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

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved. To obtain additional rights of use, visit http://www.opengeospatial.org/legal/.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>i. Preface</td>
<td>iii</td>
</tr>
<tr>
<td>ii. Document terms and definitions</td>
<td>iii</td>
</tr>
<tr>
<td>iii. Document contributor contact points</td>
<td>iii</td>
</tr>
<tr>
<td>iv. Revision history</td>
<td>iii</td>
</tr>
<tr>
<td>v. Changes to OGC Specifications</td>
<td>iv</td>
</tr>
<tr>
<td>Foreword</td>
<td>v</td>
</tr>
<tr>
<td>Introduction</td>
<td>vi</td>
</tr>
<tr>
<td>1. Scope</td>
<td>1</td>
</tr>
<tr>
<td>2. Normative references</td>
<td>1</td>
</tr>
<tr>
<td>3. Corrigendum Description</td>
<td>1</td>
</tr>
<tr>
<td>3.1 Changes to ANNEX A of 04-094</td>
<td>1</td>
</tr>
<tr>
<td>3.2 Changes to the XML Schema wfs.xsd</td>
<td>2</td>
</tr>
<tr>
<td>ANNEX A Annotated XML Schema wfs.xsd distributed with 04-094</td>
<td>4</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kammersell, William</td>
<td>BBNT Solutions LLC</td>
</tr>
<tr>
<td>Lansing, Jeff</td>
<td>SYS Technologies Inc.</td>
</tr>
</tbody>
</table>

iv.  Revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Release</th>
<th>Editor</th>
<th>Primary clauses modified</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006-02-12</td>
<td>1.0.0</td>
<td>Panagiotis (Peter) A. Vretanos</td>
<td>wfs.xsd</td>
<td>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.</td>
</tr>
</tbody>
</table>
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
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:

<table>
<thead>
<tr>
<th>Line #</th>
<th>O</th>
<th>Original Text</th>
<th>Replacement Text</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>1795</td>
<td>C</td>
<td><code>&lt;xsd:complexType name=&quot;InsertResultType&quot;&gt;</code></td>
<td><code>&lt;xsd:complexType name=&quot;InsertResultsType&quot;&gt;</code></td>
<td>The element is named &quot;InsertResults&quot; so the type should be renamed to match.</td>
</tr>
<tr>
<td>1692</td>
<td>C</td>
<td><code>Type=&quot;wfs:InsertResultType&quot;&gt;</code></td>
<td><code>type=&quot;wfs:InsertResultsType&quot;&gt;</code></td>
<td>The element is named &quot;InsertResults&quot; so the type should be renamed to match.</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Line #</th>
<th>O</th>
<th>Original Text</th>
<th>Replacement Test</th>
<th>Reason</th>
</tr>
</thead>
</table>
| 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:extension>` | `<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:simpleContent>` | | See line 99 |
| 99     | A | `</xsd:simpleContent>` | | 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="PropertyName" type="xsd:QName"/>` | `<xsd:element name="PropertyName" type="xsd:string">`  
`<xsd:annotation>`  
`<xsd:documentation>`  
The Property element is in the first place  
`</xsd:documentation>`  
`</xsd:annotation>`  
`</xsd:element>` | Fix spelling error PropertyName 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

<?xml version="1.0"?><xsd:schema
targetNamespace="http://www.opengis.net/wfs"
xmlns:wfs="http://www.opengis.net/wfs"
xmlns:ogc="http://www.opengis.net/ogc"
xmlns:ows="http://www.opengis.net/ows"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" version="1.1.0">

<!-- -------------------------------------------------------------
Includes and Imports
------------------------------------------------------------- -->
<xsd:import namespace="http://www.opengis.net/gml"
schemaLocation="../../gml/3.1.1/base/gml.xsd"/>
<xsd:import namespace="http://www.opengis.net/ogc"
schemaLocation="../../filter/1.1.0/filter.xsd"/>
<xsd:import namespace="http://www.opengis.net/ows"
schemaLocation="../../ows/1.0.0/owsGetCapabilities.xsd"/>

<!-- -------------------------------------------------------------
= BASE REQUEST TYPE =
------------------------------------------------------------- -->
<xsd:complexType name="BaseRequestType" abstract="true">
  <xsd:annotation>
    <xsd:documentation>
      XML encoded WFS operation request base, for all operations except GetCapabilities.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:attribute name="service" type="ows:ServiceType"
    use="optional" default="WFS">
    <xsd:annotation>
      <xsd:documentation>
        The service attribute is included to support service endpoints that implement more than one OGC service.
        For example, a single CGI that implements WMS, WFS and WCS services.
        The endpoint can inspect the value of this attribute to figure out which service should process the request.
        The value WFS indicates that a web feature service should process the request.
        This parameter is somewhat redundant in the XML encoding since the request namespace can be used to determine which service should process any given request. For example, wfs:GetCapabilities and easily be distinguished from wcs:GetCapabilities using the namespaces.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>

  <xsd:attribute name="version" type="xsd:string"
    use="optional" default="1.1.0">
    <xsd:annotation>
      <xsd:documentation>
        The version attribute is used to indicate the version of the WFS specification that a request conforms to. All requests in this schema conform to V1.1 of the WFS specification. For WFS implementations that support more than one version of a WFS specification ... if the version attribute is not specified then the service should assume that the request conforms to greatest available specification version.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
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).

This element may be used in place of an \texttt{wfs:PropertyName} element in a \texttt{wfs:Query} element in a \texttt{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 -- \texttt{GetFeatureWithLock}, \texttt{LockFeature}, \texttt{Insert}, \texttt{Update}, \texttt{Delete} -- in this version of the WFS specification.

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 ".*".

The \texttt{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.

A WFS may reject the traversal of nested XLinks in the returned element for a named property if \texttt{traverseXLinkDepth} is not positive integer plus "*".
<xsd:complexType name="GetCapabilitiesType">
  <xsd:annotation>
    <xsd:documentation>
      Request to a WFS to perform the GetCapabilities operation. This operation allows a client to retrieve a Capabilities XML document providing metadata for the specific WFS server.
      The GetCapabilities element is used to request that a Web Feature Service generate an XML document describing the organization providing the service, the WFS operations that the service supports, a list of feature types that the service can operate on and list of filtering capabilities that the service support. Such an XML document is called a capabilities document.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="ows:GetCapabilitiesType">
      <xsd:attribute name="service" type="ows:ServiceType" use="optional" default="WFS"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<!-- RESPONSE -->
<xsd:element name="WFS_Capabilities" type="wfs:WFS_CapabilitiesType" substitutionGroup="ows:Capabilities"/>
<xsd:complexType name="WFS_CapabilitiesType">
  <xsd:annotation>
    <xsd:documentation>
      XML encoded WFS GetCapabilities operation response. This document provides clients with service metadata about a specific service instance, including metadata about the tightly-coupled data served. If the server does not implement the updateSequence parameter, the server shall always return the complete Capabilities document, without the updateSequence parameter. When the server implements the updateSequence parameter and the GetCapabilities operation request included the updateSequence parameter with the current value, the server shall return this element with only the "version" and "updateSequence" attributes. Otherwise, all optional elements shall be included or not depending on the actual value of the Contents parameter in the GetCapabilities operation request.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="ows:CapabilitiesBaseType">
      <xsd:sequence>
        <xsd:element ref="wfs:FeatureTypeList" minOccurs="0"/>
        <xsd:element ref="wfs:ServesGMLObjectTypeList" minOccurs="0"/>
        <xsd:element ref="wfs:SupportsGMLObjectTypeList" minOccurs="0"/>
        <xsd:element ref="ogc:Filter_Capabilities"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="FeatureTypeList" type="wfs:FeatureTypeListType"/>
<xsd:complexType name="FeatureTypeListType">
  <xsd:annotation>
    <xsd:documentation>
      A list of feature types available from this server.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="Operations" type="wfs:OperationsType" minOccurs="0"/>
    <xsd:element name="FeatureType" type="wfs:FeatureTypeType" maxOccurs="unbounded"/>
  </xsd:sequence>
</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
              reference[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:choice>
  </xsd:sequence>
  <xsd:element name="Operations"

<xsd:complexType name="OperationsType">
  <xsd:sequence>
    <xsd:element name="Operation" type="wfs:OperationType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="OperationType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Insert"/>
    <xsd:enumeration value="Unsert"/>
    <xsd:enumeration value="Delete"/>
    <xsd:enumeration value="Query"/>
    <xsd:enumeration value="Lock"/>
    <xsd:enumeration value="GetGmlObject"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="OutputFormatListType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="Format" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MetadataURLType">
  <xsd:annotation>
    <xsd:documentation>
      A Web Feature Server MAY use zero or more MetadataURL elements to offer detailed, standardized metadata about the data underneath a particular feature type. The type attribute indicates the standard to which the metadata complies; the format attribute indicates how the metadata is structured. Two types are defined at present: 'TC211' or 'ISO19115' = ISO TC211 19115; 'FGDC' = FGDC CSDGM; 'ISO19139' = ISO 19139
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="type" use="required"/>
      <xsd:attribute name="format" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="GMLObjectTypeListType">
  <xsd:sequence>
    <xsd:element name="GMLObjectType" type="wfs:GMLObjectTypeType" maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>
          Name of this GML object type, including any namespace prefix
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="GMLObjectTypeType">
  <xsd:annotation>
    <xsd:documentation>
      An element of this type that describes a GML object 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>
        Name of this GML Object type, including any namespace prefix.
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Title" type="xsd:string" minOccurs="0">
      <xsd:annotation>
        Title of this GML Object type, normally used for display to a human.
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Abstract" type="xsd:string" minOccurs="0">
      <xsd:annotation>
        Brief narrative description of this GML Object type, normally used for display to a human.
      </xsd:annotation>
    </xsd:element>
    <xsd:element ref="ows:Keywords" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="OutputFormats" type="wfs:OutputFormatListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<!-- ================================================================== -->
<!-- =   DESCRIBEFEATURETYPE Request and Response                     = -->

<!-- !--  ********************************************************************************-->
<!-- !--  DESCRIBEFEATURETYPE Request and Response                        -->
<!-- !--  ********************************************************************************-->

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.
<xsd:element name="DescribeFeatureType" type="wfs:DescribeFeatureTypeType">
  <xsd:documentation>
    The DescribeFeatureType element is used to request that a Web Feature Service generate a document describing one or more feature types.
  </xsd:documentation>
</xsd:element>

<xsd:complexType name="DescribeFeatureTypeType">
  <xsd:annotation>
    <xsd:documentation>
      The DescribeFeatureType operation allows a client application to request that a Web Feature Service describe one or more feature types. A Web Feature Service must be able to generate feature descriptions as valid GML3 application schemas.
      The schemas generated by the DescribeFeatureType operation can be used by a client application to validate the output.
      Feature instances within the WFS interface must be specified using GML3. The schema of feature instances specified within the WFS interface must validate against the feature schemas generated by the DescribeFeatureType request.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="wfs:BaseRequestType">
      <xsd:sequence>
        <xsd:element name="TypeName" type="xsd:QName" minOccurs="0" maxOccurs="unbounded">
          <xsd:annotation>
            <xsd:documentation>
              The TypeName element is used to enumerate the feature types to be described. If no TypeName elements are specified then all features should be described. The name must be a valid type that belongs to the feature content as defined by the GML Application Schema.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="outputFormat" type="xsd:string" use="optional" default="text/xml; subtype=gml/3.1.1">
        <xsd:annotation>
          <xsd:documentation>
            The outputFormat attribute is used to specify what schema description language should be used to describe features. The default value of 'text/xml; subtype=3.1.1' means that the WFS must generate a GML3 application schema that can be used to validate the GML3 output of a GetFeature request or feature instances specified in Transaction operations. For the purposes of experimentation, vendor extension, or even extensions that serve a specific community of interest, other acceptable output format values may be advertised by a WFS service in the capabilities document. The meaning of such values in not defined in the WFS specification. The only proviso is such cases is that clients may safely ignore outputFormat values that do not recognize.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
For the outputFormat value of 'text/xml; subtype=gml/3.1.1' a WFS must generate a valid XML-Schema/GML3 application schema that describes the requested feature type(s).
used to specify other formats as long as those values
are advertised in the capabilities document.
For example, the value WKB may be used to indicate that a
Well Known Binary format be used to encode the output.
</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="maxFeatures"
    type="xsd:positiveInteger" use="optional">
    <xsd:annotation>
        <xsd:documentation>
The maxFeatures attribute is used to specify the maximum
number of features that a GetFeature operation should
generate (regardless of the actual number of query hits).
</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
<xsd:attribute name="traverseXlinkDepth"
    type="xsd:string" use="optional">
    <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 ")*".
If this attribute is not specified then no xlinks shall be
resolved and the value of traverseXlinkExpiry attribute (if
it specified) may be ignored.
</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.
This attribute is only relevant if a value is specified
for the traverseXlinkDepth attribute.
</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name="ResultTypeType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="results">
            <xsd:annotation>
                <xsd:documentation>
Indicates that a complete response should be generated
by the WFS. That is, all response feature instances
should be returned to the client.
</xsd:documentation>
</xsd:annotation>
</xsd:enumeration>
        <xsd:enumeration value="hits">
            <xsd:annotation>
                <xsd:documentation>
Indicates that an empty response should be generated with
the "numberOfFeatures" attribute set (i.e. no feature

instances should be returned). In this manner a client may
determine the number of feature instances that a GetFeature
request will return without having to actually get the
total result set back.

</xsd:documentation>
</xsd:annotation>
</xsd:complexType>
<xsd:element name="Query" type="wfs:QueryType">
  <xsd:annotation>
    <xsd:documentation>
      The Query element is used to describe a single query.
      One or more Query elements can be specified inside a
      GetFeature element so that multiple queries can be
      executed in one request. The output from the various
      queries are combined in a wfs:FeatureCollection element
      to form the response document.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="QueryType">
  <xsd:annotation>
    <xsd:documentation>
      The Query element is of type QueryType.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element ref="wfs:PropertyName">
        <xsd:annotation>
          <xsd:documentation>
            The Property element is used to specify one or more
            properties of a feature whose values are to be retrieved
            by a Web Feature Service.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element ref="ogc:Function">
        <xsd:annotation>
          <xsd:documentation>
            A function may be used as a select item in a query.
            However, if a function is used, care must be taken
            to ensure that the result type matches the type in the
            request. If they are not, a Web Feature Service
            will add them automatically to the Query before processing
            it. Thus a client application should, in general, be
            prepared to receive more properties than it requested.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:choice>
    <xsd:element ref="ogc:Filter" minOccurs="0" maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>
          The Filter element is used to define spatial and/or non-spatial
          constraints on query. Spatial constrains use GML3 to specify
          the constraining geometry. A full description of the Filter
          element can be found in the Filter Encoding Implementation
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
The SortBy element is used specify property names whose values should be used to order (upon presentation) the set of feature instances that satisfy the query.

The handle attribute allows a client application to assign a client-generated identifier for the Query. The handle is included to facilitate error reporting. If one Query in a GetFeature request causes an exception, a WFS may report the handle to indicate which query element failed. If the a handle is not present, the WFS may use other means to localize the error (e.g. line numbers).

For systems that implement versioning, the featureVersion attribute is used to specify which version of a particular feature instance is to be retrieved. A value of ALL means that all versions should be retrieved. An integer value 'i', means that the ith version should be retrieve if it exists or the most recent version otherwise.

This attribute is used to specify a specific WFS-supported SRS that should be used for returned feature geometries. The value may be the WFS StorageSRS value, DefaultRetrievalSRS value, or one of AdditionalSRS values. If no srsName value is supplied, then the features will be returned using either the DefaultRetrievalSRS, if specified, and StorageSRS otherwise. For feature types with no spatial properties, this attribute must not be specified or ignored if it is specified.
<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:
        </xsd:documentation>
        <xsd:documentation>
          typeName="ns1:Inwatera_1m=A, ns2:CoastL_1M=B"
        </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>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="GetGmlObjectType">
  <xsd:annotation>
    <xsd:documentation>
      A GetGmlObjectType element contains exactly one GmlObjectId.
      The value of the gml:id attribute on that GmlObjectId is used
      as a unique key to retrieve the complex element with a
      gml:id attribute with the same value.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="wfs:BaseRequestType">
      <xsd:sequence>
        <xsd:element ref="ogc:GmlObjectId"/>
      </xsd:sequence>
      <xsd:attribute name="outputFormat" type="xsd:string" use="optional" default="GML3"/>
      <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 "$" 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:extension>
  </xsd:complexContent>
</xsd:complexType>

The response to a GetGMLObject request is a GML3 fragment(s) that
This is the root element for the GetFeatureWithLock request.
The GetFeatureWithLock operation performs identically to a
GetFeature request except that the GetFeatureWithLock request
locks all the feature instances in the result set and returns
a lock identifier to a client application in the response.
The lock identifier is returned to the client application
using the lockId attribute defined on the wfs:FeatureCollection
element.
<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:complexType 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 operation.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:attribute>
    </xsd:complexType>
</xsd:element>
After the expiry period elapses, the locked resources must be released. If the expiry attribute is not set, then the default value of 5 minutes will be enforced.

The lockAction attribute is used to indicate what a Web Feature Service should do when it encounters a feature instance that has already been locked by another client application.

Valid values are ALL or SOME.

ALL means that the Web Feature Service must acquire locks on all the requested feature instances. If it cannot acquire those locks then the request should fail. In this instance, all locks acquired by the operation should be released.

SOME means that the Web Feature Service should lock as many of the requested features as it can.

This type defines the Lock element. The Lock element defines a locking operation on feature instances of a single type. An OGC Filter is used to constrain the scope of the operation. Features to be locked can be identified individually by using their feature identifier or they can be locked by satisfying the spatial and non-spatial constraints defined in the filter.

The handle attribute allows a client application to assign a client-generated request identifier to a Lock action. The handle is included to facilitate error reporting. If one of a set of Lock actions failed while processing a LockFeature request, a WFS may report the handle in an exception report to localize the error. If a handle is not present then a WFS may employ some other means of localizing the error (e.g. line number).
The value of the typeName attribute is the name of the feature type to be updated. The name specified must be a valid type that belongs to the feature content as defined by the GML Application Schema.

</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>

<!-- RESPONSE -->

<xsd:element name="LockFeatureResponse" type="wfs:LockFeatureResponseType">
  <xsd:annotation>
    <xsd:documentation>
      The LockFeatureResponse element contains a report about the completion status of a LockFeature request.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="LockFeatureResponseType">
  <xsd:annotation>
    <xsd:documentation>
      The LockFeatureResponseType is used to define an element to contains the response to a LockFeature operation.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="wfs:LockId">
      <xsd:annotation>
        <xsd:documentation>
          The LockFeatureResponse includes a LockId element that contains a lock identifier. The lock identifier can be used by a client, in subsequent operations, to operate upon the locked feature instances.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="FeaturesLocked" type="wfs:FeaturesLockedType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
          The LockFeature or GetFeatureWithLock operations identify and attempt to lock a set of feature instances that satisfy the constraints specified in the request. In the event that the lockAction attribute (on the LockFeature or GetFeatureWithLock elements) is set to SOME, a Web Feature Service will attempt to lock as many of the feature instances from the result set as possible.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="FeaturesNotLocked" type="wfs:FeaturesNotLockedType" minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>
          In contrast to the FeaturesLocked element, the FeaturesNotLocked element contains a list of ogc:Filter elements identifying feature instances that a WFS actually managed to lock.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:sequence maxOccurs="unbounded">
  <xsd:element ref="ogc:FeatureId"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FeaturesNotLockedType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element ref="ogc:FeatureId"/>
  </xsd:sequence>
</xsd:complexType>

<!-- ================================================================== -->
<!-- =   TRANSACTION Request and Response                             = -->
<!-- ================================================================== -->

<!-- REQUEST -->
<xsd:element name="Transaction" type="wfs:TransactionType">
  <xsd:annotation>
    <xsd:documentation>
      This is the root element for a Transaction request. A transaction request allows insert, update and delete operations to be performed to create, change or remove feature instances.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>

<xsd:complexType name="TransactionType">
  <xsd:annotation>
    <xsd:documentation>
      The TransactionType defines the Transaction operation. A Transaction element contains one or more Insert, Update Delete and Native elements that allow a client application to create, modify or remove feature instances from the feature repository that a Web Feature Service controls.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="ows:GetCapabilitiesType">
      <xsd:sequence>
        <xsd:element ref="wfs:LockId" minOccurs="0">
          <xsd:annotation>
            <xsd:documentation>
              In order for a client application to operate upon locked feature instances, the Transaction request must include the LockId element. The content of this element must be the lock identifier the client application obtained from a previous GetFeatureWithLock or LockFeature operation.
            </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element ref="wfs:Insert"/>
          <xsd:element ref="wfs:Update"/>
          <xsd:element ref="wfs:Delete"/>
          <xsd:element ref="wfs:Native"/>
        </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="releaseAction" type="wfs:AllSomeType" use="optional">
        <xsd:annotation>
          <xsd:documentation>
            The releaseAction attribute is used to control how a Web Feature service releases locks on feature instances after a Transaction request has been processed.
          </xsd:documentation>
        </xsd:annotation>
      </xsd:attribute>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
A value of ALL means that the Web Feature Service should release the locks of all feature instances locked with the specified lockId regardless or whether or not the features were actually modified.

A value of SOME means that the Web Feature Service will only release the locks held on feature instances that were actually operated upon by the transaction. The lockId that the client application obtained shall remain valid and the other, unmodified, feature instances shall remain locked.

If the expiry attribute was specified in the original operation that locked the feature instances, then the expiry counter will be reset to give the client application that same amount of time to post subsequent transactions against the locked features.

The LockId element contains the value of the lock identifier obtained by a client application from a previous GetFeatureWithLock or LockFeature request.

The Insert element is used to indicate that the Web Feature Service should create a new instance of a feature type. The feature instance is specified using GML3 and one or more feature instances to be created can be contained inside the Insert element.

An Insert element may contain a feature collection or one or more feature instances to be inserted into the repository.

The idgen attribute control how a WFS generates identifiers from newly created feature instances using the Insert action. The default action is to have the WFS generate a new id for the features. This is also backward compatible with WFS 1.0 where the only action was for the WFS to generate an new id.
<xsd:annotation>
  <xsd:documentation>
    The handle attribute allows a client application
to assign a client-generated request identifier
to an Insert action. The handle is included to
facilitate error reporting. If an Insert action
in a Transaction request fails, then a WFS may
include the handle in an exception report to localize
the error. If no handle is included of the offending
Insert element then a WFS may employ other means of
localizing the error (e.g. line number).
  </xsd:documentation>
</xsd:annotation>

<xsd:attribute name="inputFormat" type="xsd:string"
  use="optional" default="text/xml; subtype=gml/3.1.1">
  <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>
      PAV 12NOV2004
      WHY IS THIS HERE? WOULDN'T WE KNOW THE INCOMING SRS FROM THE
      GML GEOMETRY ELEMENTS? I ASSUME THAT IF THE INCOMING SRS
      DOES NOT MATCH ONE OF THE STORAGE SRS(s) THEN THE WFS WOULD
      EITHER PROJECT INTO THE STORAGE SRS OR RAISE AN EXCEPTION.
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>

<xsd:complexType name="IdentifierGenerationOptionType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="UseExisting">
      <xsd:annotation>
        <xsd:documentation>
          The UseExisting value indicates that WFS should not
generate a new feature identifier for the feature
being inserted into the repository. Instead, the WFS
should use the identifier encoded if the feature.
If a duplicate exists then the WFS should raise an
exception.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="ReplaceDuplicate">
      <xsd:annotation>
        <xsd:documentation>
          The ReplaceDuplicate value indicates that WFS should not
generate a new feature identifier for the feature
being inserted into the repository. Instead, the WFS
should use the identifier encoded if the feature.
If a duplicate exists then the WFS should replace the
existing feature instance with the one encoded in the
Insert action.
        </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="GenerateNew">
      <xsd:annotation>
        <xsd:documentation>
        </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
  </xsd:restriction>
</xsd:complexType>
The GenerateNew value indicates that WFS should generate a new unique feature identifier for the feature being inserted into the repository.

One or more existing feature instances can be changed by using the Update element. Changing or updating a feature instance means that the current value of one or more properties of the feature are replaced with new values. The Update element contains one or more Property elements. A Property element contains the name or a feature property whose value is to be changed and the replacement value for that property.

The Filter element is used to constrain the scope of the update operation to those features identified by the filter. Feature instances can be specified explicitly and individually using the identifier of each feature instance OR a set of features to be operated on can be identified by specifying spatial and non-spatial constraints in the filter. If no filter is specified then update operation applies to all feature instances.

The handle attribute allows a client application to assign a client-generated request identifier to an Insert action. The handle is included to facilitate error reporting. If an Update action in a Transaction request fails, then a WFS may include the handle in an exception report to localize the error. If no handle is included of the offending Insert element then a WFS may employ other means of localizing the error (e.g. line number).

The value of the typeName attribute is the name of the feature type to be updated. The name must be a valid type that belongs to the feature content as defined by the GML Application Schema.
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.
each feature instance OR a set of features to be
operated on can be identified by specifying spatial
and non-spatial constraints in the filter.
If no filter is specified then an exception should
be raised since it is unlikely that a client application
intends to delete all feature instances.

</xsd:annotation>
</xsd:element>
</xsd:sequence>
<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 an Insert action. The handle is included to
facilitate error reporting. If a Delete action
in a Transaction request fails, then a WFS may
include the handle in an exception report to localize
the error. If no handle is included of the offending
Insert element then a WFS may employee other means of
localizing the error (e.g. line number).
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
<xsd:attribute name="typeName" type="xsd:QName" use="required">
  <xsd:annotation>
    <xsd:documentation>
      The value of the typeName attribute is the name
      of the feature type to be updated. The name
      specified must be a valid type that belongs to
      the feature content as defined by the GML
      Application Schema.
    </xsd:documentation>
  </xsd:annotation>
</xsd:attribute>
</xsd:complexType>
<xsd:element name="Native" type="wfs:NativeType">
  <xsd:annotation>
    <xsd:documentation>
      Many times, a Web Feature Service interacts with a repository
      that may have special vendor specific capabilities. The native
      element allows vendor specific command to be passed to the
      repository via the Web Feature Service.
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:complexType name="NativeType">
  <xsd:attribute name="vendorId" type="xsd:string" use="required">
    <xsd:annotation>
      <xsd:documentation>
        The vendorId attribute is used to specify the name of
        vendor who's vendor specific command the client
        application wishes to execute.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
  <xsd:attribute name="safeToIgnore" type="xsd:boolean" use="required">
    <xsd:annotation>
      <xsd:documentation>
        In the event that a Web Feature Service does not recognize
        the vendorId or does not recognize the vendor specific command,
        the safeToIgnore attribute is used to indicate whether the
        exception can be safely ignored. A value of TRUE means that
        the Web Feature Service may ignore the command. A value of
        FALSE means that a Web Feature Service cannot ignore the
        command and an exception should be raised if a problem is
        encountered.
      </xsd:documentation>
    </xsd:annotation>
  </xsd:attribute>
</xsd:complexType>
The TransactionResponse element contains a report about the completion status of a Transaction operation.

The response for a transaction request that was successfully completed. If the transaction failed for any reason, an exception report is returned instead.

The TransactionSummary element is used to summarize the number of feature instances affected by the transaction.

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.

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: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>
</xsd:complexType>
<xsd:complexType name="InsertResultType">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
            Reports the list of identifiers of all features created by a transaction request. New features are created using the Insert action and the list of identifiers must be presented in the same order as the Insert actions were encountered in the transaction request. Features may optionally be correlated with identifiers using the handle attribute (if it was specified on the Insert element).
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="Feature" type="wfs:InsertedFeatureType" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="InsertedFeatureType">
    <xsd:sequence>
        <xsd:element ref="ogc:FeatureId" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="handle" type="xsd:string" use="optional">
        <xsd:annotation>
            <xsd:documentation xml:lang="en">
                This is the feature identifier for the newly created feature. The feature identifier may be generated by the WFS or provided by the client (depending on the value of the idgen attribute). In all cases of idgen values, the feature id must be reported here.
            </xsd:documentation>
        </xsd:annotation>
    </xsd:attribute>
</xsd:complexType>