OpenGIS Consortium Discussion Paper 01-036

" Gazetteer Service Draft Candidate Implementation Specification 0.84"

This paper presents a discussion of technology issues considered in a Special Interest Group of the Open GIS Consortium Technical Committee. The content of this paper is presented to create discussion in the geospatial information industry on this topic; the content of this paper is not to be considered an adopted specification of any kind. This paper does not represent the official position of the Open GIS Consortium nor of the OGC Technical Committee.

TITLE: Gazetteer Service Specification

DOCUMENT: 01-036

VERSION 0.84

DATE: *March 15, 2001*

TYPE: OGC-IP Draft Candidate Implementation Specification, Discussion Paper

This version:

http://feature.opengis.org/members/archive/arch01/01-036.pdf

Latest version:

http://feature.opengis.org/members/archive/arch01/01-036.pdf

Editor:

Rob Atkinson, Social Change Online rob@socialchange.net.au

Contributors:

Jeff Lansing, DTAI jeff@polexis.com
Benoit Borlee, IONIC Software, benoit.borlee@ionicsoft.com
Marwa Mabrouck, ESRI
Serge Margoulies, IONIC Software, serge.margoulies@ionicsoft.com
Ron Lake, Galdos, rlake@galdosinc.com
Milan Trninic, Galdos, mtrninic@galdosinc.com
Bernard Snyers, IONIC Software, bernard.snyers@ionicsoft.com
John Davidson, OGC IP Team Architect, georef@erols.com
Harry Niedzwiadek, OGC IP Team Architect, harry1@erols.com
Carl Reed, OGC IP Team Architect, creediii@mindpsring.com
David Danko, US NIMA
Paul Daisey, US Census
Simon Cox, CSIRO Minerals and Exploration

Copyright © 2001 OGC® (Social Change Online, Ionic Software), All Rights Reserved. OGC liability, trademark, document use and software licensing rules apply.

Abstract

This document describes the OGC interfaces for a network-accessible Gazetteer Service.

Status of this document

This document is a draft Candidate Implementation for OGC Gazetteer Service. It represents “work in progress” and should be treated accordingly. This version of the specification supersedes all previous Gazetteer Service documents.

The revision history is summarized below:

- 0.84 – update interfaces (request syntax and response schema) using XML Schema as revised for GFSPP; minor formatting and wordsmithing edits
- 0.8 – GML-2 compliance, harmonization of terminology with ISO 19112, examples, etc…
- 0.7 – first draft of RFC
- 0.6.2 – version that included the combined specifications for Gazetteer and Geocoder Services, as implemented for the GFS Testbed, completed on November 17, 2000
- Earlier versions were draft candidate specifications leading up to 0.6.2

Change key:

Editorial Comments

ED: [Editorial Notes are inserted in RED, as displayed here, wherever needed.]

Issues

All issues, and applicable resolutions, are documented inline. Please use the format below as a guideline for documenting issues.

Issue Name: [Issue Name in RED, YELLOW, or GREEN based upon criticality of the issue, with Red being highest priority. (Initials, Date)]

Issue Description: [Issue Description.]

Resolution: [Insert Resolution Details and History. (Initials, Date)]

Table of contents

1. Introduction

A **Gazetteer Service** is a network-accessible service that retrieves the known geometries for one or more features, given their associated well-known feature identifiers (text strings), which are specified at run-time through a query (filter) request. The identifiers are any words or terms that describe the features, which are well known to the Gazetteer Service, such as a set of place names and/or landmarks. Each instance of a Gazetteer Service has an associated vocabulary of identifiers. Thus, a Gazetteer Service may apply to a given region, such as a country, or some other specialized grouping of features. The identifiers used in a query request may be determined in any of a number of ways, such as by a Geoparser Service [Ref5]. The returned geometries are expressed in an OGC Spatial Reference System [Ref6].

A **Gazetteer Service** may be thought of as a special case of a Geocoder Service [Ref4] that is an authority for place names and their associated geometries. This is expected to be consistent with the definition of a gazetteer in ISO CD 19112 – “Geographic Information – Spatial referencing by geographic identifiers” [add reference here]. This specification adopts a data model based on CD 19112, with derivation and mappings explicitly noted.

This interface specification for Gazetteer Service is harmonized with the syntax and semantics of other proposed OGC interfaces, namely Web Feature Server [Ref1] and Geocoder Service [Ref4]. This is expected to be formalized at some stage through inheritance from a common “Basic Services Model” (BSM) which will provide common abstract schemas for OGC Web Services. Until that time this specification is intended to be complete through use of explicit references to published documents. Later revisions may include new references to common specifications and remove redundant material.

A Gazetteer Service exposes both a potentially structured vocabulary and a feature collection, and thus may support, with the same data structure, all the capabilities of a hierarchical vocabulary and a Web Feature Server (WFS). However, this version of the specification is limited to the latter case. Thus, it only applies to Gazetteer Services that deliver features that instantiate a well-defined OGC GML schema [Ref2], in response to any set of geographical queries, as defined in the OGC Filter Encoding Spec [Ref3]. It is the responsibility of the service to advertise which geographical queries it supports.

1.1. Use Cases (Informative)

Navigation Use Case

In this use case, a viewer client application wishes to give the user the ability to navigate through a spatial resource using place names. This can be done by either allowing structured searches or by browsing a tree of terms to narrow down search requirements. We support the assertion that the tree is actually the general case, and that searches return trees of varying complexity. A simple Gazetteer Service can return an unstructured list that is either the child nodes of a query term, or possibly a complete list of nodes generated by walking the tree. In the trivial case, the tree depth is one and there is no hierarchy. In this case, terms are presented in a tree with arbitrary depth. The user can then select the geographic feature of interest for a variety of actions, for example:

1. Zoom a map or otherwise run a query relating to the feature;
2. Refine navigation choices with a new list of features within the feature specified; and

3. Focus a locator map on a new scale, to show position of bottom most (target) term.

The interface would either build such a tree through recursive calls to the Gazetteer Service, driven by type schema information, or be able to extract a fully formed tree from a single call. (Type schema information may be interface specific, externally defined or derived from the hierarchy of terms returned.)

Geocoding Use Case

This Use Case pertains to a Geocoder Service that invokes a Gazetteer Service to extract a particular set of geometries that pertain to a given set of identifiers, which are derived from arbitrary resources. For example, these identifiers may have been obtained through a Geoparser Service.

The use of a network-accessible Gazetteer Service by a Geocoder Service may have the following advantages:

- The target vocabulary might be huge
- The target vocabulary may be highly specialized
- The target vocabulary may be partitioned (e.g. use Australian Gazetteer Service to get Australian place names)
- The geometry of the Gazetteer Service may be inconvenient for the Geocoder Service to manage internally (e.g. size, multiple equivalents, on-the-fly generalization and re-projection required)

The semantics of the Geocoder Service may require normalization of terms based on rules not available to the service (e.g., a deeply nested hierarchy

Other uses may include:

- Acquisition of a vocabulary for re-use (i.e. a generic Geoparser Service is pointed at documents containing names of water features in Uzbekhistan – it might load the relevant Gazetteer Service out-of-band and cache it for some period).
- A set of terms may be sent as a batch and a single response retrieved for improved efficiency. (Note this may simply replace the need for a Geocoder Service to remember common terms – although its own encoding of the geocoded output should re-use features.)

Geoparsing Use Case

A Geoparser Service seeks candidate terms that it will employ in a text search through some mechanism such as pattern matching or look-up in a well-known vocabulary. The target vocabulary might well be associated with a Gazetteer Service. Accordingly, a Geoparser Service may reasonably be expected to call upon specific instances of Gazetteer Services, with well-known vocabularies, as required to make use of these vocabularies.

Service Invocation Use Case

A client or agent application performing service chaining may wish to exploit a Gazetteer Service to broaden particular search queries at run-time. For example, if the application is looking for information relevant to a particular suburb, it may wish to invoke a Gazetteer Service to find the city that contains the suburb in order to invoke a service organized by city names.

Consider the case of a tool to help locate relevant information – given a term, it might search for all synonyms of the term, thus making the term broader until a suitable result set is found. In this way, today's rather primitive Web search collections could possibly be exploited to harvest related content.

There may be no a-priori knowledge of a desired Gazetteer Service. This knowledge may be derived at run time according to other metadata from the source feature or document. Accordingly, the client should not need to store all possible instances of Gazetteer Services, but rather needs to be able to discover them at run time, as needed.

1.2. Architectural Constraints (Informative)

The following requirements were identified from the Use Cases for Gazetteer Services.

Feature Types

For a service to be trusted as a Gazetteer it must provide Feature Types derived by extension from the well-known Feature Type "`GazetteerEntryType`".

The base schema defines the naming, semantics and allowable geometric properties that a `GazetteerEntryType` must have. Derived schemas allow additional properties to be stored and returned without compromising the ability of clients to understand the elements in the base schema.

The base `GazetteerEntryType` also defines the syntax of a mechanism for describing relationships between gazetteer entries. These are typically parent-child relationships as per the ISO CD 19112 data model.

Web Feature Server compatibility

By defining the interfaces for a Gazetteer Service to be syntactically and semantically similar with the interfaces for WFS, it is possible to then use a WFS instance, and its associated databases of features, as the underlying implementation for a general purpose Gazetteer Service. However, this is not meant to constrain implementations of Gazetteer Services to be dependent on WFS capabilities. As with any OGC Web Service, the interface is designed to be independent of any particular implementation.

Consistently with the Open GIS Web Services architecture, and directly derived from the proposed WFS specification, the response to a <`GetCapabilities`> request for a Gazetteer Service contains the list of supported well-known Feature Types defined for the service, as shown in Appendix B.

Client applications may use the <`DescribeFeatureType`> interface to discover the specific set of properties for these Feature Types, and consequently allow additional query constraints and display of specific properties. This implies that it will be possible to construct a valid query without explicit knowledge of additional properties specific to the implementation.

Gazetteer Services require the <`GetFeature`> interface to return a set of feature instances whose types are must be listed in the <`GetCapabilities`> response and must conform to the schema returned by <`DescribeFeatureType`>.

With these interfaces, Web Service clients are able to:

- Determine if a WFS implementation is acting as a Gazetteer Service.
- Possibly query both Geocoder and Gazetteer Services in an interoperable way.
- Create extended types suitable for specific applications that can be portable.

Three interfaces apply to this version of Gazetteer Service:

  - GetCapabilities
  - DescribeFeatureType
  - GetFeature

In the future, Gazetteer Services may have transactional interfaces added to enable updates by ad-hoc authorities, in which case they could optionally implement transaction interfaces like those now specified for WFS [Ref1].

Query constraints

A Gazetteer Service must be able to support spatial and thematic search constraints, per the OGC Filter Encoding Spec [Ref3].

Feature Relationships

A Gazetteer entry will exploit the GML relationship properties to implement the ISO CD 19112 “location instance” hierarchy model [insert references] such that:

- Each “location instance” may have zero or more “parent location instances” for which the “location instance” is a sub-division.
- An ISO “parent location instance” is referred in this specification as a “Broader Term” with a property identifier “BT”.
- Each “location instance” may have zero or more “child location instances” which sub-divide the “location instance”.
- An ISO “child location instance” is referred in this specification as a “Narrower Term” with a property identifier “NT”.
- Each “location instance” may have zero or more “alternative geographic identifiers” which may be used as synonyms for the “location instance”.
- ISO “alternative geographic identifiers” are referred in this specification as a “Related Term” with a property identifier “RT”.

Terms addressable via a URI

The Gazetteer Service should allow feature relationships to be traversed easily. If links to remote features or relations (“`featureAssociations`” as defined by GML [Ref2]) are returned in the result, they must be compliant with GML2.

2. Interface Definitions (Normative)

The following interfaces are required for a Gazetteer Service:

- GetCapabilities
- DescribeFeatureType
- GetFeature

This specification provides the syntax of request and responses using the HTTP Protocol. Each request is given in both the CGI parameter form (HTTP-GET) and the XML document (HTTP-POST) form.

2.1. Interface *<GetCapabilities>*

Request

This interface is used to request a capabilities document from a Gazetteer Service.

Note: It is expected that this section will be replaced in a later revision, in whole or in part, with a reference to the "Basic Services Model".

Request Syntax

HTTP-POST

A valid XML document will be send to the service using the HTTP POST protocol. The document will conform to the following schema:

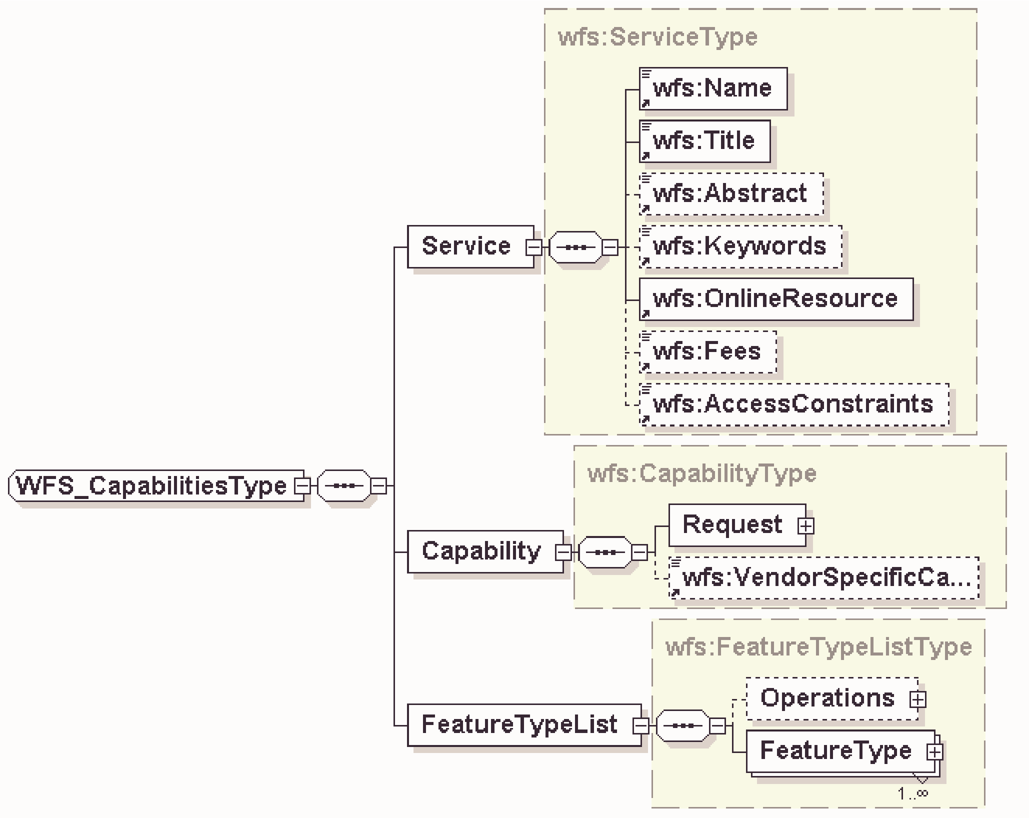
```
  <!-- ======================================================
Root element
  ====================================================== -->
<element name="GetCapabilities" type="wfs:GetCapabilitiesType"/>
<!-- ======================================================
Types
====================================================== -->
<complexType name="GetCapabilitiesType">
  <attribute name="version" type="CDATA" use="required"/>
</complexType>
```

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<GetCapabilities version="1.0.0">
</GetCapabilities>
```

Response (Return Values)

The capabilities response document derives from a common capabilities response using the derivation mechanisms of XML-schema. The response is an XML instance that conforms to the schema defined in Appendix A and shown graphically here:



Specifically, the Gazetteer Capabilities Response Schema must:

- Use the `FeatureTypeList` element to advertise that the service can be queried for a feature that matches the `GazetteerEntry` type, or a type that extends it, and can be substituted for it.

- Identify the vocabulary (e.g., a named Spatial Reference System by Geographic Identifiers) the Gazetteer entries define.

Issue Name: JVD-13Mar01-1 (JVD, 13Mar01)]

Issue Description: Which element in Capabilities response is used to identify the vocabulary?

Resolution: [Insert Resolution Details and History. (Initials, Date)]

My analysis is that until this is addressed in the BSM we can only put in service level Keyword elements for this. I had proposed that Keywords were a complex type that included role, term and vocabulary (scope) – as per vocab.xsd - instead of a simple string, but this seems to have been lost in the rebuild to XML-Schema. The alternative is that each property in GML adds an attribute that defines the scope of the content. This could be legal – but again needs a well defined thing to point a URI at – so its premature.

- Identify the vocabulary used for the contents of the `featureDesignation` property in the Gazetteer Feature Schema (Appendix B).

Issue Name: JVD-13Mar01-2 (JVD, 13Mar01)]

Issue Description: Which element in Capabilities response is used to identify the vocabulary?

Resolution: [Insert Resolution Details and History. (Initials, Date)]

Exceptions

None.

2.2. Interface *<GetFeature>*

Request

A Gazetteer service implements the ability to ask for a collection of features using the properties of the `GazetteerEntry` type or a feature type that extends this type. The schema of a Gazetteer response is defined in Appendix B.

Note: It is expected that this section will be replaced in a later revision, in whole or in part, with a reference to the “Basic Services Model” [Ref7].

Request Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/namespaces/wfs"
xmlns:wfs="http://www.opengis.net/namespaces/wfs"
xmlns="http://www.w3.org/2000/10/XMLSchema"
elementFormDefault="qualified">
   <annotation>
      <appinfo>GetFeatureRequest.xsd v0.2 2001-02</appinfo>
      <documentation xml:lang="en">WFS interface schema. Copyright
(c) 2001 OGC, All Rights Reserved.</documentation>
   </annotation>
   <!-- ===========================================================
   Includes and Imports
   ============================================================ -->
   <include
schemaLocation="http://www.opengis.net/namespaces/wfs/FilterReques
t.xsd"/>
   <!-- ===========================================================
   Global elements and attributes
   ============================================================ -->
   <element name="Query" type="wfs:QueryType"/>
   <!-- ===========================================================
   Root element
   ============================================================ -->
   <element name="GetFeature" type="wfs:GetFeatureType"/>
   <element name="GetFeatureWithLock"
type="wfs:GetFeatureWithLockType"/>
   <!-- ==========================================================
   Types
   ============================================================ -->
   <complexType name="GetFeatureType">
      <sequence>
         <element ref="wfs:Query" maxOccurs="unbounded"/>
         <element ref="wfs:Filter" minOccurs="0"/>
      </sequence>
      <attribute name="outputFormat" use="default" value="GML2"/>
      <attribute ref="wfs:handle" use="default" value=""/>
      <attribute ref="wfs:maxFeatures" use="default" value=""/>
   </complexType>
   <complexType name="GetFeatureWithLockType">
      <sequence>
      <element ref="wfs:Query" maxOccurs="unbounded"/>
      <element ref="wfs:Filter" minOccurs="0"/>
    </sequence>
    <attribute name="outputFormat" use="default" value="GML2"/>
    <attribute ref="wfs:handle" use="default" value=""/>
    <attribute ref="wfs:maxFeatures" use="default" value=""/>
  </complexType>
  <complexType name="QueryType">
    <sequence>
      <element ref="wfs:PropertyName" minOccurs="0"
maxOccurs="unbounded"/>
      <element ref="wfs:Filter" minOccurs="0"/>
    </sequence>
    <attribute ref="wfs:handle" use="default" value=""/>
    <attribute ref="wfs:typeName" use="required"/>
  </complexType>
  <simpleType name="OutputFormatType">
    <restriction base="string">
      <enumeration value="GML2"/>
      <enumeration value="CDATA"/>
    </restriction>
  </simpleType>
</schema>
```

The <GetFeature> interface allows selection queries to be serialized as XML documents and sent to the service via HTTP-POST. A limited subset may be supported via an equivalent CGI parameter syntax.

At a minimum, a Gazetteer service must support the following Filter operators on the <GetFeature> request:

1. the “PropertyIsEqualTo” operand against all the properties
2. the use of the “Inside” spatial operator with a Box geometry
3. the ability to join clauses with the <AND> conditional, and clauses be able to support the <OR> conditional against sub clauses.

Issue Name [GAZ-20]

Issue Description: [how to formally restrict this or discover it? – I think that this is likely to be implied by the service claiming to support the Gazetteer] ???

Resolution: If the WFS capabilities is not rich enough to describe this, it must be implied from the service metadata. On the other hand, support for this is very easily tested, so it’s a reasonable enough

Issue Name [GAZ-21 RA]

Issue Description: Requirements of filter and capabilities here are quite simple, and apply to all WFS: need to be able to ask for related features to be returned. If property “BT” is a link to another feature, then can we ask for properties BT.* If a property is complex (eg an address object – can we ask for address.zipcode?

Resolution:

Refer to Appendix B for the definition of the Gazetteer Service query schema.

Refer to Appendix C for examples of the use of the OGC Filter Encoding syntax to invoke Gazetteer Services.

Response (Return Values)

The response document is a GML `Feature Collection` containing only feature members that use the `GazetteerEntryType` or a type derived from `GazetteerEntryType` and declared equivalent to `GazetteerEntryType` using the XML Schema *substitutionGroup* attribute.

Refer to Appendix B for the definition of the Gazetteer Service feature schema. The Gazetteer feature schema is part of the normative specification and is described below (and in Appendix B). It defines the semantics and structure of the response. This specification is limited to a GML 2 encoding that has traversable relationships expressed using the GML 2 feature relationship schema.

Issue Name [JVD-14Mar01-1]

Issue Description: Should the response from GetFeature validate against the gazetteer.xsd schema?

If not, what schema is used to define the structure of the GetFeature response? The response from Ionic Gazetteer doesn't validate (doesn't seem to have a "xsi:schemaLocation" attribute referencing the 'gazetteer.xsd'). When I add it, it still doesn't validate. True – we don't believe its conformant, have reported the issues and have put in hacks to fix it at the client end for now…RA

Resolution: It should validate against the schema returned by DescribeFeatureType which must import gazetteer.xsd.

Exceptions

None.

2.4. Interface *<DescribeFeatureType>*

Request

The function of the <DescribeFeatureType> interface is to provide a client the means to request a definition of any feature type that a particular Gazetteer Service can provide. The description that is generated will define how a Gazetteer Service expects a client application to express the state of a feature to be created. In other words, the result of a <DescribeFeatureType> request is a GML2 feature schema definition.

A <DescribeFeatureType> request is composed of a list of names of feature types that are to be described.

Request Syntax

The request is defined by the following XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/namespaces/wfs"
xmlns:wfs="http://www.opengis.net/namespaces/wfs"
xmlns="http://www.w3.org/2000/10/XMLSchema"
elementFormDefault="qualified">
   <annotation>
      <appinfo>DescribeFeatureTypeRequest.xsd v0.2 2001-
02</appinfo>
      <documentation xml:lang="en">WFS interface schema. Copyright
(c) 2001 OGC, All Rights Reserved.</documentation>
   </annotation>
   <!-- ===========================================================
   Includes and Imports
   ============================================================ -->
   <!-- ===========================================================
   Global elements and attributes
   ============================================================ -->
   <!-- ===========================================================
   Root element
   ============================================================ -->
   <element name="DescribeFeatureType"
type="wfs:DescribeFeatureTypeType"/>
   <!-- ===========================================================
   Types
   ============================================================ -->
   <complexType name="DescribeFeatureTypeType">
      <sequence>
         <element name="TypeName" type="string" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
      <attribute name="outputFormat" type="wfs:OutputFormatType"
use="default" value="XMLSCHEMA"/>
   </complexType>
   <simpleType name="OutputFormatType">
      <restriction base="string">
         <enumeration value="XMLSCHEMA"/>
         <enumeration value="CDATA"/>
      </restriction>
   </simpleType>
</schema>
```

Type Definitions

TypeName

Used to list the set of feature types to describe. Currently, this specification defines only one type of feature: `GazetteerEntry`.

Attributes

`outputFormat`

Used to indicate the schema description language that should be used to describe a feature schema. The only mandated format is XMLSCHEMA; other vendor specific formats specified in the capabilities document are also possible.

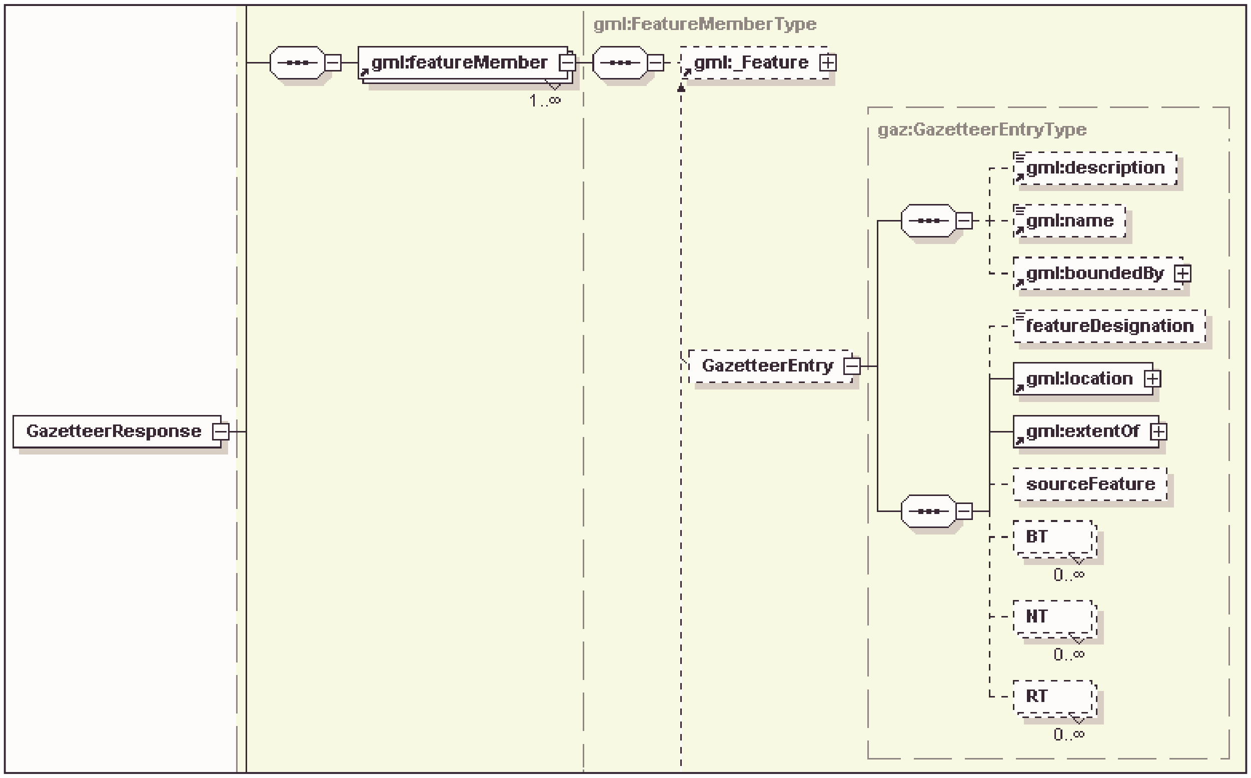
Response

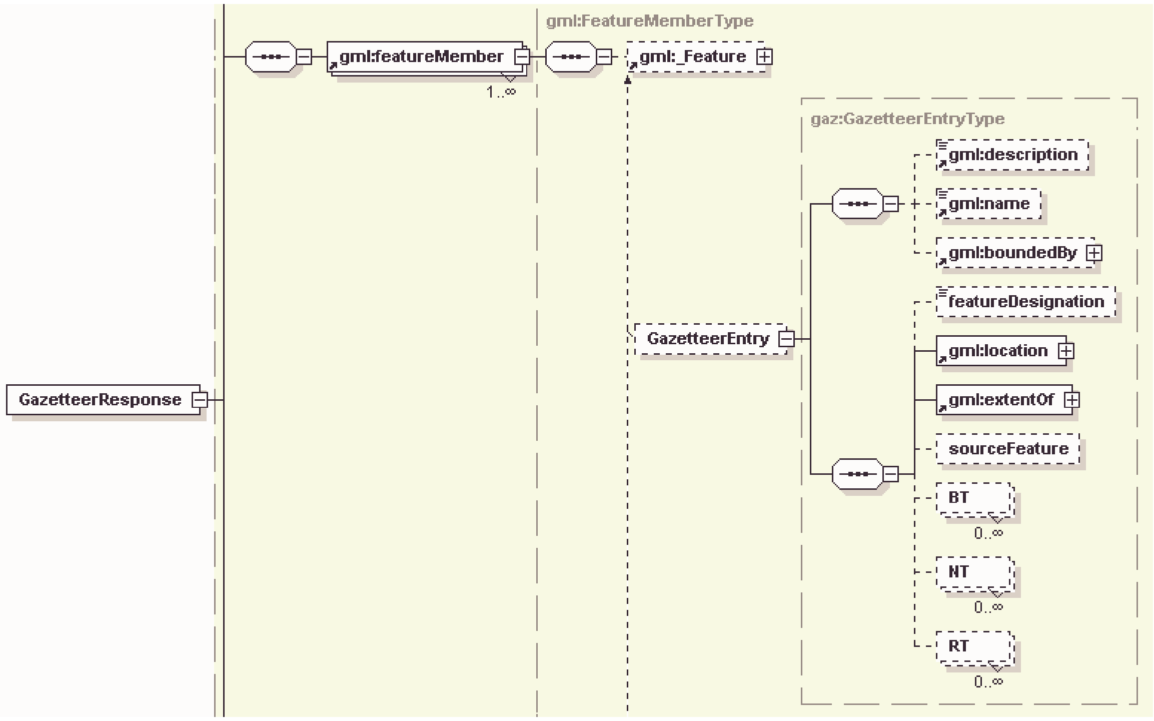
The response document is a XML Schema document defining the structure of `GazetteerEntryType`, or a type derived from `GazetteerEntryType` and declared equivalent to `GazetteerEntryType` using the XML Schema *substitutionGroup* attribute.

The `GazetteerEntryType` is declared in http://opengis.net/namespaces/gml/gazetteer/gazetteer.xsd

The local schema may restrict the generic schema according to the allowed values in the `featureDesignation` property, and whether or not the various GML geolink properties are available. A mechanism to reference remote vocabularies is yet to be specified, but is under consideration.

The Gazetteer Feature Schema (Appendix B) can be visualized thus:





Elements

`GazetteerResponse`

An element of type `GazetteerResponseType` defining a GML feature collection returned by a `<GetFeature>` request. Extends `gml: AbstractFeatureCollectionType` and is substitutable for `gml:_FeatureCollection`.

`GazetteerEntry`

A member of `GazetteerResponse` of type `GazetteerEntryType` defining a GML feature returned by a `<GetFeature>` request. Extends `gml:AbstractFeatureType` and is substitutable for `gml:_Feature`.

featureDesignation

An optional string element describing the use/purpose/significance of the `GazetteerEntry` instance.

gml:location

A geometry property of the `GazetteerEntry` instance.

gml:extentOf

The bounding rectangle property of the `GazetteerEntry` instance.

BT

A relationship of the `GazetteerEntry` instance to features defining its “broader terms”. For example, a BT feature of “Paris” could be “France”.

NT

A relationship of the `GazetteerEntry` instance to features defining its “narrower terms”. For example, a NT feature of “Paris” could be “Place de la Bastille”.

RT

A relationship of the `GazetteerEntry` instance to features defining its “related terms”. For example, a RT feature of “Paris” could be “Niece”.

Example Response

```
<?xml version="1.0" encoding="UTF-8"?>

<gml:featureCollection
xmlns:gazetteer="http://www.opengis.net/namespaces/gazetteer"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/namespaces/gazetteer/gazetteer.xs
d">
      <gml:boundedBy>
            <gml:Box srsName="EPSG:4326">
                  <gml:coordinates>150,-35 152,-33</gml:coordinates>
            </gml:Box>
      </gml:boundedBy>
      <gml:featureMember>
            <gazetteer:GazetteerEntry fid="NIMA-223434">
                  <gml:name>Sydney</gml:name>
                  <gazetteer:featureDesignation>Populated Place - State
Capital</gazetteer:featureDesignation>
                  <gml:location>
<gml:Point srsName="EPSG:4326">
                  <gml:coordinates>151,34</gml:coordinates>
</gml:Point>
                  </gml:location>
                  <gml:extentOf/>
                  <gazetteer:sourceFeature xlink:type="locator"
xlink:href="http://citystatistics.com/wfs?request=GETFEATURE&amp;FEATURE_ID=
Sydney,AU"/>
            <gazetteer:BT xlink:type="locator" xlink:href="#NIMA-2234"/>
            <gazetteer:NT xlink:type="locator" xlink:href="#AUS-0101"/>
            <gazetteer:NT xlink:type="locator" xlink:href="#AUS=0101"/>
            </gazetteer:GazetteerEntry>
      </gml:featureMember>
      <gml:featureMember>
```

```
            <gazetteer:GazetteerEntry fid="NIMA-223434">
                  <gml:name>Sydney</gml:name>
                  <gazetteer:featureDesignation>Populated Place - State
Capital</gazetteer:featureDesignation>
                  <gml:location>
                        <gml:Point srsName="EPSG:4326">
                              <gml:coordinates>151,-34</gml:coordinates>
                        </gml:Point>
                  </gml:location>
                  <gml:extentOf/>
                  <gazetteer:sourceFeature xlink:type="locator"
xlink:href="http://citystatistics.com/wfs?request=GETFEATURE&amp;FEATURE_ID=
Sydney,AU"/>
                  <gazetteer:BT xlink:type="locator" xlink:href="#NIMA-
2234"/>
                  <gazetteer:NT xlink:type="locator" xlink:href="#AUS-
0101"/>
                  <gazetteer:NT xlink:type="locator"
xlink:href="#AUS=0101"/>
            </gazetteer:GazetteerEntry>
      </gml:featureMember>
      <gml:featureMember>
            <gazetteer:GazetteerEntry fid="AUS-0101">
                  <gml:name>Annandale</gml:name>
                  <gazetteer:featureDesignation>Populated Place -
Suburb</gazetteer:featureDesignation>
                  <gml:location>
                        <gml:Point srsName="EPSG:4326">
                              <gml:coordinates>150.5,-34</gml:coordinates>
                        </gml:Point>
                  </gml:location>
                  <gazetteer:BT xlink:type="locator" xlink:href="#NIMA-
223434"/>
            </gazetteer:GazetteerEntry>
      </gml:featureMember>
      <gml:featureMember>
            <gazetteer:GazetteerEntry fid="AUS-0102">
                  <gml:name>Bondi</gml:name>
                  <gazetteer:featureDesignation>Populated Place -
Suburb</gazetteer:featureDesignation>
                  <gml:location>
                        <gml:Point srsName="EPSG:4326">
                              <gml:coordinates>151.5,-34</gml:coordinates>
                        </gml:Point>
                  </gml:location>
                  <gazetteer:BT xlink:type="locator" xlink:href="#NIMA-
223434"/>
            </gazetteer:GazetteerEntry>
      </gml:featureMember>
</gml:featureCollection>
```

Exceptions

None.

References

For the latest version of any OGC specification please consult the list of OpenGIS Specifications available at http://www.opengis.org.

1. **Web Feature Server Specification** (OGC Draft Candidate Implementation Specification) available online: **<**http://feature.opengis.org/members/archive/arch01/01-023r1.pdf**>**
2. **Geography Markup Language (GML) v2.0** **<**http://feature.opengis.org/members/archive/arch01/01-029.pdf**>**
3. **Filter Encoding Specification** (OGC Draft Candidate Implementation Specification) available online: **<**http://coverage.opengis.org/ip2000/docs/OGC_Filter_Encoding_0_0_11.pdf**>**
4. **Geocoder Service Specification** (OGC Draft Candidate Implementation Specification) available online: **<**http://ip.opengis.org/gfspp/docs/30jan01_Draft_GeocoderServiceV0.7.45.doc**>**
5. **Geoparser Service Specification** (OGC Draft Candidate Implementation Specification) available online: <http://ip.opengis.org/gfspp/docs/00-055(Geoparse)r5.doc>
6. Whiteside, A, and J. Bobbit. 2000. *Recommended Definition Data for Coordinate Reference Systems and Coordinate Transformations*. OGC Project Document 00-040r7.
7. **Basic Services Model** (Draft Candidate Implementation Specification 0.0.8) available online: <http://feature.opengis.org/members/archive/arch01/01-022r1.pdf>

Glossary

Term	Definition A

Appendix A: GetCapabilities Schema (Normative)

The schema listed below defines the structure of a <GetCapabilities> response.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/namespaces/wfs"
xmlns:wfs="http://www.opengis.net/namespaces/wfs"
xmlns="http://www.w3.org/2000/10/XMLSchema" elementFormDefault="qualified">
   <annotation>
      <appinfo>GetCapabilitiesResponse.xsd v0.2 2001-02</appinfo>
      <documentation xml:lang="en">WFS interface schema. Copyright (c) 2001
OGC, All Rights Reserved.</documentation>
   </annotation>
   <!-- ==============================================================
   Includes and Imports
   ============================================================== -->
   <!-- ==============================================================
   Global elements and attributes
   ============================================================== -->
   <element name="Abstract" type="string"/>
   <element name="AccessConstraints" type="string"/>
   <element name="Delete" type="string"/>
   <element name="Fees" type="string"/>
   <element name="GML.2" type="string"/>
   <element name="Insert" type="string"/>
   <element name="Keywords" type="string"/>
   <element name="Lock" type="string"/>
   <element name="Name" type="string"/>
   <element name="OnlineResource"/>
   <element name="Query" type="string"/>
   <element name="SRS" type="string"/>
   <element name="ServiceName" type="string"/>
   <element name="Title" type="string"/>
   <element name="Update" type="string"/>
   <element name="VendorSpecificCapabilities" type="string"/>
   <element name="XMLSCHEMA">
      <complexType/>
   </element>
   <!-- ==============================================================
   Root element
   ============================================================== -->
   <element name="WFS_Capabilities" type="wfs:WFS_CapabilitiesType"/>
   <!-- ==============================================================
   Types
   ============================================================== -->
   <complexType name="WFS_CapabilitiesType">
      <sequence>
         <element name="Service" type="wfs:ServiceType"/>
         <element name="Capability" type="wfs:CapabilityType"/>
         <element name="FeatureTypeList" type="wfs:FeatureTypeListType"/>
      </sequence>
      <attribute name="version" type="string" use="fixed" value="0.0.1"/>
      <attribute name="updateSequence" type="string" use="default"
value="0"/>
   </complexType>
   <complexType name="ServiceType">
      <sequence>
         <element ref="wfs:Name"/>
         <element ref="wfs:Title"/>
         <element ref="wfs:Abstract" minOccurs="0"/>
         <element ref="wfs:Keywords" minOccurs="0"/>
         <element ref="wfs:OnlineResource"/>
      <element ref="wfs:Fees" minOccurs="0"/>
      <element ref="wfs:AccessConstraints" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="CapabilityType">
    <sequence>
      <element name="Request" type="wfs:RequestType"/>
      <element ref="wfs:VendorSpecificCapabilities" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="FeatureTypeListType">
    <sequence>
      <element name="Operations" type="wfs:OperationsType"
minOccurs="0"/>
      <element name="FeatureType" type="wfs:FeatureTypeType"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <complexType name="RequestType">
    <choice maxOccurs="unbounded">
      <element name="GetCapabilities" type="wfs:GetCapabilitiesType"/>
      <element name="DescribeFeatureType"
type="wfs:DescribeFeatureTypeType"/>
      <element name="Transaction" type="wfs:TransactionType"/>
      <element name="GetFeatureType" type="wfs:GetFeatureTypeType"/>
      <element name="LockFeatureType" type="wfs:LockFeatureTypeType"/>
    </choice>
  </complexType>
  <complexType name="GetCapabilitiesType">
    <sequence>
      <element name="DCPType" type="wfs:DCPTypeType"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <complexType name="DescribeFeatureTypeType">
    <sequence>
      <element name="SchemaDescriptionLanguage"
type="wfs:SchemaDescriptionLanguageType"/>
      <element name="DCPType" type="wfs:DCPTypeType"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <complexType name="TransactionType">
    <sequence>
      <element name="DCPType" type="wfs:DCPTypeType"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <complexType name="GetFeatureTypeType">
    <sequence>
      <element name="ResultFormat" type="wfs:ResultFormatType"/>
      <element name="DCPType" type="wfs:DCPTypeType"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <complexType name="LockFeatureTypeType">
    <sequence>
      <element name="DCPType" type="wfs:DCPTypeType"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <complexType name="DCPTypeType">
    <sequence>
      <element name="HTTP" type="wfs:HTTPType"/>
    </sequence>
  </complexType>
  <complexType name="FeatureTypeType">
    <sequence>
      <element ref="wfs:Name"/>
      <element ref="wfs:Title" minOccurs="0"/>
      <element ref="wfs:Abstract" minOccurs="0"/>
      <element ref="wfs:Keywords" minOccurs="0"/>
      <element ref="wfs:SRS"/>
      <element name="Operations" type="wfs:OperationsType"
minOccurs="0"/>
      <element name="LatLonBoundingBox"
type="wfs:LatLonBoundingBoxType"/>
      <element name="MetadataURL" type="wfs:MetadataURLType"
minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <complexType name="GetType">
    <attribute name="onlineResource" type="string" use="required"/>
  </complexType>
  <complexType name="HTTPType">
    <choice maxOccurs="unbounded">
      <element name="Get" type="wfs:GetType"/>
      <element name="Post" type="wfs:PostType"/>
    </choice>
  </complexType>
  <complexType name="LatLonBoundingBoxType">
    <attribute name="minx" type="string" use="required"/>
    <attribute name="miny" type="string" use="required"/>
    <attribute name="maxx" type="string" use="required"/>
    <attribute name="maxy" type="string" use="required"/>
  </complexType>
  <complexType name="MetadataURLType">
    <simpleContent>
      <restriction base="string">
        <attribute name="type" use="required">
          <simpleType>
            <restriction base="NMTOKEN">
              <enumeration value="TC211"/>
              <enumeration value="FGDC"/>
            </restriction>
          </simpleType>
        </attribute>
        <attribute name="format" use="required">
          <simpleType>
            <restriction base="NMTOKEN">
              <enumeration value="XML"/>
              <enumeration value="SGML"/>
              <enumeration value="TXT"/>
            </restriction>
          </simpleType>
        </attribute>
      </restriction>
    </simpleContent>
  </complexType>
  <complexType name="OperationsType">
    <choice maxOccurs="unbounded">
      <element ref="wfs:Insert"/>
      <element ref="wfs:Update"/>
      <element ref="wfs:Delete"/>
      <element ref="wfs:Query"/>
      <element ref="wfs:Lock"/>
    </choice>
  </complexType>
  <complexType name="PostType">
    <attribute name="onlineResource" type="string" use="required"/>
  </complexType>
  <complexType name="ResultFormatType">
```

```
      <sequence maxOccurs="unbounded">
         <element ref="wfs:GML.2"/>
      </sequence>
   </complexType>
   <complexType name="SchemaDescriptionLanguageType">
      <sequence maxOccurs="unbounded">
         <element ref="wfs:XMLSCHEMA"/>
      </sequence>
   </complexType>
</schema>
```

This GetCapabilities Schema document can be found at http://www.opengis.net/namespaces/wfs/GetCapabilitiesResponse.xsd

Appendix B: Gazetteer Feature Schema (Normative)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- File: gazetteer.xsd  -->
<schema targetNamespace="http://www.opengis.net/namespaces/gazetteer"
xmlns="http://www.w3.org/2000/10/XMLSchema"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gazetteer="http://www.opengis.net/namespaces/gazetteer"
elementFormDefault="qualified" version="2.02">
      <annotation>
            <appinfo>GazetteerEntry.xsd v0.0.2</appinfo>
            <documentation>Schema for Generic Gazetteer Entry. Gazetteer
Services must deliver  </documentation>
      </annotation>
      <!-- import constructs from the GML Feature and Geometry schemas -->
      <import namespace="http://www.opengis.net/gml"
schemaLocation="http://www.opengis.net/namespaces/gml/core/feature.xsd"/>
      <!-- ==============================================================
      global element declarations
      =================================================================== -
->
      <!-- this defines that the response should contain a single feature
collection -->
      <element name="GazetteerResponse"
type="gazetteer:GazetteerResponseType"
substitutionGroup="gml:_FeatureCollection"/>
      <complexType name="GazetteerResponseType">
            <complexContent>
                  <extension base="gml:AbstractFeatureCollectionType">
                        <sequence>
                              <!-- include any Gazetteer response metadata
here -->
                        </sequence>
                  </extension>
            </complexContent>
      </complexType>
      <!-- this defines that the GazetteerEntry type is a valid instance of
the abstract _Feature -->
      <element name="GazetteerEntry" type="gazetteer:GazetteerEntryType"
substitutionGroup="gml:_Feature"/>
      <!-- this sets up the extended properties we expect and allow in a
Gazetteer -->
      <complexType name="GazetteerEntryType">
            <complexContent>
                  <extension base="gml:AbstractFeatureType">
                        <sequence>
                              <element name="featureDesignation"
type="string" minOccurs="0"/>
                              <element ref="gml:location"/>
                              <element ref="gml:extentOf"/>
                              <element name="sourceFeature"
type="gazetteer:RemoteFeatureRefType" minOccurs="0"/>
                              <element name="BT"
type="gazetteer:TermRelationshipType" minOccurs="0" maxOccurs="unbounded"/>
                              <element name="NT"
type="gazetteer:TermRelationshipType" minOccurs="0" maxOccurs="unbounded"/>
                              <element name="RT"
type="gazetteer:TermRelationshipType" minOccurs="0" maxOccurs="unbounded"/>
                        </sequence>
                  </extension>
            </complexContent>
      </complexType>
      <complexType name="RemoteFeatureRefType">
            <attributeGroup ref="xlink:locatorLink"/>
      </complexType>
      <complexType name="TermRelationshipType">
            <attributeGroup ref="xlink:locatorLink"/>
      </complexType>
</schema>
```

The Gazetteer XML Schema document can be found at http://www.opengis.net/namespaces/gazetteer/gazetteer.xsd

A sample extended Feature Type schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
xmlns:gaz="http://www.opengis.net/namespaces/gazetteer"
elementFormDefault="qualified" attributeFormDefault="unqualified">
      <!-- import constructs from the GML Feature and Geometry schemas -->
      <xsd:import namespace="http://www.opengis.net/namespaces/gazetteer"
schemaLocation="gazetteer.xsd"/>
      <xsd:element name="PopulatedPlace" type=" PopulatedPlaceType"
substitutionGroup="gazetteer:GazetteerEntry">
            <xsd:annotation>
                  <xsd:documentation>An example extended Gazetteer Entry
feature Type</xsd:documentation>
            </xsd:annotation>
            <xsd:complexType name="PopulatedPlaceType">
                  <xsd:complexContent>
                        <xsd:extension base="gazetteer:GazetteerEntryType">
                              <xsd:sequence>
                                    <xsd:element name="population"
type="xsd:integer"/>
                              </xsd:sequence>
                        </xsd:extension>
                  </xsd:complexContent>
            </xsd:complexType>
      </xsd:element>
</xsd:schema>
```

Appendix C: Gazetteer Example (Informative)

Example 1: Accessing geographic properties of a known feature

Query:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetFeature xmlns:gml="http://www.opengis.net/gml"
xmlns:fo="http://www.w3.org/1999/XSL/Format" maxFeatures="3">
   <Query typeName="GNS_GAZ">
      <Filter>
         <FeatureId fid="GNS_GAZ.-474536" />
      </Filter>
   </Query>
</GetFeature>
```

Results:

```
<?xml version="1.0" encoding="utf-8" ?>
  <gml:featureCollection scope="http://benoit/servlet/wfs/GNS_GAZ?"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gazetteer="http://www.opengis.net/namespaces/gazetteer"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xmlns:wfs="http://www.ionicsoft.com/wfs">
    <gml:boundedBy>
      <gml:Box srsName="EPSG:4326">
        <gml:coordinates>34.416668,29.691668
34.416668,29.691668</gml:coordinates>
      </gml:Box>
    </gml:boundedBy>
  <gml:featureMember>
    <wfs:GNS_GAZ fid="GNS_GAZ.-474536">
      <gml:description>Garf el Themed</gml:description>
      <gml:name>Garf el Themed</gml:name>
      <gazetteer:featureDesignation>RDGE
</gazetteer:featureDesignation>
      <gml:location>
        <gml:Point srsName="EPSG:4326">

<gml:coordinates>34.4166667,29.6916667,0.0</gml:coordinates>
        </gml:Point>
      </gml:location>
      <gazetteer:sourceFeature
xlink:type="locator"></gazetteer:sourceFeature>
      <gazetteer:BT xlink:type="locator"
xlink:href="/servlet/wfs/GNS_GAZ?request=GETFEATURE&amp;TYPENAME=GNS_
GAZ&amp;FEATUREID=GNS_GAZ.-451924"></gazetteer:BT>
    </wfs:GNS_GAZ>
  </gml:featureMember>
  </gml:featureCollection>
```

Example 2: Mixing name and spatial queries

Query:

```
<?xml version="1.0" encoding="UTF-8"?>
       <GetFeature xmlns:gml="http://www.opengis.net/gml"
```

```
xmlns:fo="http://www.w3.org/1999/XSL/Format"
       maxFeatures="300">
        <Query typeName="GNS_GAZ">
         <Filter>
          <And>
          <Or>
           <PropertyIsEqualTo>
            <PropertyName>name</PropertyName>
            <Literal>Garf el Themed</Literal>
           </PropertyIsEqualTo>
           <PropertyIsEqualTo>
            <PropertyName>name</PropertyName>
            <Literal>Jericho</Literal>
           </PropertyIsEqualTo>
          </Or>
          <Intersects>
            <PropertyName>gml:location</PropertyName>
            <gml:Box srsName="EPSG:4326">
             <gml:coordinates>34.13,29.43
35.72,33.28</gml:coordinates>
            </gml:Box>
          </Intersects>
          </And>
         </Filter>
        </Query>
       </GetFeature>
```

Response:

```
<?xml version="1.0" encoding="utf-8" ?>
  <gml:featureCollection scope="http://benoit/servlet/wfs/GNS_GAZ?"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gazetteer="http://www.opengis.net/namespaces/gazetteer"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xmlns:wfs="http://www.ionicsoft.com/wfs">
    <gml:boundedBy>
      <gml:Box srsName="EPSG:4326">
        <gml:coordinates>34.416668,29.691668
35.45,31.866667</gml:coordinates>
      </gml:Box>
    </gml:boundedBy>
  <gml:featureMember>
    <wfs:GNS_GAZ fid="GNS_GAZ.-1144927">
      <gml:description>Jericho</gml:description>
      <gml:name>Jericho</gml:name>
      <gazetteer:featureDesignation>PPL
</gazetteer:featureDesignation>
      <gml:location>
        <gml:Point srsName="EPSG:4326">
          <gml:coordinates>35.45,31.8666667,0.0</gml:coordinates>
        </gml:Point>
      </gml:location>
      <gazetteer:sourceFeature
xlink:type="locator"></gazetteer:sourceFeature>
    </wfs:GNS_GAZ>
  </gml:featureMember>
  <gml:featureMember>
    <wfs:GNS_GAZ fid="GNS_GAZ.-474536">
      <gml:description>Garf el Themed</gml:description>
      <gml:name>Garf el Themed</gml:name>
      <gazetteer:featureDesignation>RDGE
</gazetteer:featureDesignation>
      <gml:location>
        <gml:Point srsName="EPSG:4326">

<gml:coordinates>34.4166667,29.6916667,0.0</gml:coordinates>
```

```
        </gml:Point>
      </gml:location>
      <gazetteer:sourceFeature
xlink:type="locator"></gazetteer:sourceFeature>
      <gazetteer:BT xlink:type="locator"
xlink:href="/servlet/wfs/GNS_GAZ?request=GETFEATURE&amp;TYPENAME=GNS_
GAZ&amp;FEATUREID=GNS_GAZ.-451924"></gazetteer:BT>
    </wfs:GNS_GAZ>
  </gml:featureMember>
  </gml:featureCollection>
```

top contents spec glossary references