentation xml:lang="en">
               For systems that do not support atomic transactions,
               the TransactionResults element may be used to report
               exception codes and messages for all actions of a
               transaction that failed to execute successfully.
            </xsd:documentation>
         </xsd:annotation>
      </xsd:element>
      <xsd:element name="InsertResults"
                   type="wfs:InsertResultType">
         <xsd:annotation>
            <xsd:documentation xml:lang="en">
               A transaction is a collection of Insert,Update and Delete
               actions.  The Update and Delete actions modify features
               that already exist.  The Insert action, however, creates
               new features.  The InsertResults element is used to
               report the identifiers of the newly created features.
            </xsd:documentation>
         </xsd:annotation>
      </xsd:element>
   </xsd:sequence>
   <xsd:attribute name="version"
                  type="xsd:string" use="required" fixed="1.1.0">
      <xsd:annotation>
         <xsd:documentation>
            The version attribute contains the version of the request
            that generated this response.  So a V1.1.0 transaction
            request generates a V1.1.0 transaction response.
         </xsd:documentation>
      </xsd:annotation>
   </xsd:attribute>
</xsd:complexType>
<xsd:complexType name="TransactionSummaryType">
   <xsd:annotation>
      <xsd:documentation xml:lang="en">
         Reports the total number of features affected by some kind
         of write action (i.e, insert, update, delete).
      </xsd:documentation>
   </xsd:annotation>
```



```
      <xsd:sequence>
         <xsd:element name="totalInserted"
                      type="xsd:nonNegativeInteger"
                      minOccurs="0"/>
         <xsd:element name="totalUpdated"
                      type="xsd:nonNegativeInteger"
                      minOccurs="0"/>
         <xsd:element name="totalDeleted"
                      type="xsd:nonNegativeInteger"
                      minOccurs="0"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="TransactionResultsType">
      <xsd:annotation>
         <xsd:documentation>
            The TransactionResults element may be used to report exception
            codes and messages for all actions of a transaction that failed
            to complete successfully.
         </xsd:documentation>
      </xsd:annotation>
      <xsd:sequence>
         <xsd:element name="Action" type="wfs:ActionType"
                      minOccurs="0" maxOccurs="unbounded">
            <xsd:annotation>
               <xsd:documentation>
                  The Action element reports an exception code
                  and exception message indicating why the
                  corresponding action of a transaction request
                  failed.
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="ActionType">
      <xsd:sequence>
         <xsd:element name="Message" type="xsd:string"
                      minOccurs="0" maxOccurs="1">
            <xsd:annotation>
               <xsd:documentation>
                  If an action fails, the message element may be used
                  to supply an exception message.
               </xsd:documentation>
            </xsd:annotation>
         </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="locator" type="xsd:string" use="required">
         <xsd:annotation>
            <xsd:documentation>
               The locator attribute is used to locate an action
               within a &lt;Transaction&gt; element.  The value
               of the locator attribute is either a string that
               is equal to the value of the handle attribute
               specified on an  &lt;Insert&gt;, &lt;Update&gt;
               or &lt;Delete&gt; action.  If a value is not
               specified for the handle attribute then a WFS
               may employ some other means of locating the
               action.  For example, the value of the locator
               attribute may be an integer indicating the order
               of the action (i.e. 1=First action, 2=Second action,
               etc.).
            </xsd:documentation>
         </xsd:annotation>
      </xsd:attribute>
      <xsd:attribute name="code" type="xsd:string" use="optional">
         <xsd:annotation>
            <xsd:documentation>
               The code attribute may be used to specify an
               exception code indicating why an action failed.
            </xsd:documentation>
         </xsd:annotation>
      </xsd:attribute>
```



```
1794    </xsd:complexType>
1795    <xsd:complexType name="InsertResultType">
1796       <xsd:annotation>
1797          <xsd:documentation xml:lang="en">
1798             Reports the list of identifiers of all features created
1799             by a transaction request.  New features are created using
1800             the Insert action and the list of idetifiers must be
1801             presented in the same order as the Insert actions were
1802             encountered in the transaction request.  Features may
1803             optionally be correlated with identifiers using the
1804             handle attribute (if it was specified on the Insert
1805             element).
1806          </xsd:documentation>
1807       </xsd:annotation>
1808       <xsd:sequence>
1809          <xsd:element name="Feature"
1810                       type="wfs:InsertedFeatureType"
1811                       maxOccurs="unbounded"/>
1812       </xsd:sequence>
1813    </xsd:complexType>
1814    <xsd:complexType name="InsertedFeatureType">
1815       <xsd:sequence>
1816          <xsd:element ref="ogc:FeatureId" maxOccurs="unbounded">
1817             <xsd:annotation>
1818                <xsd:documentation xml:lang="en">
1819                   This is the feature identifier for the newly created
1820                   feature.  The feature identifier may be generated by
1821                   the WFS or provided by the client (depending on the
1822                   value of the idgen attribute).  In all cases of idgen
1823                   values, the feature id must be reported here.
1824                </xsd:documentation>
1825             </xsd:annotation>
1826          </xsd:element>
1827       </xsd:sequence>
1828       <xsd:attribute name="handle" type="xsd:string" use="optional">
1829          <xsd:annotation>
1830             <xsd:documentation xml:lang="en">
1831                If the insert element that generated this feature
1832                had a value for the "handle" attribute then a WFS
1833                may report it using this attribute to correlate
1834                the feature created with the action that created it.
1835             </xsd:documentation>
1836          </xsd:annotation>
1837       </xsd:attribute>
1838    </xsd:complexType>
1839 </xsd:schema>
```

