ICS 35.240.70
A 75

中华人民共和国国家标准

GB/T 30169—2013/ISO 19142:2010

地理信息　基于网络的要素服务

Geographic information—Web feature service

(ISO 19142:2010,IDT)

2013-12-17 发布　　2014-04-20 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　次

前　言

本标准按照 GB/T 1.1—2009 给出的规则起草。

本标准使用翻译法等同采用国际标准 ISO 19142:2010《地理信息　基于网络的要素服务》。

本标准作了以下编辑性修改：

——用“本标准”代替“本国际标准”；

——删除了该国际标准的“封面”“目次”和“前言”；

——在 14.2.2 中，将 Title、Abstract 中 Language 的缺省值设置为 IETF RFC 4646 中规定的代表中文的编码“zh”；

——增加了资料性附录 G，列出了本标准定义的操作及参数名称的中英文对照。

本标准由国家测绘地理信息局提出。

本标准由全国地理信息标准化技术委员会(SAC/TC 230)归口。

本标准起草单位：武汉大学测绘遥感信息工程国家重点实验室、国家信息中心、武大吉奥信息技术有限公司。

本标准主要起草人：龚健雅、高文秀、宦茂盛、朱欣焰、吴华意、邓跃进、石雯雯。

引　言

基于网络的要素服务(Web Feature Service,WFS)代表了通过网络创建、修改和交换地理信息方式上的转变。不同于通过文件传输协议(File Transfer Protocol,FTP)以文件级共享地理信息,WFS 实现了要素和要素特性级直接访问和共享地理信息,允许用户只获取或修改所需的数据,而非获取包含所需数据和其他更多非所需数据的整个文件。这些数据可以用于很多种应用目的,包括并非数据提供者预计的应用目的。

在 ISO 19119 定义的有关服务分类中,WFS 主要被定义为一种要素访问服务,同时还包含要素类型服务、坐标变换/转换服务和地理数据格式转换服务等多种元素。

地理信息　基于网络的要素服务

1　范围

本标准规定了一种不依赖数据存储方式的处理和访问地理要素的服务的行为规范，它定义了发现、查询、锁定、事务操作以及管理存储的参数化查询表达式的操作。

发现操作允许询问服务所具备的能力，并检索该服务所提供的定义地理要素类型的应用模式。

查询操作允许基于客户端定义的有关要素特性的约束条件，从数据存储中获取相应的要素或要素特性的值。

锁定操作允许为了修改或删除要素、排他性地访问要素。

事务操作允许创建、修改、替换和删除指定数据存储中的要素。

存储的查询操作允许客户端创建、删除、列举和描述参数化的查询表达式，这些表达式存储在服务器端，可以采用不同的参数值重复调用这些表达式。

注：本标准并不涉及有关访问控制的内容。

本标准定义了十一种操作：

1）　GetCapabilities(获取服务能力)(属于发现操作)；
2）　DescribeFeatureType (描述要素类型)(属于发现操作)；
3）　GetPropertyValue(获取特性值)(属于查询操作)；
4）　GetFeature(获取要素)(属于查询操作)；
5）　LockFeature (锁定要素)(属于锁定操作)；
6）　GetFeatureWithLock (获取要素并锁定)(属于查询和锁定操作)；
7）　Transaction (事务操作)(属于事务操作)；
8）　CreateStoredQuery (创建存储的查询)(属于存储的查询操作)；
9）　DropStoredQuery (删除存储的查询)(属于存储的查询操作)；
10）　ListStoredQueries (列举存储的查询)(属于存储的查询操作)；
11）　DescribeStoredQueries (描述存储的查询)(属于存储的查询操作)。

2　一致性

表 1 规定了本标准定义的一致性类，为了保证与每个类保持一致性，应满足附录 A 中规定的测试。

表 1 也列举了：

——每个 WFS 一致性类要符合的过滤器编码(Filter Encoding，FE)(见 ISO 19143:2010，第 2 章)一致性测试。

——每个 WFS 一致性类要符合的 GML(见 GB/T 23708—2009)一致性测试。

表 1 WFS 一致性类

一致性类名	操作或行为	WFS 一致性测试	FES 一致性测试	GML 一致性测试
简单 WFS	服务应实现下列操作：GetCapabilities、DescribeFeatureType、ListStoredQueries、DiscribeStoredQueries 以及至少包含 StoredQuery（存储的查询）行为的 GetFeature 操作。 应提供能根据 id 获取要素的存储的查询，也可以提供其他存储的查询。 另外，此服务应至少符合 HTTP GET、HTTP POST 或 SOAP 一致性类中的其中之一	A.1.1	ISO 19143：2010，A.1	GB/T 23708—2009，A.1.1、A.1.4、A.1.5、A.1.7、B.3、B.5、B.2.3
基础 WFS	服务应实现简单 WFS 的一致性类，同时实现具备查询行为的 GetFeature 操作和 GetPropertyValue 操作	A.1.2	ISO 19143：2010，A.2、A.7、A.8、A.10、A.11、A.12、A.14	GB/T 23708—2009，B.4
事务 WFS	服务应实现基础 WFS 一致性类，还需实现 Transaction 操作	A.1.3		
带锁 WFS	服务应实现事务 WFS 一致性类，还需至少实现 GetFeatureWithLock 与 LockFeature 两操作中的一个	A.1.4		
HTTP GET	服务应为其所提供的操作实现关键字-值对(KVP)编码	A.1.5		
HTTP POST	服务应为其提供的操作实现 XML 编码	A.1.6		
SOAP	服务应在 SOAP Envelope(SOAP 信封)中实现 XML 编码的请求和响应结果	A.1.7		
继承	服务应为 XPath 表达式中实现 schema-element()(模式元素)函数	A.1.8	ISO 19143：2010，A.15	
远程解析	服务应实现解析远程资源引用的能力	A.1.9		GB/T 23708—2009，B.2.1
分页响应	服务应实现在响应中分页显示要素或值的能力	A.1.10		GB/T 23708—2009，B.3
标准连接	服务应用除空间和时间算子之外所有的 Filter(过滤)算子来实现连接谓词	A.1.11	ISO 19143：2010，A.8、A.10	
空间连接	服务应使用空间算子实现连接谓词	A.1.12	ISO 19143：2010，A.11、A.12	
时间连接	服务应使用时间算子实现的连接谓词	A.1.13	ISO 19143：2010，A.9、A.10	
要素版本	服务应实现追踪要素版本的能力	A.1.14	ISO 19143：2010，A.11	
管理存储的查询	服务应实现 CreateStoredQuery 与 DropStoredQuery 操作	A.1.15	ISO 19143：2010，A.1	

3 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 23708—2009 地理信息 地理标记语言(GML)(ISO 19136:2007,IDT)

ISO/TS 19103:2005 地理信息 概念模式语言(Geographic information—Conceptual schema language)

ISO 19143:2010 地理信息 过滤器编码(Geographic information—Filter Encoding)

IETF RFC 2616 超文本传输协议(1999 年 6 月)(Hypertext Transfer Protocol - HTTP/1.1 (June 1999))

IETF RFC 4646 识别语言的标签(2006 年 9 月)(Tags for the Identifying Languages (September 2006))

OGC 06-121r3:2009 OGC 网络服务通用规范(2009 年 2 月 9 日)(OGC Web Services Common Specification,OGC® Implementation Specification (9 February 2009))

OGC 07-092r3 OGC 命名空间中 URN 的定义标识符(2009 年 1 月 15 日)(Definition identifier URNs in OGC namespace, OGC® Best Practices (15 January 2009))

W3C SOAP W3C 简单对象访问协议(SOAP)1.2(2007 年 4 月 27 日)(Simple Object Access Protocol (SOAP) 1.2, W3C Note (27 April 2007))

W3C WSDL W3C 网络服务描述语言(WSDL)1.2(2001 年 3 月 15 日)(Web Services Description Language (WSDL) 1.1, W3C Note (15 March 2001))

W3C XML Namespaces XML 中的命名空间(1999 年 1 月 14 日)(Namespaces in XML, W3C Recommendation (14 January 1999))

W3C XML Path Language XML 路径语言(XPath)(2007 年 1 月 23 日)(XML Path Language (XPath) 2.0, W3C Recommendation (23 January 2007))

W3C XML Schema Part 1 XML 模式 第 1 部分:结构(2001 年 5 月 2 日)(XML Schema Part 1: Structures, W3C Recommendation (2 May 2001))

W3C XML Schema Part 2 XML 模式 第 2 部分:数据类型 (2001 年 5 月 2 日)(XML Schema Part 2: Datatypes, W3C Recommendation (2 May 2001))

4 术语和定义

下列术语和定义适用于本文件。

4.1

属性<XML> attribute <XML>

<XML>元素(4.6)中包含的名称—值对。

[GB/T 23708—2009,4.1.3]

注：本标准中，除非有特别说明，属性均指 XML 属性。

4.2

客户端 client

能从服务器(4.28)调用操作(4.17)的软件组件。

[GB/T 25597—2010,4.1]

4.3

坐标 coordinate

表示 n 维空间中点位置的某一序列 n 个数之一。

[GB/T 17694—2009,B.24]

4.4

坐标参照系 coordinate reference system

通过基准和对象相关联的坐标系(4.5)。

[GB/T 17694—2009,B.88]

4.5

坐标系 coordinate system

说明给点赋坐标(4.3)的数学规则集。

[GB/T 17694—2009,B.90]

4.6

元素<XML> element<XML>

XML 文档中的基本信息项,其中包含子元素、属性(4.1)和字符数据。

[GB/T 23708—2009,4.1.23]

4.7

要素 feature

现实世界现象的抽象。

[GB/T 17694—2009,B.179]

注:要素可以类型或实例的形式出现,当仅表达一种含义时,应使用要素类型或要素实例。

4.8

要素的标识符 feature identifier

唯一指定要素(4.7)实例的标识符。

4.9

过滤器表达式 filter expression

用 XML 编码的谓词表达式。

[ISO 19143:2010,4.11]

4.10

接口 interface

描述实体行为特征的命名操作(4.17)集合。

[GB/T 17694—2009,B.260]

4.11

连接谓词 join predicate

包含一个或多个子句的过滤器表达式(4.9),这些子句约束来自两个不同要素类型的特性。

[ISO 19143:2010,4.16]

注:本标准中,实体的类型是指要素(4.7)类型。

4.12

连接元组 join tuple

满足包含连接谓词(4.11)的过滤器的两个或多个对象实例的集合。

注:本标准中,对象实例是指要素(4.7)实例。

4.13

本地资源 local resource

系统能够直接控制的资源。

注：在本标准中，系统就是指基于网络的要素服务，而本地资源就是指服务器能够直接控制的数据库中的资源。

4.14

寻址属性　locator attribute

指引用**本地资源**(4.13)或**远程资源**(4.20)的**属性**(4.1)。

注：在XML编码中，该属性通常被称作href，包含指向远程资源的URI引用(见W3C XLink)。

4.15

MIME类型　Multipurpose Internet Mail Extensions(MIME) type

规定消息体中数据的媒体类型和子类型，并完整说明这些数据的原本表示(规范形式)。

[IETF RFC 2045:1996]

4.16

命名空间<XML>　namespace<XML>

经**URI(统一资源标识符)**(4.33)引用标识的名称集合，这些名称在XML文档中为**元素**(4.6)和**属性**(4.1)的名称。

[GB/T 17694—2009，B.318]

4.17

操作　operation

对象可以被调用执行的转换或查询的规范。

[GB/T 17694—2009，B.332]

4.18

特性　property

对象的某个方面或属性，通过名称来引用。

[ISO 19143:2010，4.21]

4.19

资源　resource

能满足某种需求的资产或手段。

[GB/T 19710—2005，4.10]

注：本标准中，资源是指**要素**(4.7)，或要素可识别的组成部分(例如元素的特性)。

4.20

远程资源　remote resource

不在系统直接控制下的资源。

注：本标准中，系统是基于网络的要素服务，远程资源不是存储在该服务能够直接控制的数据库中，也不能通过该服务直接存储。

4.21

请求　request

客户端(4.2)对**操作**(4.17)的调用。

[GB/T 25597—2010，4.10]

4.22

重新定位<引用>　relocate<reference>

更新对已经移动或复制到新地方的资源的引用。

示例：**服务器**(4.28)要生成对GetFeature的**请求**(4.21)的**响应**(4.24)，它应要复制引用的**要素**(4.7)到响应文档，而且服务器也应要“重新定位”包含在引用要素中的原始引用到响应文档中的备份。

4.23

解析　resolve

获取被引用的资源，并将其插入到服务器生成的响应文档中。

注：插入可以要么通过用资源的备份替换内联的引用，要么通过重新定位响应文档中的资源备份的引用位置。

4.24

响应 response

服务器(4.28) 返回给客户端(4.2) 的操作(4.17) 结果。

[GB/T 25597—2010，4.11]

4.25

响应模型 response model

定义出现在查询操作(4.17) 的响应(4.24) 结果中的每种要素属性的模式(4.26) 。

注：该模型是客户端可通过 DescribeFeatureType 操作获得要素类型的模式(见第 9 章)。

4.26

模式 schema

模型的形式化描述。

[GB/T 25597—2010，4.11]

注：通常，模式是对象的特征和与其他对象关系的抽象表示。XML 模式代表 XML 对象(例如，文档或文档的一部分)的属性(4.1) 与元素(4.6) 之间的关系。

4.27

模式< XML 模式> schema < XML Schema >

同一目标命名空间(4.16) 中模式(4.26) 组件的集合。

[GB/T 23708—2009，4.1.54]

示例：W3C XML 模式的模式组成为类型、元素(4.6) 、属性(4.1) 、群体等。

4.28

服务器 server

服务(4.29) 的具体实例。

[GB/T 25597—2010，4.12]

4.29

服务 service

实体通过接口(4.10) 提供功能的可区分部分。

[GB/T 17694—2009，B.427]

4.30

服务元数据 service metadata

描述服务器(4.28) 上可用的操作(4.17) 和地理信息的元数据。

[GB/T 25597—2010，4.14]

4.31

遍历< XML > traversal < XML >

针对任何目的使用 Xlink 链接。

[W3C XLink:2001]

4.32

元组 tuple

值的有序列表。

[GB/T 23708—2009，4.1.63]

注：在本标准中，有序列表是每个指定要素类型的要素(4.7) 的一个有限序列。

4.33

统一资源标识 Uniform Resource Identifier;URI

资源的唯一标识符，其结构同 IETF RFC 2396 保持一致。

[GB/T 23708—2009，4.1.65]

注：通用的语法是< scheme >:< scheme-specified-part >，**namespace**(4.16)的分层语法为< scheme >://< authority >< path >? < query >。

5 约定

5.1 缩略语

下列缩略语适用于本文件。

CGI:通用网关接口(Common Gateway Interface)

CRS:坐标参照系(Coordinate Reference System)

DCP :分布式计算平台(Distributed Computing Platform)

EPSG:欧洲石油调查局(European Petroleum Survey Group)

FES:过滤器编码规范(Filter Encoding Specification)

GML:地理标记语言(Geography Markup Language)

HTTP:超文本传输协议(Hypertext Transfer Protocol)

HTTPS:安全超文本传输协议(Secure Hypertext Transfer Protocol)

IETF:网络工程任务组(Internet Engineering Task Force)

KVP:关键字和值的对(Keyword-value pairs)

MIME:多用途网络邮件扩展协议(Multipurpose Internet Mail Extensions)

OGC :开放地理信息联盟(Open GIS Consortium)

OWS:OGC 网络服务(OGC Web Service)

SOAP:简单对象访问协议(Simple object access protocol)

URI:统一资源标识符(Uniform Resource Identifier)

URL:统一资源定位器(Uniform Resource Locator)

URN:统一资源名称(Uniform Resource Name)

VSP:厂商设定参数(Vendor Specific Parameter)

WFS :基于网络的要素服务(Web Feature Service)

WSDL:网络服务描述语言(Web Service Description Language)

XML:可扩展标记语言(Extensible Markup Language)

5.2 示例的使用

本标准大量地使用了 XML 示例，用来解释本标准讨论的 WFS 的各个方面。有些示例在附录 B 中，有些示例嵌在正文中。所有的示例引用的是虚拟的服务器和数据，因此，本标准不保证从本标准拷贝的任何 XML 或关键字一值对编码的示例，都可以正确实现或可以通过某个 XML 验证工具的验证。

5.3 XML 模式

本标准利用 XML 模式[见 W3C XML 模式第 1 部分(Part 1)，W3C XML 模式第 2 部分(Part 2)]片段用来定义 WFS 操作的 XML 编码，这些片段分别对应于附录 C 中的相关有效模式文件中。

5.4 UML 标记

5.4.1 类图

本标准采用统一建模语言(UML)描述各操作的结构图，UML 标记的主要元素如图 1 所示。

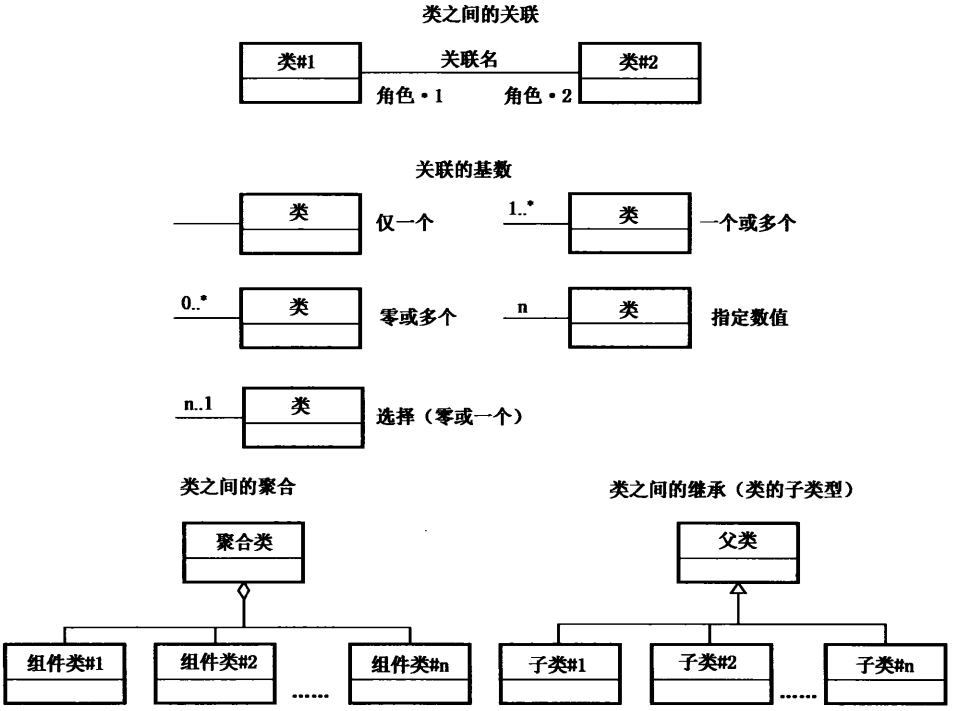

图 1 UML 标记

在这些类图中，本标准用到了以下 UML 类：

a) << DataType >>：缺乏标识（独立存在和可能带副作用）的参数集描述符。DataType 是不带任何操作的类，主要用来存放信息。

b) << Enumeration >>：一种数据类型，其实例包含一组可选字符值，是某个类中可能值的简短列表。

c) << CodeList >>：可变大小的枚举类型，用以表达较长的可能的选择值列表。如果这个选择列表完全已知，则应使用枚举类型；如果只知道可能的选择性，则应使用 code list。

d) << Interface >>：一组操作的定义，拥有该接口的对象都可以支持这些操作。接口类不能包含任何属性。

e) << Type >>：典型类型类，用以规范某个域的实例（对象）以及适用这些对象的操作。Type 类可以包含属性和关联。

f) << Union >>：交替出现的属性组成的一组列表，每次只能出现其中的一种属性。

见 ISO/TS 19103:2005 的 6.8.2 和 D.8.3。

本标准中使用了如下的标准数据类型：

1) CharacterString——字符串；

2) LocalisedCharacterString——与位置有关的字符串；

3) Boolean——指定是 TRUE（真）或是 FALSE（假）的值；

4) URI——提供更多信息的资源识别符；

5) Integer——整型数。

5.4.2 状态机符号

动态建模的使用方法在 UML 参考手册中有详细描述。主要技术是状态视图，状态视图的 UML 符号的说明如图 2 所示。

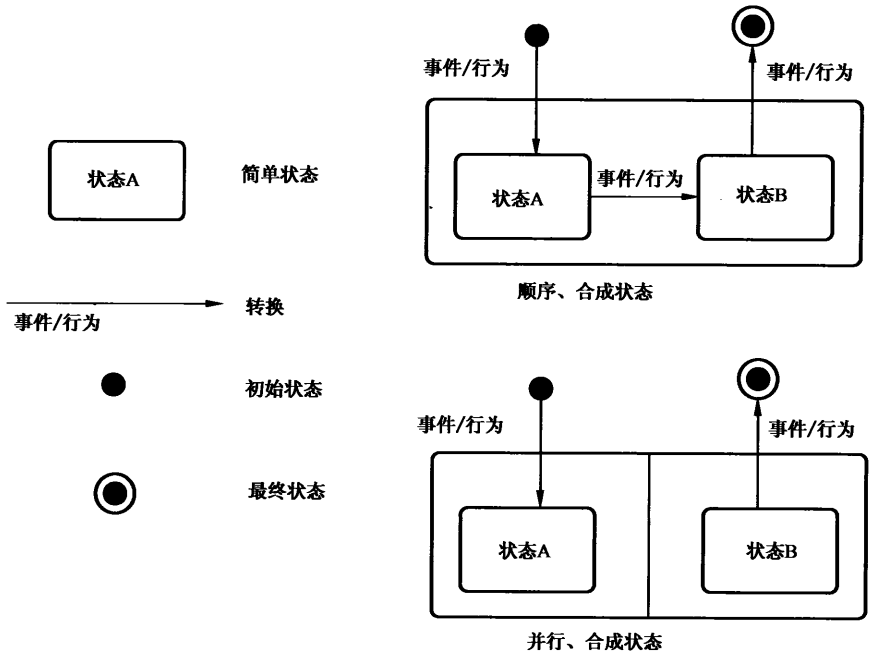

图 2　UML 状态图的符号说明

6　基本服务元素

6.1　概述

本章描述基于网络的要素服务独立于特定操作的，或者是若干操作或接口共同的内容。

6.2　版本号和协商

6.2.1　版本号的格式和值

版本号应按照 OGC 06-121r3:2009 规范中 7.3.1 描述的方式编码，本标准的实现应使用“2.0.0”作为协议版本号。

6.2.2　在服务元数据中和在请求中的格式

符合本标准的 WFS 应在它的服务器元数据中标明版本号为“2.0.0”。一个服务实例可以支持多种版本，客户端可以根据协商规则发现它们的版本值(见 OGC 06-121r3:2009 规范中 7.3.2)。

在客户端对某个基于网络的要素服务实例的请求中使用的版本号应等于这个实例已经声明支持的版本号(除了在 6.2.3 中将要描述的协商外)。如果服务器收到其不支持的版本的请求，这个服务器应抛出一个 InvalidParameterValue(无效参数值)异常。

6.2.3　版本号协商

根据在 OGC 06-121r3:2009 的 7.3.2 描述的规则，所有 WFS 实例应要支持版本协商。

6.2.4　请求编码

本标准定义 WFS 请求的两种编码方法，一是用 XML 作为编码语言，二是用关键字－值对(KVP)编码请求中的多个参数。

示例："REQUEST=GetCapabilities"，其中"REQUEST"是关键字，"GetCapabilities"是值。

KVP编码是XML编码的子集，但是KVP编码不适用于WFS的某些操作，例如Transaction(事务)操作。两种编码方法请求得到的响应或是异常报告都是相同的。

表2列出了本标准中定义的WFS操作和它们适用的编码方法。

表2 操作请求编码

操作	请求编码
GetCapabilities(获取服务能力)	XML & KVP
DescribeFeatureType(描述要素类型)	XML & KVP
GetPropertyValue(获取特性值)	XML & KVP
GetFeature(获取要素)	XML & KVP
GetFeatureWithLock(获取要素并锁定)	XML & KVP
LockFeature(锁定要素)	XML & KVP
Transaction(事务操作)	XML
CreateStoredQuery(创建存储的查询)	XML
DropStoredQuery (删除存储的查询)	XML & KVP
ListStoredQueries(列举存储的查询)	XML & KVP
DescribeStoredQueries(描述存储的查询)	XML & KVP

6.2.5 关键字－值对(KVP)参数编码规则

6.2.5.1 概述

在6.2.5.2和6.2.5.3中描述WFS请求的KVP编码内容。

6.2.5.2 参数顺序和大小写

参数名称不区分大小写，但是参数值要区分大小写。在本标准中，为了印刷上清晰，在KVP编码中参数名称一般使用大写，但并不是硬性要求。

请求中参数的顺序是任意指定的。

WFS要应对遇到不属于本标准参数的情况，如果是依据本标准操作而产生结果，该WFS应忽略这类参数。

6.2.5.3 参数列表

参数都要按照ISO 19143:2010的5.5中的描述进行编码。

示例：在多项连接查询的KVP编码中，括号可以用来间隔多个参数，以便实现这些参数的正确排列。下面示例显示了括号是如何在KVP编码的两个连接查询中使用的。

TYPENAMES=(ns1:F1,ns2:F2)(ns1:F1,ns1:F1)&ALIASES=(A,B)(C,D)&FILTER=(<Filter>…for A,B …</Filter>)(<Filter>…for C,D…</Filter>)

这个KVP编码片段是两个连接查询的编码：第一个查询连接了要素类型"ns1:F1"和"ns2:F2"，它们分别赋予别名"A"和"B"；第二个查询连接了要素类型"ns:F1"和"ns:F1"(一个自联合)，它们分别赋予别名"C"和"D"；"FILTER"(过滤器)参数指定两个过滤器，每个查询对应一个过滤器，由括号分开。所有这些参数值1∶1排列。这个片段等同于下面两个独立的KVP编码片段：

TYPENAMES = ns1:F1,ns2:F2&ALIASES = A,B&FILTER = < Filter >…for A,B…</Filter >

TYPENAMES = ns1:F1,ns1:F1&ALIASES = C,D&FILTER = < Filter >…for C,D…</Filter >

6.3 命名空间

命名空间(见 W3C XML Namespaces)是用来区分不同 XML 文档之间的词汇。WFS 有 4 个标准的命名空间定义,即:

- (http://www.opengis.net/wfs/2.0) - WFS 接口词汇;
- (http://www.opengis.net/gml/3.2) - GML 词汇(见 GB/T 23708—2009);
- (http://www.opengis.net/fes/2.0) - OGC 过滤器词汇(见 ISO 19143 :2010, 5.4);
- (http://www.opengis.net/ows/1.1) - OWS 常见词汇(见 OGC 06-121r3:2009)。

另外,WFS 实现会用到一个或多个 GML 应用模式,这些模式将又会用到一个或多个命名空间(例如,http://www.someserver.com/myns)。尽管本标准中许多示例使用的是单一命名空间,但也允许使用多个命名空间,如同 11.3.3 中所述。

在 XML 编码的请求中所使用的命名空间要用请求的根节点的符号"xmlns:prefix=namespace_uri"(见 W3C XML Namespaces)。

在 KVP 编码的请求中,参数"NAMESPACE"(见 7.6.6)用来声明在请求中使用的任何命名空间。

6.4 服务绑定

本标准主要定义了 WFS 请求和响应消息的编码,该编码独立于任何通信协议。本标准的实现应支持 HTTP GET,HTTP POST 或基于 HTTP POST 之上的 SOAP 等三个协议中的一个。附录 D 包含了服务绑定的具体细节。

7 通用元素

7.1 要素编码

遵循本标准的服务器应对 GML 编码的地理要素进行操作,GML 版本支持 GB/T 23708—2009,但是,本标准的操作也允许使用其他 GML 版本,所以服务器也可能支持其他要素的编码方式。服务器应在其服务能力描述文档中以标准的输入和输出形式声明所有支持的 GML 版本。

服务器也可以支持其他非 GML 要素编码,这些编码在其服务能力描述文档中列出(见表 12),但是,本标准并不描述基于这些编码的操作。

7.2 资源标识符

7.2.1 指定资源标识符

WFS 中的每个要素实例在创建时就赋予了一个不变的唯一资源标识符。

标识符应在 WFS 所有的操作中不发生变化,包括删除操作,这意味着一旦一个标识符被指定,就不能被其他要素重复使用。

资源标识符并不是为了将 WFS 资源和现实世界对象关联起来,因此该值在脱离 WFS 实例的有效范围后就没有了意义。

7.2.2 资源标识符编码

对于用 GML 编码的要素,资源标识符编码为 XML 属性 gml:id(标识符),本标准不描述资源标识符在其他输出格式中的编码方式。

在过滤器表达式中，具体要素实例可以使用元素 fes:ResourceID(资源标识符)来标识。如果这个服务器支持版本控制，要素的具体版本可以通过包含在 fes:ResourceID 元素(见 ISO 19143:2010，7.11.2)中的属性 versionAction(版本行为)来引用。

7.2.3 版本确认

如果服务器支持版本控制，那么该服务器应维护每个要素实例的版本信息。本标准没有对如何维护版本信息作出任何假设。过滤器编码标准中定义的功能允许基于资源标识符进行版本协商(见 ISO 19143:2010，7.11.2)。

7.3 特性引用

7.3.1 XPath 子集

GML 允许地理要素具有复杂的或组合的非几何特性，由此引发了应该如何在多个应用到引用值的地方(如查询和过滤器表达式中)引用该特性复杂值的组成部分的问题。当 WFS 提供的要素内容是用 XML 编码时，该 WFS 应使用 XPath[见 W3C XML Path Language (W3C XML 路径编码语言)]表达式来引用要素的特性和特性值的组成部分。ISO 19143:2010 中 7.4.4 描述了服务器应支持的必要的 XPath 最小子集。

支持 schema-element() (模式元素)的 XPath 函数是可选的，但是，如果服务器遵守继承的一致性类(见表 1)，服务器也应支持 schema-element() 函数(见 ISO 19143:2010，7.4.4)。

7.3.2 访问函数

在 GML 中，要素特性的值可以是成行编码的值，也可以是通过一个简单的 XLink(见 GB/T 23708—2009，7.2.3)引用的值。这意味着在使用 XPath 表达式的过程中，服务器可能需要解决资源问题。为适应这个需要，所有 WFS 的实现应提供 XPath 访问函数 wfs:valueOf( )的具体实现。

该函数的参数是某要素的特性名称，而该函数的响应即是这个指定特性的值。响应可能是该指定特性的值的一个文本节点，也可能是一组元素节点。

该函数适用于一个服务器上所有被引用的本地资源，并且，如果该服务器在其服务能力描述文档中(见表 13)发布该函数功能，那么这个函数也要适用于所有远程资源。如果该服务器只适用于本地资源，当它遇到远程资源时，应抛出一个 OptionNotSupported(不支持的选项)异常(见 OGC 06-121r3:2009，表 25)。

在 XPath 表达式中何时使用 valueOf()函数是由服务器应用模式决定的。如果为了允许使用引用值而声明一个特性(见 GB/T 23708—2009，7.2.3.3、7.2.3.7)，那么 wfs:valueOf()函数宜在 XPath 表达式中指定(见 B.2.2、B.4.5)。

7.4 谓词表达式编码

本标准定义的许多操作都包含谓词表达式，用来指定所操作要素的子集。谓词表达式可以列举要操作的一组具体要素集，也可以通过对要素类的特性和特性值明确限定来指定一组要素。

基于 XML 编码的谓词表达式应用 ISO 19143:2010 中 7.2 描述的元素 fes:Filter 来编码。

基于 KVP 编码的谓词表达式应用 ISO 19143:2010 中表 2 描述的参数来编码。

一个基于 WFS 实现的具体要素谓词集应在使用过滤器能力部分的服务能力描述文档中进行声明(见 8.3.3)。

本标准的所有实例至少应实现查询一致性类操作(见 ISO 19143:2010 中表 1)。

本标准实现基础 WFS 一致性类(见表 1)的所有实例也应实现最小空间过滤器的一致性类(见 ISO

19143:2010 中表 1)。

7.5 异常报告

当 WFS 处理请求时遇到错误或是收到一个不可识别的请求,它将会生成一个 XML 文档,指明有错误发生。XML 错误响应的格式在 OWS 通用实现规范[OGC 06-121r3:2009]第 8 章定义的异常响应语义中指定,并且应经过有效性验证。

ows:ExceptionReport (异常报告)元素可以包含一个或多个用 ows:Exception(异常)指定的 WFS 操作异常。必选属性 version(版本)用来标明服务异常报告模式的版本号。本标准的该版本号为"2.0.0"。可选属性 language(语言)用来标明所用语言,语言参数的代码列表在 IETF RFC 4646 中定义。

单个的异常信息包含在 ows:ExceptionText(异常文本)元素中。必选属性 code(代码)用来把异常代码与相应的信息关联在一起。

可选属性 locator(定位符)用来指出异常在请求过程中发生的位置,表 3 说明了每个异常代码对应的 locator 参数值。

表 3 WFS 异常编码

exceptionCode (异常编码)值	编码语义	"locator"值	一致性类
CannotLockAllFeatures (不能锁定所有要素)	一个 lockAction(锁定行为)值为 ALL(全部)的锁定请求不能对所有请求的要素进行锁定	如果操作包含可选"handle"(句柄)参数,报告"handle"参数值作为"location"(位置)参数值	锁定 WFS
DuplicateStoredQueryValue (存储的查询值已存在)	为一个存储的查询表达式定义的标识符重复了	"locator"参数应包含重复的标识符值	管理存储的查询
DuplicateStoredQueryParameterName (存储的查询参数名已存在)	为一个存储的查询参数所指定的名称已经在同样的存储的查询定义中使用了	"locator"参数应列出重复的存储的查询参数的名称	管理存储的查询
FeaturesNotLocked (不能锁定要素)	对于不支持自动数据锁定(见 15.2.3.1)的服务器,这个异常表示一个事务操作正在修改一个要素,此要素之间没有使用 LockFeature(见第 12 章)或 GetFeatureWithLock(见 13 章)操作对其锁定	如果操作包含可选的"handle"参数,那么"location"参数的值为"handle"参数值	锁定 WFS
InvalidLockId (无效锁标识符)	由于一个事务操作的 lockId(锁标识符)参数值不是由服务器所生成的,所以无效	"location"参数应包含这个无效的 lockId 参数值	锁定 WFS
InvalidValue (无效值)	一个 Transaction(见第 15 章)试图使用不符合要素模式的方法插入或修改一个数据组件的值	"locator"应包含错误修改的特性名称	事务 WFS
LockHasExpired (锁已经过期)	一个 Transaction 或者 LockFeature 操作指定的锁标识符已过期而不再有效	"locator"参数应包含过期的锁标识符	锁定 WFS

表 3（续）

exceptionCode（异常编码）值	编码语义	“locator”值	一致性类
OperationParsingFailed（操作解析失败）	请求格式不对，不能为服务器解析	如果可以使用“handle”参数值那么“locator”参数应包含“handle”参数值，否则包含错误形式表示的操作名称	全部（见表 1）
OperationProcessingFailed（操作处理失败）	处理操作时遇到错误	如果可以使用“handle”参数，那么“locator”参数应包含“handle”参数值，否则包含失败操作的名称	全部（见表 1）
ResponseCacheExpired（响应缓存过期）	用于支持分页的响应缓存已经过期，并且结果不再有效	如果操作包含可选的“handle”参数，那么“locator”参数值为“handle”参数值	响应分页

服务应为本标准中定义的所有一致性类，实现在 OGC 06-121r3:2009 中表 25 定义的通用异常代码。

注：在单个异常报告中，可能报告多个异常，在这种情形下，服务实例应尽可能报告尽量多的异常，以便清楚地描述一个问题。

示例：如果由于参数的值无效而导致解析操作失败，那么该服务应报告一个 OperationParsingFailed 异常和一个 InvalidParameterValue（无效的参数值）异常。

附录 B 中包含有异常报告的示例。

7.6 通用请求参数

7.6.1 概述

7.6.2 到 7.6.6 阐述本标准中定义的 WFS 操作的通用请求参数。

7.6.2 基本请求类型

7.6.2.1 请求语义

除了 GetCapabilities 操作外，所有 WFS 操作都继承自一个抽象的基本请求类型（图 3）。

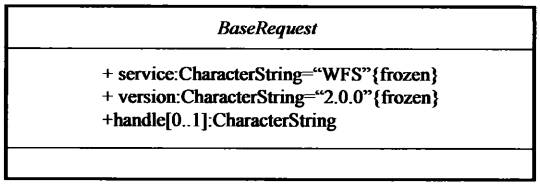

图 3 基本请求

7.6.2.2 XML 编码

下面的 XML 模式片段详述了 BaseRequest 的 XML 编码。

```
<xsd:complexType name = "BaseRequestType" abstract = "true">
  <xsd:attribute name = "service" type = "xsd:string" use = "required" fixed = "WFS"/>
```

```
    <xsd:attribute name = "version" type = "xsd:string"use = "required" fixed = "2.0.0"/>
    <xsd:attribute name = "handle" type = "xsd:string"/>
  </xsd:complexType >
```

7.6.2.3 KVP 编码

表 4 定义了 BaseRequest 类型的 KVP 编码。

必选关键字 REQUEST(请求)的值指明调用的服务操作。

注：XML 编码的请求没有参数 REQUEST 是因为根元素的名称已经指明了调用的服务操作的名称。

表 4 基本请求类型的 KVP 编码

URL 关键字	操作	O/M[a]	描述
SERVICE (服务)	所有操作	M	见 7.6.2.4
VERSION[b](版本) (所有操作)	除 GetCapabilities 之外的所有操作	M	见 7.6.2.5
[a] O=Optional(可选);M=Mandatory(必选)。 [b] VERSION 对于除 GetCapabilities 操作之外的所有操作都是必选的。			

7.6.2.4 service 参数

必选参数 service 是用来指明正在调用的特定服务器中可用的服务类型，当调用 WFS 时，参数 service 的值应是“WFS”。

在 XML 编码中，这个参数应用一个名为 service 的属性来编码(见 7.6.2.2)。

在 KVP 编码中，这个参数应用关键字 SERVICE 来编码的(见 7.6.2.3)。

7.6.2.5 version 参数

所有的 WFS 请求(除 GetCapabilities 操作外)应包含一个命名为 version 的参数，用来指明请求编码遵从 WFS 规范的哪一个版本，并且用在 6.2.3 描述的版本协商中。当依照本标准对一个 WFS 请求进行编码时，所属版本的值应固定为 2.0.0，与本标准的版本相对应。

在 XML 编码中，这个参数应用一个名为 version 的属性来编码(见 7.6.2.2)。

在 KVP 编码中，这个参数使用关键字 VERSION 来编码的(见 7.6.2.3)。

7.6.2.6 handle(句柄)参数

参数 handle 在请求编码中是可选参数，目的是为了让一个客户端应用将一个容易记忆的名称和一个请求关联起来，以便进行错误处理。

在 XML 编码中，这个参数应用一个名为 handle 的属性来编码(见 7.6.2.2)。

在 KVP 编码中，没有定义这个参数。

如果客户端为一个操作指定了一个 handle 并且遇到了异常(见 7.5)，为了明确发生异常的操作或行为，WFS 应将属性 handle 的值赋给 ows:ExceptionText 元素的属性 locator(见 OGC 06-121r3:2009 的第 8 章)。如果没有指定 handle 参数，该服务器应忽略 ows:ExceptionText 元素中的属性 locator，或者可以使用其他的方法，例如行的序号，来定位操作中的异常。当属性 handle 与 Transaction 操作共同使用时(见 15 章)，handle 属性非常有用，Transaction 操作中可能包含多种功能，为每个功能指定一个 handle 可以让服务器准确地定位 Transaction 操作中的异常。

7.6.3 StandardPresentationParameters(标准表达参数)

7.6.3.1 参数语义

StandardPresentationParameters(标准表达参数)(见图 4)是用来控制如何在响应文档中表达查询结果。这些参数可能出现在 GetPropertyValue(见第 10 章)、GetFeature(见第 11 章)和 GetFeature-WithLock(见第 13 章)操作中。

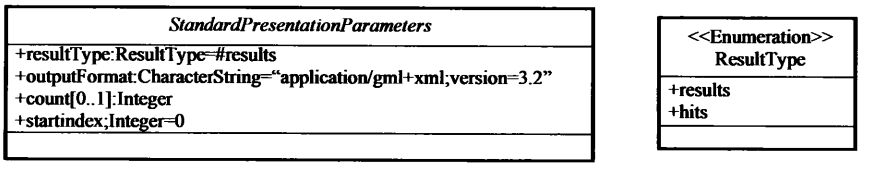

图 4 StandardPresentationParameters

7.6.3.2 XML 编码

下面的片段定义了 StandardPresentationParameters 的 XML 编码:

```
<xsd:attributeGroup name = "StandardPresentationParameters">
  <xsd:attribute name = "startIndex"
    type = "xsd:nonNegativeInteger" default = "0"/>
  <xsd:attribute name = "count"
    type = "xsd:nonNegativeInteger"/>
  <xsd:attribute name = "resultType"
    type = "wfs:ResultTypeType" default = "results"/>
  <xsd:attribute name = "outputFormat"
    type = "xsd:string" default = "applicaton/gml + xml; version = 3.2"/>
</xsd:attributeGroup>
<xsd:simpleType name = "ResultTypeType">
  <xsd:restriction base = "xsd:string">
    <xsd:enumeration value = "results"/>
    <xsd:enumeration value = "hits"/>
  </xsd:restriction>
</xsd:simpleType>
```

7.6.3.3 KVP 编码

表 5 定义了标准表达参数的 KVP 编码。

表 5 标准表达参数的 KVP 编码

URL 关键字	操作	O/M[a]	默认值	描述
STARTINDEX (起始索引)	GetPropertyValue, GetFeature, GetFeatureWithLock	O	1	见 7.6.3.4
COUNT (计数)	GetPropertyValue, GetFeature, GetFeatureWithLock	O	1	见 7.6.3.5
OUTPUTFORMAT (输出格式)	DescribeFeatureType, GetPropertyValue, GetFeature, GetFeatureWithLock	O	application/gml+xml; version=3.2	见 7.6.3.7
RESULTTYPE (结果类型)	GetPropertyValue, GetFeature, GetFeatureWithLock	O	results(结果)	见 7.6.3.6

[a] O=Optional(可选); M=Mandatory(必选)。

7.6.3.4 startIndex 参数

可选参数 startIndex 可以用来指明在响应文档中服务器应开始表达的结果在结果集中的索引位置。

在 XML 编码的请求中,这个参数应用一个名为 startIndex(见 7.6.3.2)的属性来编码。

在 KVP 编码的请求中,这个参数使用关键字 STARTINDEX(见 7.6.3.3)来编码。

7.6.3.5 count 参数

可选参数 count 可以用来限定在响应文档中一个请求包含的明确请求的值的数量(例如要素或特性值)。只有用被明确请求的类型的值作为 typeNames(类型名称)参数的值(见 7.9.2.4.1)时,才能计入总数中,而包含在该明确请求的值类型中嵌套的值不参与计数。

在 XML 编码的请求中,这个参数应用一个名为 count(见 7.6.3.2)的属性来编码。

在 KVP 编码的请求中,这个参数是使用关键字 COUNT(见 7.6.3.3)来编码。

示例:在以下 XML 部分中要素总数为 4。嵌入在要素 gml:id="4"中的对 gml:id="2"要素的引用和要素 gml:id="5"不包含在总数中。

```
<? xml version = "1.0"? >
< wfs:FeatureCollection
  timeStamp = "2010 - 08 - 01T22:47:02"
  numberMatched = "4" numberReturned = "4"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema - instance"
  xsi:schemaLocation = "http://www.someserver.com/myns ./myns.xsd
                http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
```

```
                http://www.opengis.net/gml/3.2
                http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    <wfs:member>
      <myns:Feature gml:id="1">
      ...
      </myns:Feature>
    </wfs:member>
    <wfs:member>
      <myns:Feature gml:id="2">
      ...
      </myns:Feature>
    </wfs:member>
    <wfs:member>
      <myns:Feature gml:id="3">
      ...
      </myns:Feature>
    </wfs:member>
    <wfs:member>
      <myns:Feature gml:id="4">
        <myns:Property1> ... <myns:Property1>
        <myns:Property2> ... <myns:Property2>
        <myns:Property3 xlink:href="#2"/>
        <myns:Property4>
          <myns:Feature gml:id="5">
          ...
          </myns:Feature>
        </myns:Property4>
      </myns:Feature>
    </wfs:member>
</wfs:FeatureCollection>
```

在连接查询时(见 7.9.2.5.3.1),明确请求的值类型的每个元组整体计数为 1,而每个元组内部包含的值成员不参加计数。

计数的值应用于整个结果集(即,处理一个或多个查询操作产生的结果集),并且这个计数约束是按照各查询结果出现的顺序记录的,一旦达到了计数的限值,请求操作可能中断,包含最多计数值的响应文档发送给客户端。

计数参数没有预定义的默认值,如果没有该参数则意味着结果集的所有值都要将服从服务器的配置限定提供给客户端。如果某服务器的确配置了计数限值,那么将会在服务能力描述文档中的 Count-Default(计数缺省值)参数显示该计数限值(见表 14)。

7.6.3.6 resultType 参数

在 XML 编码的请求中,这个参数应用一个名为 resultType(见 7.6.3.2)的属性来编码。

在 KVP 编码的请求中,这个参数使用关键字 RESULTTYPE(见 7.6.3.3)来编码。

WFS 能够通过两种方式响应一个请求操作(异常报告除外):既可以生成一个包含满足请求操作的资源的完整响应文档,也可以简单地生成一个空的响应容器,指明请求操作返回资源的总数的计数。

WFS 具体生成哪种响应方式取决于可选参数 resultType 的值。

这个参数可能的值为“results”(结果)和“hits”。

如果参数 resultType 的值设置为“results”,表明服务器要生成一个完整的文档,包含满足操作的资源,而且该响应文档的根元素应包含其实际表达的资源数量的计数(见 7.7.4.3),响应容器的根元素应同时包含操作实际上找到的资源总数的计数,这个总数总是等于或大于响应文档表达的资源数量(见7.7.4.2)。

如果参数 resultType 的值设为“hits”,表明服务器生成一个不包含任何资源实例的空的响应文档,并且响应容器的根元素应包含操作找到的资源总数的计数(见 7.7.4.2),响应文档(见 7.7.4.3)中表达的资源数量的值应设为 0。

7.6.3.7 outputFormat 参数

可选参数 outputFormat 指明了查询操作响应中资源的编码格式。

在 XML 编码的请求中,这个参数应用一个名为 outputFormat(见 7.6.3.2)的属性来编码。

在 KVP 编码的请求中,这个参数用关键字 OUTPUTFORMAT(见 7.6.3.3)来编码。

该参数的默认值为“application/gml+xml; version=3.2”,表明响应文档中的资源会使用版本号为 3.2 的 GML(见 GB/T 23708—2009)进行编码。每个遵循本标准的 WFS 应支持这个默认值。

一个服务器在其能力描述文档(见 8.3)中可能为参数 outputFormat 声明其他值,表明支持包括 GML 旧版本在内的多种输出形式。但是本标准并不涉及 outputFormat 参数的任何其他值的详细含义。在服务能力描述文档指定其他 outputFormat 参数值的情况下,本标准建议对于每个值都有一个描述性的说明。

7.6.4 StandardResovleParameters(标准解析参数)

7.6.4.1 参数语义

遵循本标准的服务器应实现解析本地资源引用的功能,服务器根据请求可以选择性地实现解析远程资源引用的功能,并且要通过使用服务能力描述文档中的 ImplementsRemoteResolve(实现远程解析)来申明实现了该功能(见表 13)。

正如 7.6.4.2～7.6.4.7 中描述,服务器如何处理资源引用是由参数 resolve(解析)、resolveDepth(解析深度)和 resolveTimeout(解析终止时间)来控制的。

StandardResovleParameters(标准解析参数)参数(图 5)可以出现在 GetPropertyValue(见第 10 章)、GetFeature(见第 11 章)和 GetFeatureWithLock 操作(见第 13 章)中。

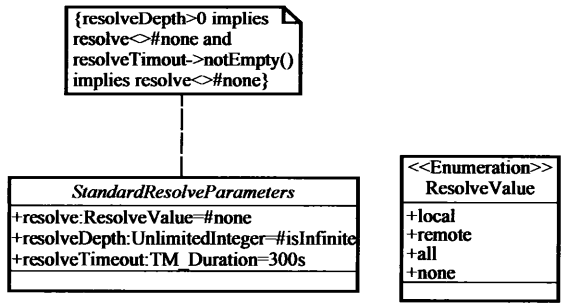

图 5 StandardResolveParameters

7.6.4.2 XML 编码

下面的代码定义了标准解析参数集的 XML 编码。

```
<xsd:attributeGroup name = "StandardResolveParameters">
  <xsd:attribute name = "resolve"
                 type = "wfs:ResolveValueType" default = "none"/>
  <xsd:attribute name = "resolveDepth"
                 type = "wfs:positiveIntegerWithStar" default = " * "/>
  <xsd:attribute name = "resolveTimeout"
                 type = "xsd:positiveInteger" default = "300"/>
</xsd:attributeGroup>
<xsd:simpleType name = "ResolveValueType">
  <xsd:restriction base = "xsd:string">
    <xsd:enumeration value = "local"/>
    <xsd:enumeration value = "remote"/>
    <xsd:enumeration value = "all"/>
    <xsd:enumeration value = "none"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "positiveIntegerWithStar">
    <xsd:union memberTypes = "xsd:positiveInteger wfs:StarStringType"/>
</xsd:simpleType>
<xsd:simpleType name = "StarStringType">
    <xsd:restriction base = "xsd:string">
      <xsd:enumeration value = " * "/>
    </xsd:restriction>
</xsd:simpleType>
```


7.6.4.3 KVP 编码

表 6 定义了标准的 resolve 参数的 KVP 编码。

表 6 标准解析参数的 KVP 编码

URL 关键字	操作	O/M[a]	默认值	描述
RESOLVE（解析）	GetPropertyValue，GetFeature，GetFeatureWithLock	O	None	见 7.6.4.4
RESOLVEDEPTH（解析深度）	GetPropertyValue，GetFeature，GetFeatureWithLock	O	*	见 7.6.4.5 RESOLVE 参数应包含一个非“none”的值
RESOLVETIMEOUT（解析终止时间）	GetPropertyValue，GetFeature，GetFeatureWithLock	O	由服务器指定（见 ResolveTimeoutDefault 表 14）	见 7.6.4.6 RESOLVE 参数应包含一个非“none”的值
[a] O=Optional(可选)；M=Mandatory(必选)。				

7.6.4.4 resolve 参数

可选的 resolve 参数控制一个操作是否可以解析资源引用，并且指定要解析的资源引用（即本地或

远程)。

在 XML 编码的请求中,这个参数应用一个名为 resolve 的属性(见 7.6.4.2)来编码。

在 KVP 编码的请求中,这个参数是用关键字 RESOLVE 来编码(见 7.6.4.3)。

resolve 参数的值域有:"local"(本地)、"remote"(远程)、"all"(全部)或"none"(没有)。根据服务器的实现水平,可以支持该值域中的某些值或所有值,具体支持哪些 resolve 参数值需要在服务器的服务能力描述文档中进行申明(见 8.3.3)。

resolve 参数值"remote"指操作只能解析远程资源引用。

resolve 参数值"local"指操作只能解析本地资源引用。

resolve 参数值"all"指操作可以解析所有资源引用。

resolve 参数值"none"指操作不能解析任何资源引用,这也是在没有指定 resolve 参数时的缺省值。

7.6.4.5 resolveDepth 参数

可选参数 resolveDepth 表明在响应文档中嵌套资源引用会解析到什么程度。

在 XML 编码的请求中,这个参数应用一个名为 resolveDepth 的属性来编码。

在 KVP 编码的请求中,这个参数是用关键字 RESOLVEDEPTH 来编码。

这个参数的有效值范围包括正整数和"*"。

只有当包含参数 resolveDepth 的操作实现了参数 resolve,且参数 resolve 的值没有被设置为"none"时,参数 resolveDepth 的值才能被使用。

如果参数 resolve 没有指定值或者设置为"none",那么服务器将会忽略参数 resovleDepth 的任何值。

如果参数 resolveDepth 的值指定为"0",那么服务器不会解析任何资源引用。

如果参数 resolveDepth 的值指定为"1",那么服务器会解析即时资源引用,并且将它的值包含在响应文档中;但是如果这个解析的资源包含任何嵌套的资源引用,这些嵌套资源则不会被解析。

如果参数 resolveDepth 的值为"*",那么服务器会解析所有的即时资源引用和所有嵌套的资源引用。

当解析完指定 resolveDepth 的引用,服务器将让所有更深层次嵌套的引用重定位到它们对应的资源。

示例 1:在某个服务器上 WFS 的数据库中的本地资源的引用可能会在响应文档中当作远程资源引用到该服务器上。

一旦服务器在解析完指定的 resolveDepth 之前发现循环引用的情况,该服务器会终止进一步解析嵌套资源,并将最后解析到的该资源引用链上的引用作为已经解析完成的相应资源。

示例 2:以下面资源引用链:A -> B -> C -> D -> E -> B 为例。如果 resolveDepth 的值设置为 8,那么服务器对所有资源的解析到 E 为止(也就是第四级),然后将资源 E 指向已经解析过的资源 B。此时,因为由于循环引用的原因接下来一直到第 8 级的资源都已经解析过,所以服务器将终止资源解析。

7.6.4.6 resolveTimeout 参数

可选参数 resolveTimeout 控制一个服务器在解析资源引用时应等待多长时间来接收响应。

在 XML 编码的请求中,resolveTimeout 参数应用一个名为 resolveTimeout 的属性来编码(见 7.6.4.2)。

在 KVP 编码的请求中,resolveTimeout 参数 RESOLVETIMEOUT 关键字来编码(见 7.6.4.3)。

resolveTimeout 参数是 xsd:positiveInteger(正整数)类型,使用秒来指定终止时间。如果没有指定 resolveTimeout 参数,那么服务器等待时间与实现相关,并在服务能力描述文档中使用 ResolveTimeoutDefault(缺省的解析终止时间)约束(见表 14)声明等待时间。

只有当包含 resolveTimeout 参数的请求元素中指定了 resolve 参数，并 resolve 参数值没有被设置为“none”，那么才能使用 resolveTimeout 参数指定的值。

如果 resolve 参数的值设置为“none”，那么服务器将忽略所有为 resolveTimeout 参数指定的值。

7.6.4.7 无法解析的引用

当服务器无法解析资源引用时，服务器仅仅在响应中报告无法解析的原始 URI，但这种情况并不是一种异常。

7.6.5 StandardInputParameters(标准输入参数)

7.6.5.1 参数语义

StandardInputParameters(标准输入参数)(见图 6)是一组参数，用来确定在输入时资源的编码方式和这些资源可能包含的任何要素几何体的 CRS。

这些参数可以指定为 Transaction 操作(见第 15 章)中的 Insert(插入)(见 15.2.4)、Update(更新)(见 15.2.5)和 Replace(替换)(见 15.2.6)操作。

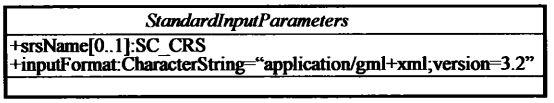

图 6 StandardInputParameters

7.6.5.2 XML 编码

以下 XML 片段定义了标准输入参数的 XML 编码：

```
<xsd:attributeGroup name = "StandardInputParameters">
    <xsd:attribute name = "inputFormat" type = "xsd:string"
      default = "application/gml + xml; version = 3.2"/>
    <xsd:attribute name = "srsName" type = "xsd:anyURI"/>
</xsd:attributeGroup>
```

7.6.5.3 KVP 编码

因为 KVP 编码在 Transaction 操作(见第 15 章)中没有定义，所以 KVP 编码的请求没有定义标准输入参数。

7.6.5.4 inputFormat(输入格式)参数

在 XML 编码的请求中，inputFormat 参数应用一个名为 inputFormat 的属性(见 7.6.5.2)来编码。

在 KVP 编码的请求中，没有定义 inputFormat 参数。

当使用 Transaction 操作中的 Insert 操作(见 15.2.4.1)输入新元素时，或者使用 Update 操作(见 15.2.5.1)或 Replace 操作(见 15.2.6.1)更新要素时，可使用 inputFormat 参数来确定描述要素的要素编码方式。

对于实现了事务 WFS 一致性类(见表 1)的服务器，属性 inputFormat 的默认值为“application/gml+xml; version=3.2”，表明使用版本号为 3.2 的 GML(见 GB/T 23708—2009)编码输入要素。

一个服务器可能在能力描述文档(见 8.3)中声明 inputFormat 参数的其他值，表明支持包括 GML 旧版本在内的多种输入格式，但是本标准不涉及 inputFormat 参数的任何其他值的详细含义。在服务

能力描述文档指定其他 inputFormat 参数值的情况下，本标准建议对于每个值都有一个描述性的说明。

7.6.5.5 srsName(空间参照系名称)参数

在 XML 编码的请求中，srsName 参数应用一个名为 srsName 的属性(见 7.6.5.5)来编码。

在 KVP 编码的请求中，不支持标准输入参数 srsName。

srsName 参数可指定为 Transaction 操作(见第 15 章)的一个参数，或者事务操作 Insert(见 15.2.4.1)、Update(见 15.2.5.1)或 Replace(见 15.2.6.1)的一个参数。

如果 srsName 参数指定为 Transaction 操作的一个参数，那么 srsName 属性值将采用默认的 CRS 编码 Transaction 中的要素几何体(见 15.2.2)。

如果一个 srsName 值被指定为 Transaction 操作中的 Insert、Update 或 Replace 的一个参数，那么该值将取代在 Transaction 操作中使用 srsName 参数指定的任何 CRS 值。

srsName 参数(见 ISO 19143:2007，10.1.3.1)也可以用来确定单个几何体的 CRS 和替换之前确定的 CRS。

如果指定了 srsName 参数，那么 srsName 参数值等同于在服务能力描述文档(见 8.3)中使用 wfs:DefaultCRS 元素指定的默认 CRS 值，或者等同于有关要素类型的任何一个 wfs:OtherCRS 值(参见表 11)。

如果指定的要素类型不支持指定的 CRS，那么 WFS 将抛出一个 InvalidParameterValue(无效参数值)异常(见表 13)。

如果没有指定 srsName 参数，那么 WFS 将使用服务能力描述文档(见 8.3)中 wfs:DefaultCRS 元素指定的默认 CRS 对几何形状进行编码。

7.6.6 KVP 编码的请求中其他通用关键字

表 7 定义了 KVP 编码的 WFS 请求的其他关键字。

表 7 其他 KVP 编码的 WFS 请求的通用关键字

URL 关键字	操作	O/M[a]	描　述
NAMESPACES (命名空间)	所有操作	O	用于指明多个命名空间和它们的前缀。格式应是 *xmlns(prefix, escaped_url)*，此式中 *escaped_url* 在[OGC 06-121r3:2009]中有定义(见 11.3)。如果没有指定前缀，就使用缺省的命名空间。多个命名空间可以通过逗号分隔的 xmlns()值来限定。 因为 XML 有其他声明命名空间的机制(见 6.3)，所以这个参数不是由 XML 编码的请求定义
Vendor-specific parameters (提供商指定参数)		O	一个服务器可以实现不是本标准的其他 KVP 参数，也就是提供商指定参数(Vendor-specific parameters，VSP)。VSP 允许提供商实现其他可以加强请求结果的参数。一个服务器应在 VSP 缺失、变形或者 VSP 不能解析的情况下生成有效的结果。未知 VSP 应被忽略。 一个服务器可以选择不声明部分或者全部 VSP。如果 VSP 包含在服务能力 XML(见 8.3)中，那么 ows:ExtendedCapabilities(扩展的能力)元素(见 OGC 06-121r3:2009，7.4.6)应进行相应的扩展。可以引入包含 ows:ExtendedCapabilities 元素的其他模式文档。任何被声明的 VSP 应包含或者引用其他描述其含义的元数据(见 8.4)。WFS 实例应选择 VSP 名称，以使其名称不与本标准定义的 WFS 参数有冲突

[a] O=Optional(可选)；M=Mandatory(必选)。

7.7 StandardResponseParameters(标准响应参数)

7.7.1 参数语义

StandardResponseParameters(标准响应参数)(见图 7)是用来定义 GetPropertyValue(见第 10 章)、GetFeature(见第 11 章)和 GetFeatureWithLock(见第 13 章)三个操作的结果集的响应参数。

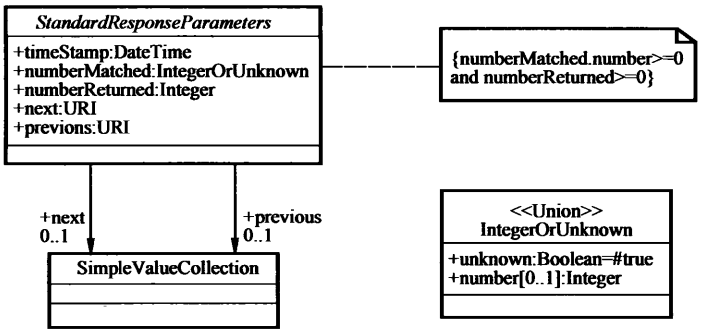

图 7 StandardResponseParameters

7.7.2 XML 编码

下面的 XML 模式片段定义了标准响应参数的 XML 编码。

```
<xsd:attributeGroup name = "StandardResponseParameters">
    <xsd:attribute name = "timeStamp" type = "xsd:dateTime" use = "required"/>
    <xsd:attribute name = "numberMatched" type = "xsd:nonNegativeIntegerOrUnknown" use
= "required"/>
    <xsd:attribute name = "numberReturned" type = "xsd:nonNegativeInteger" use = "re-
quired"/>
    <xsd:attribute name = "next" type = "xsd:anyURI"/>
    <xsd:attribute name = "previous" type = "xsd:anyURI"/>
</xsd:attributeGroup>
<xsd:simpleType name = "nonNegativeIntegerOrUnknown">
    <xsd:union>
      <xsd:simpleType>
        <xsd:restriction base = "xsd:string">
          <xsd:enumeration value = "unknown"/>
        </xsd:restriction>
      </xsd:simpleType>
      <xsd:simpleType>
        <xsd:restriction base = "xsd:nonNegativeInteger"/>
      </xsd:simpleType>
    </xsd:union>
</xsd:simpleType>
```

7.7.3 KVP 编码

WFS 的响应消息只有 XML 编码。

7.7.4 参数讨论

7.7.4.1 timeStamp(时间戳)参数

当一个结果集产生时,WFS 将使用 timeStamp 参数来标记时间和日期。

7.7.4.2 numberMatched(匹配个数)参数

numberMatched 参数在响应文档中用来表明结果集中与操作所请求的类型相匹配的要素或值的总个数,这个参数值不一定和响应文档中实际出现的要素或值的计数相等(见 7.7.4.3)。如果服务器不能给出匹配的要素或值的个数,那么它应使用值"unknown"(未知)来说明这一点。

通过将参数 resultType(见 7.6.3.6)设置为"hits",不必返回结果集就可以得到操作返回的要素或值的总数。对于支持响应分页的服务,需要设置下一个参数的值,这样能得到响应要素或值的第一个子集。

7.7.4.3 numberReturned(返回个数)参数

numberReturned 参数是指响应文档中出现的要素或值的总数。

注:见 7.6.3.4 介绍的计数规则。

如果确定了参数 count(见 7.6.3.5)的值,那么参数 numberReturned 等于或小于参数 count 的指定值。

如果操作中没有明确指定参数 count 的值,那么在服务能力描述文档(见 8.3.3)中可以配置和声明一个默认值,这和在操作中指定 numberReturned 的值的效果是一样的。

如果操作中没有明确指定参数 count 的值,也没有在服务器配置中设置 count 的值,那么响应参数 numberReturned 的值则等于响应参数 numberMatched 的值。

7.7.4.4 Response paging(分页响应)

7.7.4.4.1 概述

分页响应是指客户端能够在一组返回的要素或值中一次滚动显示 N 个要素或值,这与在搜索引擎的结果中一次滚动显示一页的效果一样。

支持分页响应的服务器需要在能力描述文档中使用 ImplementsResultPaging(实现结果分页)约束(见表 13)声明这种能力。

分页响应是通过返回集(见 10.3.1,11.3.1)定义的 previous(上一页)参数和 next(下一页)参数来实现的。

为了触发分页操作,可以在请求中设置 count 参数(见 7.6.3.5),也可以在服务能力描述文档(见表 14)中声明并在服务器中实现这些参数。

previous 或 next 属性的值是由服务器 URI 生成的,URI 主要用于检索相关数据集或结果。这些 URI 的具体格式依赖于服务器如何缓存操作结果,以便操作结果能以一次一个子集的形式向客户端传送。

在处理 previous 或 next 这两个 URI 时,服务器要么生成一个包含上一组或下一组要素或值(或者连接查询情况下的连接元组)的有效响应,要么生成异常消息以说明这些结果集不存在(可能因为客户端进入上一个或下一个链接的等待时间太长而导致服务器中的缓存结果被清除)。在该情况下,服务器应生成一个 ResponseCacheExpired 异常(见表 3)。

因为分页响应需要在能力描述文档中使用 ResponseCacheTimeout 约束(见表 14),所以服务器要说明结果集缓存的时间。

示例：下面用 GetFeature(见第 11 章)请求来说明这些参数的用法。

```
<GetFeature service="WFS" version="2.0.0" count="100"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:cw="http://www.someserver.com/cw"
  xmlns:fes="http://www.opengis.net/ogc/1.1"
  xmlns:gml="http://www.opengis.net/gml/3.2">
  <Query typeNames="cw:MyFeatureType"/>
</GetFeature>
```

与服务器的互操作过程如下：

a) 客户端向支持分页响应的服务器发送请求。

b) 服务器返回包含结果集的前 100 个结果的 wfs:FeatureCollection 元素。设置 next 属性以让客户端可以检索接下来的 100 个要素，由于这是结果集中的第一组要素，所以没有设置 previous 属性。

c) 客户端遍历“next”URI。

d) 服务器返回另一个包含结果集中接下来 100 个结果的 wfs:FeatureCollection 元素。在这种情况下，previous 和 next 属性都要设置，这样客户端就可以检索步骤 a)中 GetFeature 请求的结果集中的前 100 个要素或后 100 个要素。

e) 客户端继续遍历每个“next”URI，直到 wfs:FeatureCollection 元素中没有设置 next 属性。这说明步骤 a)中的 GetFeature 请求的结果集的最后一组结果已经被检索到了。

f) 同样，客户端遍历“previous”URI 直到 wfs:FeatureCollection 元素中没有设置 previous 属性，这说明第一组的 100 个要素已经被检索到了。

7.7.4.4.2 分页响应的事务一致性

7.7.4.4.2.1 声明事务一致性

服务器在能力描述文档中要说明是否使用 PaginglsTransactionSafe 约束(见表 14)来支持分页响应的事务一致性。

7.7.4.4.2.2 支持事务一致性的分页响应

声明事务一致性的服务器应保持分页操作之间的事务一致性。这样就能保证客户端分页查看结果集时的数据内容与相应请求实现的次数保持一致。

7.7.4.4.2.3 不支持事务一致性的分页响应

没有声明事务一致性的服务不要求保持分页操作之间的事务一致性或状态。这样在分页操作之间可能会在完整的结果集内添加新的要素，更新或删除要素，因而可能会跳过或重复结果集元素。

7.8 使用 schemaLocation(模式位置)属性

为了保证结果的有效性，WFS 生成的任何 GML 文档都要引用一个合适的 GML 应用模式文档。正如 XML 模式规范(见 W3C XML Schema 第 1 部分)中的定义，需要使用 schemaLocation 属性来完成。

示例：下面的 XML 片段说明了在根元素中使用 schemaLocation 属性来说明用来证实 XML 模式文档有效性的位置。

```
<? xml version="1.0" ? >
<wfs:FeatureCollection
```

```
    xmlns = "http://www.someserver.com/myns"
    xmlns:myns = "http://www.someserver.com/myns"
    xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xsi = http://www.w3.org/2001/XMLSchema-instance
  xsi:schemaLocation = "http://www.someserver.com/myns

  http://www.someserver.com/wfs.cgi? request = DescribeFeatureType&amp; typenames =
TreesA_1M,RoadL_1M">
```

该示例中，服务使用 DescribeFeatureType 操作（见第 9 章）访问自身来引用声明了 TreesA_1M 和 RoadL_1M 的模式文档。

7.9 查询表达式

7.9.1 概述

查询表达式用来指导服务器在其数据库中查询满足过滤器表达式编码的资源。查询表达式可以查询单一类型的资源，也可以连接两种或两种以上的资源类型并查询资源元组。

在本标准中，可查询的资源是要素（见 7.1 和 ISO 19143:2010 中的 6.3.3.1.1）。

一个查询表达式处理的是满足其过滤器表达式的一组要素或要素元组，结果集由本标准定义的一些操作进行处理。

示例：GetFeature 操作（见第 11 章）使用查询表达式来指定响应文档中传送给客户端的要素子集。同样，LockFeature 操作（见第 12 章）使用查询表达式来指定要锁定的要素子集。

本标准定义了查询表达式的两种类型：即席查询表达式（ad hoc query expression）和存储的查询表达式（stored query expression）。

即席查询表达式在 XML 中编码为 wfs:Query 元素，因为这种表达式在服务器中没有存储，并且只用在运行的时候才知道，所以它们称为“ad hoc”（特定）。

存储的查询表达式在 XML 中编码为 wfs:StoredQuery 元素，存储的查询表达式事先存储在服务器的数据库中，可以随时使用表达式中的过滤器实现。

这两种查询表达式都继承于在 ISO 19143:2010 第 6 章中定义的抽象查询表达式元素。

7.9.2 即席查询表达式

7.9.2.1 请求语义

即席查询表达式一般用在 GetPropertyValue 操作（见第 10 章）、GetFeature 操作（见第 11 章）、GetFeatureWithLock 操作（见第 13 章）或 LockFeature 操作（见第 12 章）中，用来查询要进行操作的要素集。

如图 8 中所示，一个即席查询表达式包括 typeNames 参数（类型名称）、映射子句、选择子句和排序子句。

必选参数 typeNames 列出要查询的一个或多个要素的类型名称。

可选的映射子句指明了应出现在结果集中的可选要素特性的子集。XML 编码的即席查询表达式中的映射子句也可以用来基于每个特性级别控制对响应文档中的资源引用的解析（见 7.6.4）。

注：对于 KVP 编码的请求，只能基于操作级别控制对响应文档中的资源引用的解析，是利用 RESOLVE、RESOLVEDEPTH 和 RESOLVETIMEOUT 关键字（见 7.6.4.3）来实现的，这与 XML 编码的请求在操作级别和特性级别上控制资源引用的解析是有区别的。

可选的选择子句指定了从服务器的数据库中选择要素的条件。

可选的排序子句指定了响应文档中要素的排序方式。

当在响应文档中表达要素的几何形状时，即席查询表达式同时也允许声明和使用 CRS。

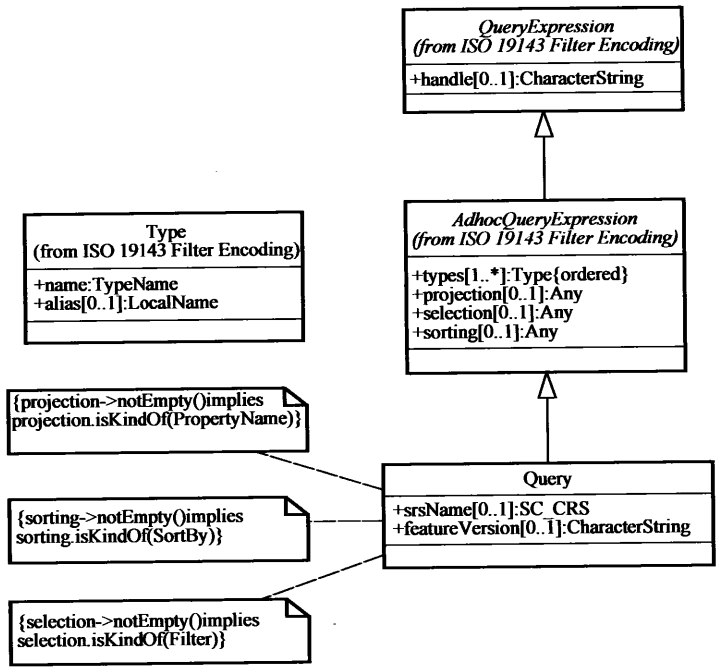

图 8　即席查询表达式

7.9.2.2　XML 编码

下面的 XML 模式片段定义了即席查询表达式的 XML 编码方式。

```
<xsd:element name = "Query" type = "wfs:QueryType"
                substitutionGroup = "fes:AbstractAdhocQueryExpression"/>
<xsd:complexType name = "QueryType">
  <xsd:complexContent>
    <xsd:extension base = "fes:AbstractAdhocQueryExpressionType">
      <xsd:attribute name = "srsName" type = "xsd:anyURI"/>
      <xsd:attribute name = "featureVersion" type = "xsd:string"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

wfs:Query 元素包含了继承于 fes:AbstractAdhocQueryExpressionType(抽象即席查询表达式类型)类型(见 ISO 19143:2010，6.3.2)的 typeNames 参数。

7.9.2.3　KVP 编码

表 8 定义了即席查询表达式的 KVP 编码方式。

该表包括取自 7.9.2.4 和 ISO 19143:2010 表 2 的参数，这些参数是对即席查询表达式进行 KVP 编码的必选参数。

表 8 即席查询 KVP 编码的关键字

URL 关键字	O/M[a]	描述
TYPENAMES(类型名称)	M[b]	见 7.9.2.4.1
ALIASES(别名)	O	见 7.9.2.4.3
SRSNAME(空间参照系名称)	O	见 7.9.2.4.4
Projection clause(映射子句)	O	见表 9
FILTER(过滤器)	O	见 ISO 19143:2010, 6.3.3
FILTER_LANGUAGE(过滤器_语言)	O	见 ISO 19143:2010, 6.3.3
RESOURCEID(资源标识符)	O	见 ISO 19143:2010, 6.3.3
BBOX(外接矩形)	O	见 OGC 06-121r3
SORTBY(排序)	O	见 ISO 19143:2010,第八章 SORTBY 参数用来指定一组特性名称,其特性值用来对满足此请求的要素实例集进行排序。SORTBY 参数值包含格式"PropertyName [ASC|DESC] [,PropertyNam [AASC|DESC], …]",其中字符 ASC 指升序排序,字符 DESC 指降序排序。如果 ASC 和 DESC 都没有指定,那么默认值为升序排序。例如:"SORTBY=Field1 DESC,Field2 DESC,Field3",此例中的结果以 Field1 降序排列,Field2 降序排列,Field3 升序排列

[a] O=Optional(可选);M=Mandatory(必选)。

[b] TYPENAMES 参数在所有实例中是必选的,除了指定 RESOURCEID 参数的情况(见 7.9.2.4.1)。

如果 WFS 请求中出现多个 KVP 编码的查询表达式,那么每个查询表达式的参数需要用圆括号相互间隔(见 6.2.5.3)。如果一个查询表达式实现了一个可选参数,那么请求中的所有查询表达式都要实现这个值。在由 KVP 编码的包含多个查询表达式的请求中,对于互斥的参数,只需实现一个就可以;对于实现的这个参数,没有编码的查询表达式都要实现它的值。

示例:如果 KVP 编码请求包含多个查询表达式,其中一个表达式使用 FILTER 关键词来编码谓词,则请求中的所有查询表达式都应使用 FILTER 关键词。例如在一个有多个查询表达式的 KVP 编码请求中,不能够在其中一个表达式中使用 FILTER 关键词来编码谓词,而另一个表达式使用 RESOURCEID 关键词。

如果 BBOX 关键词在一个请求中使用,则只能在单一的限定范围内编码(见 OGC 06-121r3:2009, 10.2.3)。但是,限定范围内的谓词应适用于请求中的所有查询表达式(见附录 B.8.5.4)。与之不同的是,FILTER 和 RESOURCEID 关键词可以编码多个谓词,相当于在 KVP 编码请求中编码每一个查询表达式(见附录 B.8.4.15)。

正如在 XML 编码的请求中一样,WFS 请求中多个 KVP 编码的查询表达式是相互独立的。

7.9.2.4 参数讨论

7.9.2.4.1 typeNames 参数

在 XML 编码的即席查询表达式中,typeNames 参数使用 wfs:Query 元素中的 typeNames 属性进行编码。该属性继承抽象元素 fes:AbstractAdhocQueryExpressionType(见 ISO 19143,6.3.2)。

在 KVP 编码的即席查询表达式中,TYPENAMES 参数使用 TYPENAMES 关键字进行编码(见表 8)。除非指定 RESOURCEID 参数,否则任意情形下,TYPENAMES 参数都是必选的。当指定 RE-

SOURCEID 参数时，TYPENAMES 参数会被忽略，原因在于每个要素实例都可通过自身的资源标识符独立地被识别(见 7.2)。如果 TYPENAMES 参数和 RESOURCEID 参数都被指定，则由 RESOURCEID 参数标识的所有要素实例类型都由 TYPENAMES 参数指定，否则服务器将抛出一个 InvalidParameterValue 异常(见 OGC 06-121r3：2009，表 25)，该异常中的"locator"属性(见 OGC 06-121r3：2009，8.4)被设置为"RESOURCEID"。

必选参数 typeNames(见 ISO 19143:2010，6.3.3.1.1)在即席查询表达式中被用来对一个或多个关联要素的名称进行编码。单个要素类型名称应编码为 QName(见 W3C XML Schema 第 2 部分)。

每个由 typeNames 参数值列出的 QName 值应与服务能力描述文档(见 8.3)中的要素类型名称相匹配。

7.9.2.4.2 schema-element()函数

如果 typeNames 参数值列表包含一个单独的名称，那么 schema-element()函数可以生成基于指定要素类型和由指定要素类型构成的任意要素类型的一系列查询。

示例："typeNames="schema-element(ns1:Vehicle)"这一个查询语句可能不仅会查询 ns1:Vehicle 要素类型，还会查询到 ns1:Cars，ns1:Boats 等要素类型。

7.9.2.4.3 aliases 参数

在 XML 编码的即席查询表达式中，aliases 参数采用 wfs:Query 元素中的 aliases 属性进行编码。该属性继承抽象元素：fes:AbstractAdhocQueryExpressionType(见 ISO 19143，6.3.2)。

在 KVP 编码的即席查询表达式中，aliases 参数采用 ALIASES 关键字进行编码。

可选参数 aliases 在查询表达式中可用来指定由 typeNames 参数值指定的要素类型名称的别名列表。单个要素类型的别名可在任何地方使用；要素类型名称则可于查询表达式的上下文中。

aliases 参数值中的元素数量应与 typeNames 参数值中对应的要素类型名称的数量相匹配，为 1:1。

示例 1：TYPENAMES=(ns1:FeatureType1，ns2:FeatureType2)(ns2:FeatureType2，ns2:FeatureType3)&ALIASES=(A,B)(C,D)

这个 KVP 编码的示例包含两个都能执行连接查询的即席查询表达式。第一个表达式连接 ns1:FeatureType1 和 ns2:FeatureType2 别名为 A 和 B。第二个表达式连接 ns2:FeatureType2 和 ns2:FeatureType3 别名为 C 和 D。

在单个查询表达式的上下文中，aliases 参数值中的每个别名都应被指定并且唯一。

如果使用了 aliases 参数，那么每个要素类型名都应指定别名，并将别名作为 typeNames 参数的值。

别名主要用于实现连接操作(见 7.9.2.5.3.1)的查询表达式中，以支持自连接。自连接是指一个要素类型和自身相连接。

示例 2：typeNames="myns:Feat1 myns:Feat1" aliases="a b"

这个 XML 编码的示例给出了 typeNames 参数的编码，第一个要素类型是 myns:Feat1，别名为"a"，第二个要素类型是 myns:Feat1，别名为"b"。在请求中可使用"/a/property_name"来引用 myns:Feat1 的第一个实例属性，用"/b/property_name"来引用 myns:Feat1 的第二个实例属性。其中 property_name 代表要素 myns:Feat1 的任意一个属性名称。

7.9.2.4.4 srsName 参数

可选属性 srsName 可以用来表示一个具体的 WFS 支持的对响应文档中返回的要素几何形状的 CRS 转换方式。

srsName 参数值可以设置为服务能力描述文档(见 8.3)中要素类型的 wfs:DefaultCRS 值或任何一个 wfs:OtheCRS 值。如果没有设置 srsName 参数的值，那么在响应文档中使用 wfs:DefaultCRS 值编码要素的几何形状。

对于没有空间特性的要素类型，srsName 属性没有任何意义，并将被忽略。

服务能力描述文档（见 8.3）中声明的多个 wfs:OtherCRS 值可在用于存储要素的 CRS 和使用 srsName 属性请求的 CRS 之间进行转换。

遵循本标准的服务器能够处理使用下面格式的 srsName 属性值。

urn:ogc:def: objectType:authority:version: < EPSG code > [（见 OGC 07-092r3）]

在以上格式中，objectType 存储了"crs" 的值，authority 是指"crs"值，< EPSG code >是指实际的 EPSG code 值的占位符。

示例：srsName="urn:ogc:def:crs:EPSG::26986"

7.9.2.4.5 映射子句

7.9.2.4.5.1 请求语义

WFS 生成的每个要素应包含模式描述（见第 9 章）中所有要素类型的必选特性，也可以包含模式描述中的其他特性。

映射子句列举了一个要素的哪些非必选特性需要包含在一个请求响应中，见图 9。

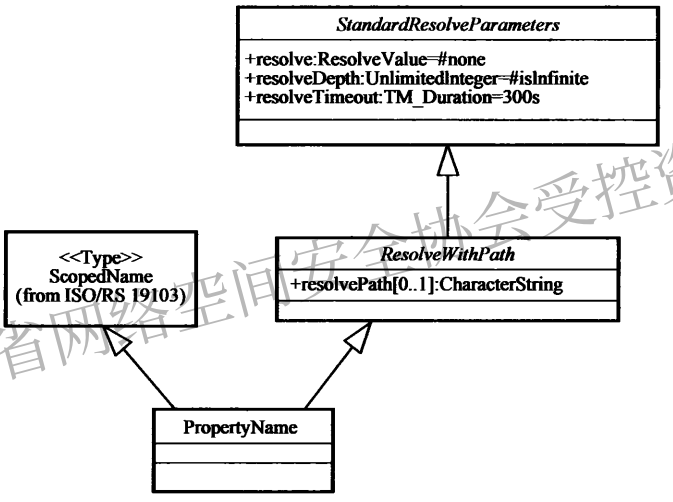

图 9 查询映射子句

注：在描述要素类型时，描述结构具有一定的灵活性，特别考虑了每个特性的可选性或必选性。用来定义 GML 应用模式的 W3C XML 模式（见 W3C XML Schema 第 1 部分）语言使用特定标识来说明特性元素是必选还是可选或者使用多少个特性才算有效。因此，描述一个要素的 GML，可能只包含一部分在模式上有效的特性。WFS 客户端宜能够处理使用映射子句，获取的特性值比请求还要多的情况。

7.9.2.4.5.2 XML 编码

下面的 XML 模式片段定义了 wfs:PropertyName 元素的 XML 编码方式。

```
< xsd:element name = "PropertyName"
                    substitutionGroup = "fes:AbstractProjectionClause">
   < xsd:complexType >
     < xsd:simpleContent >
       < xsd:extension base = "xsd:QName">
         < xsd:attributeGroup ref = "wfs:StandardResolveParameters"/>
         < xsd:attribute name = "resolvePath" type = "xsd:string"/>
```

```
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
```

7.9.2.4.5.3 KVP 编码

表 9 定义了映射子句的 KVP 编码方式。

表 9 映射子句的 KVP 编码

URL 关键字	O/M[a]	描述
PROPERTYNAME(特性名称)	O	该响应包含一组可选特性。 如果多个要素类型指定为 TYPENAMES 关键字(在非连接查询中)的值,那么相应的一组参数 lists 被指定(见 6.2.5.3)。每个子列应与 TYPENAMES 参数中列举的要素类型名称一一对应
StandardResolveParameters (标准解析参数)		见表 6
[a] O=Optional(可选);M=Mandatory(必选)。		

7.9.2.4.6 参数讨论

7.9.2.4.6.1 PropertyName 参数

对于 XML 编码的请求,映射子句使用一个或者多个 wfs:PropertyName 元素进行编码。每个 wfs:PropertyName 元素的值是一个 QName,此 QName 的值和一个要素类型的一个特性名称相对应,这个要素类型为对应要素的 GML 描述的父级元素 wfs:Query 中 typeName 属性的一个要素类型。

对于 KVP 编码的请求,映射子句使用 PROPERTYNAME 关键字进行编码(见表 9)。PROPERTYNAME 关键字的值为一列或多列 QName,此 QName 的值对应 TYPENAME 关键字值中的一个要素类型的一个特性名称。多列特性名称对应多重查询表达式,它们之间使用圆括号(见 6.2.5.3)隔开。

7.9.2.4.6.2 Standard resolve 参数

在 XML 编码的请求中,标准解析参数编码为 wfs:PropertyName 元素的 resolve,resolveDepth 和 resolveTimeout 属性(见 7.6.4.2)。对于 XML 编码的请求,映射子句中的标准解析参数针对每个特性控制如何在响应文档中对资源引用进行解析。wfs:Property 元素中指定的这些参数的值应替换其任何父级元素中相应的参数值。

在 KVP 编码的请求中,标准解析参数使用 RESOLVE, RESLVEDEPTH, 和 RESOLVETIMEOUT 关键字进行编码(见 7.6.4.3)。对于 KVP 编码的请求,如何对响应文档中资源的引用进行解析不是由每个特性分别控制,也不等同于在这个级别上标准解析参数的 KVP 编码,而是由可能用于 KVP 编码请求中的 RESOLVE,RESLVEDEPTH,和 RESOLVETIMEOUT 关键字来控制响应文档中所有要素特性的资源引用的解析。

7.9.2.4.7 resolvePath 参数

在 XML 编码的请求中,参数 resolvePath 使用名为 resolvePath 的属性进行编码。

KVP 编码的请求中没有 resolvePath 参数。

参数 resolvePath 更改 resolve 参数的行为方式。当参数 resolve 值的设为 local(本地),remote(远程)或 all(全部)时,该参数常规的行为方式是依据 resolveDepth 参数指定的深度来解析所有资源的引用。

但参数 resolvePath 只会沿着一条特定的特性路径激发响应文档中的资源引用。

只有当最近的 resolve 参数(在范围内)的值没有设置为“none”,resolvePath 参数的值才能被使用。如果最近的 resolve 属性设置为“none”,那么 resolvePath 参数的任何值都是无效的。

resolvePath 参数的值是一个路径表达式,但它具体的编码取决于要素的编码方式。

对于本标准规范的要素编码语言 GML,参数 resolutionPath 的值为 XPath(见 W3C XML Path Language)表达式,其结果是一个对象元素。在这些示例中,解析会停止,并且参数 resolveDepth 的值被忽略。

示例 1:下面是 resolvePath 参数用来查询要素类型 Parcel 的样例。

要素示例样例:

```
<Parcel gml:id="DEXXXX00000000">
    <registerEntry>
      <LandRegisterEntry gml:id="DEXXXX00000001">
        <relatedTo xlink:href="#DEXXXX00000002"/>
      <LandRegisterEntry>
    </registerEntry>
</Parcel>
<LandRegisterEntry gml:id="DEXXXX00000002">
    <sequenceNumber>1</sequenceNumber>
</LandRegisterEntry>
```

查询样例:

```
<wfs:Query typeNames="Parcel">
    <wfs:PropertyName resolve="all"
    resolvePath="valueOf(relatedTo)">valueOf(registerEntry)</wfs:PropertyName>
    <fes:Filter>
      <! – some filter expression –>
    </fes:Filter>
</wfs:Query>
```

使用 resolvePath 参数的结果为:除了符合条件的所有 Parcel 类型的要素,相关的 LandRegisterEntry 类型的要素和通过特性“relatedTo”而与之相关的 LandRegisterEntry 类型的要素,将一同返回给客户端。

如果在 Xpath 表达式中服务器支持 schema-element()函数,那么在 resolvePath 参数中也支持这个函数。如果使用 schema-element()函数,那么它不仅需要匹配 schema-element()函数参数的元素名称,同时还要匹配直接或间接包含在替代集中的所有元素。

示例 2: schema-element(gml:AbstractFeature)将会匹配所有要素。

7.9.2.5 选择子句

7.9.2.5.1 XML 编码

对于 XML 编码的请求,选择子句编码为 fes:Filter 元素(见 ISO 19143:2010,第 7 章)。

7.9.2.5.2 KVP 编码

对于 KVP 编码的请求,选择子句编码为 FILTER,RESOURCEID 或 BBOX 关键字(见 ISO 19143:2010,表 2)。

7.9.2.5.3 连接操作

7.9.2.5.3.1 连接查询

WFS 可选择性地支持连接查询。

连接查询会在一列要素类型中检索出符合过滤表达式(见 ISO 19143:2010,第 7 章)连接条件的要素元组。如果满足连接条件,那么这些检索出来的要素元组就会在查询表达式的结果集中。

一个连接查询编码方式如下:

1) 使用 typeNames 参数(见 7.9.2.4.1)列出要进行连接查询的要素类型。

2) 使用一个过滤器表达式(见 ISO 19143:2010,第 7 章)声明对要素类型的连接条件。

实现连接查询的服务器要实现一个内部连接,其含义为只有满足连接条件的要素元组才能出现在结果集中。

示例 1:下面的查询表达式使用一个连接查询来检索标识符为"12345"的某人的配偶。

```
<wfs:Query typeNames = "myns:Person myns:Person" aliases = "a b">
  <fes:Filter>
    <fes:And>
      <fes:PropertyIsEqualTo>
        <fes:ValueReference>a/Identifier</fes:ValueReference>
        <fes:Literal>12345</fes:Literal>
      </fes:PropertyIsEqualTo>
      <fes:PropertyIsEqualTo>
        <fes:ValueReference>a/spouse</fes:ValueReference>
        <fes:ValueReference>b/Identifier</fes:ValueReference>
      </fes:PropertyIsEqualTo>
    </fes:And>
  </fes:Filter>
</wfs:Query>
```

在这个示例中,连接谓词"a/spouse"和"b/Identifier"用来定位标识符为"12345"的某人的配偶。因为 myns:Person 要素类型和它自己连接来标识这个配偶,所以这也是一个子关联查询的示例。

示例 2:下面查询表达式使用一个空间连接来查询所有包含湖泊的公园。

```
<wfs:Query typeNames = "myns:Parks myns:Lakes">
    <fes:Filter>
      <fes:Contains>
        <fes:ValueReference>/ns1:Parks/geometry</fes:PropertyName>
        <fes:ValueReference>/ns1:Lakes/geometry</fes:PropertyName>
      </fes:Contains>
</fes:Filter>
</wfs:Query>
```

在 wfs:Query 元素中使用 typeNames 属性(如 typeName="myns:Parks myns:Lakes")来表示要进行连接查询的要素类型。连接谓词通过 fes:Filter 元素来声明,并检索形状满足空间关系 fes:Contains(包含)的 ns1:Park 和 ns1:Lake 要素对。使用连接查询的请求结果在 11.3.3.6 中有所描述。

根据在连接谓词中使用的操作,连接查询可以分为标准,空间,时间三类。

如果除空间和时间操作外其他所有的过滤操作都可以用在连接谓词中,那么这个服务器实现的是标准连接查询,如上面第一个示例。

如果连接谓词中使用了空间操作，那么这个服务实现的是空间连接查询，如上面第二个示例。

如果连接谓词中使用了时间操作，那么这个服务实现的是时间连接查询。

如果 WFS 支持连接查询，那么它应在能力描述文档（见 8.3）中声明 ImplementsStandardJoins（实现标准连接），ImplementsSpatialJoins（实现空间连接）和 ImplementsTemporalJoins（实现时间连接）三个约束。

7.9.2.5.3.2 共享特性

在 GML（见 GB/T 23708—2009）中允许连接查询包含一个映射子句，此映射子句实现了两个或多个连接要素类型的共同特性名称。在这种连接查询的响应中，共同特性将出现在包含此共同特性的响应元组的所有要素中。

示例：假设两个要素类型 ns1:A 和 ns2:B 共享一个特性元素 ns3:C。如果检索 ns1:A 和 ns2:B 的特定连接查询的映射子句实现了共同特性 ns3:C，那么一个服务器如何知道客户端指的是来自要素类型 ns1:A 的特性 ns3:C 还是要素类型 ns2:B 的特性 ns3:C？答案是服务器不知道而且不需要知道，原因是特性 ns3:C 将同时出现在在响应文档中的要素类型 ns1:A 和 ns2:B 中。

7.9.2.5.3.3 连接查询中 schema-elment()函数的使用

如果实现一个连接查询，那么不能使用 schema-element()函数。

如果一个连接查询中使用了 schema-element()函数，那么 WFS 要抛出一个 OperationNotSupported 异常（见表 3），在这个异常中，“locator”属性（见 OGC 06-121r3:2009，8.4）值为“schema-element()”。

7.9.2.5.3.4 坐标参照系的处理

当在一个查询中比较两个具有不同 srsName 属性值的几何对象时，服务器在比较之前，要么将一个几何对象的坐标参照系转换为另一个几何对象的坐标参照系，要么将两个几何对象的坐标参照系转换为第三方共同的坐标参照系。

如果在一个查询中一个或两个几何对象都没有设置相应的 CRS，那么服务器假设这个几何对象使用服务能力描述文档中声明的要素类型的默认 CRS。对两个几何对象的比较按照上面描述的方式进行。

7.9.2.5.3.5 量测单位的处理

本标准没有声明支持任何量测单位之间的转换。

7.9.2.5.4 排序子句

7.9.2.5.4.1 请求语义

即席查询表达式的排序子句通过使用 SortBy 值（见图 10 和 ISO 19143:2010，第 8 章）来实现。

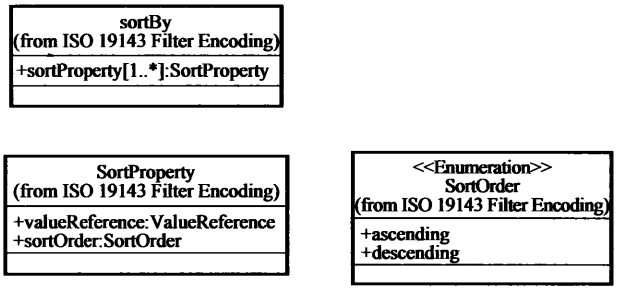

图 10　查询排序子句

7.9.2.5.4.2　XML 编码

在 XML 中，即席查询表达式的排序子句编码为 fes:SortBy 元素(见 ISO 19143:2010，第 8 章)。

即席查询表达式中，fes:SortBy 子元素最大值为 1，最小值为 0。

7.9.2.5.4.3　KVP 编码

在 KVP 编码的请求中，排序子句编码为关键字 SORTBY(见 ISO 19143:2010，表 2)。

7.9.2.5.4.4　排序处理

WFS 接受到一个没有排序子句的即席查询表达式，将会生成一个响应文档，此响应文档中要素按照服务器选择的顺序来进行排序。但是为了遵循本标准，服务器应保证当第一次实现不包含排序子句的即席查询表达式时，使用的任何排序方式将在后续对同样的要素集进行同样的即席查询表达式时继续使用。

示例：如果客户端没有实现排序子句，一个服务器可能会选择要素的 gml:id 来对要素进行排序。后面对同样的数据集实现同样的查询表达式得到的结果在响应文档中宜还是按照同样的顺序对要素进行排序。

当 WFS 接受到包含即席查询表达式的请求且其中带有排序子句时，服务器在响应文档中按照请求的顺序对要素进行排序。

只有在单个即席查询表达式中支持排序。请求中多个即席查询表达式生成的结果集中，不支持对所有的要素进行全局排序。当然这个结论的一个例外就是请求包含单个查询表达式的情况。

当处理一个带有排序子句的请求时，在通过 count 属性对结果集进行缩减之前，需要对满足请求参数要求的所有数据进行排序。

7.9.3　存储的查询表达式

7.9.3.1　请求语义

存储的查询表达式可用在操作 GetPropertyValue(见第 10 章)、GetFeature(见第 11 章)、GetFeatureWithLock(见第 13 章)或 LockFeature(见第 12 章)中，用来检索需要操作的要素集。

存储的查询表达式(见图 11)是持续的、参数化的、可识别的查询表达式。存储的查询表达式通过使用它的标识符能够重复调用，每一次它的参数都可设置为不同的值。

所有服务器都要具备列举、描述和实现存储查询的能力。另外，服务器要提供根据要素的标识符检索要素的存储表达式。服务器还可以提供其他服务器附加的存储表达式。

第 14 章描述了一组管理存储的查询表达式的方法。

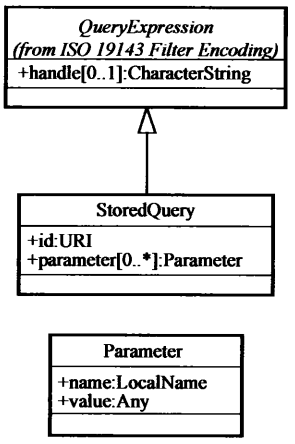

图 11 StoredQuery

7.9.3.2 XML 编码

下面 XML 模式片段定义了存储的查询表达式的 XML 编码。

```
<xsd:element name = "StoredQuery" type = "wfs:StoredQueryType"
                    substitutionGroup = "fes:AbstractQueryExpression"/>
<xsd:complexType name = "StoredQueryType">
    <xsd:complexContent>
      <xsd:extension base = "fes:AbstractQueryExpressionType">
        <xsd:sequence>
          <xsd:element name = "Parameter" type = "wfs:ParameterType"
                                minOccurs = "0" maxOccurs = "unbounded"/>
        </xsd:sequence>
        <xsd:attribute name = "id" type = "xsd:anyURI" use = "required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
<xsd:complexType name = "ParameterType" mixed = "true">
<xsd:sequence>
    <xsd:any namespace = "##other" processContents = "lax" minOccurs = "0"
        maxOccurs = "1"/>
    </xsd:sequence>
    <xsd:attribute name = "name" type = "xsd:string" use = "required"/>
</xsd:complexType>
```

抽象类型 fes:AbstractQueryExpressionType 在 ISO 19143:2010,6.2 中有所描述。

7.9.3.3 KVP 编码

表 10 定义了存储的查询表达式的 KVP 编码。

表 10　存储的查询 KVP 编码的关键字

URL 关键字	O/M[a]	描述
STOREDQUERY_ID	M	调用的存储的查询的标识符
storedquery_parameter=value	O	一个存储的查询的每个参数在 KVP 中编码为一个关键字-值对。 存储的查询参数不能使用与任何 WFS 参数名称相冲突的名称
[a] O=Optional(可选);M=Mandatory(必选)。		

不同于在特殊查询表达式中单个 KVP 编码的请求可以实现多个查询,一般情况每个 KVP 编码的请求只能实现一个存储的查询表达式。

7.9.3.4　存储的查询标识符

在 XML 编码的请求中,存储的查询表达式编码为 wfs:StoredQuery 元素上的 id 属性。

在 KVP 编码的请求中,存储的查询表达式编码为 STOREDQUERY_ID 关键字。

服务器赋予每个存储的查询表达式唯一的标识符,用于查询。

7.9.3.5　存储的查询参数

在 XML 编码的请求中,每个存储的查询参数编码为 wfs:Parameter 元素。参数的名称编码为 wfs:Parameter 元素的 name 属性值。参数的值编码为 wfs:Pamarmeter 元素的内容。

在 KVP 编码的请求中,存储的查询参数编码为关键字-值对。参数名称编码为关键字,参数的值编码为关键字-值对的值。

示例:这个示例是一个 GetFeature 查询,它使用带有"AREA"参数的 GetTreesByArea 存储的查询表达式。

http://www.someserver.com/wfs.cgi? request = GetFeature&storedquery_id = urnx:wfs:StoredQueryId:SomeCompanyName:GetTreesByArea&AREA=10000

服务器要保证存储的查询参数的名称与 WFS 的 KVP 关键字没有冲突。

存储的查询参数是通过名称引用的,所以当编码一个存储的查询表达式时,存储的查询参数能够任意排序。

7.9.3.6　GetFeatureById 存储的查询

所有的服务器可以实现接收单一参数的存储的查询,该参数名为"id",类型为 xsd:string,返回一个单一要素,作为参数 id 的具体值。

该存储的查询有这样一个标识符:

urn:ogc:def:query:OGC-WFS::GetFeatureById

示例:下面 URL 使用必选的 GetFeatureById 存储的查询从服务器数据库里检索和获取单一要素。

```
http://www.someserver.com/wfs.cgi? SERVICE = WFS&
                                   VERSION = 2.0.0&
                                   REQUEST = GetFeature&
                                   STOREDQUERY_ID = urn:ogc:def:query:OGC-WFS::GetFeatureById&
                                            ID = INWATERA_1M.1013
```

8　GetCapabilities 操作

8.1　概述

GetCapabilities 操作返回一个服务器元数据文档,描述服务器提供的 WFS 服务。

所有 WFS 应实现 GetCapabilities 操作的 KVP 编码。

WFS 可以选择性地实现 GetCapabilities 操作的 XML 编码。

8.2 请求

8.2.1 请求语义

图 12 描述了 GetCapabilities 请求的模式。

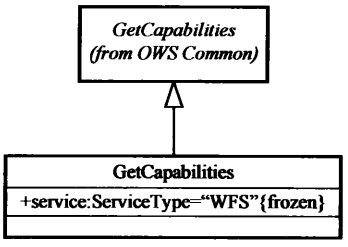

图 12 GetCapabilities 请求

8.2.2 XML 编码

下列 XML 模式片断定义了 GetCapabilities 请求的 XML 编码：

```
<xsd:element name = "GetCapabilities" type = "wfs:GetCapabilitiesType"/>
  <xsd:complexType name = "GetCapabilitiesType">
   <xsd:complexContent>
     <xsd:extension base = "ows:GetCapabilitiesType">
      <xsd:attribute name = "service" type = "ows:ServiceType" use = "required"fixed = "
WFS"/>
     </xsd:extension>
   </xsd:complexContent>
  </xsd:complexType>
```

基本类型 ows:GetCapabilitiesType 在 OWS 通用实现规范中已有定义(参见 OGC 06-121r3:2009，7.2.4)。

8.2.3 KVP 编码

GetCapabilities 请求的 KVP 编码见 OGC 06-121r3:2009 的 7.2.2。

8.3 响应

8.3.1 响应语义

图 13 描述了 GetCapabilities 的响应模式。

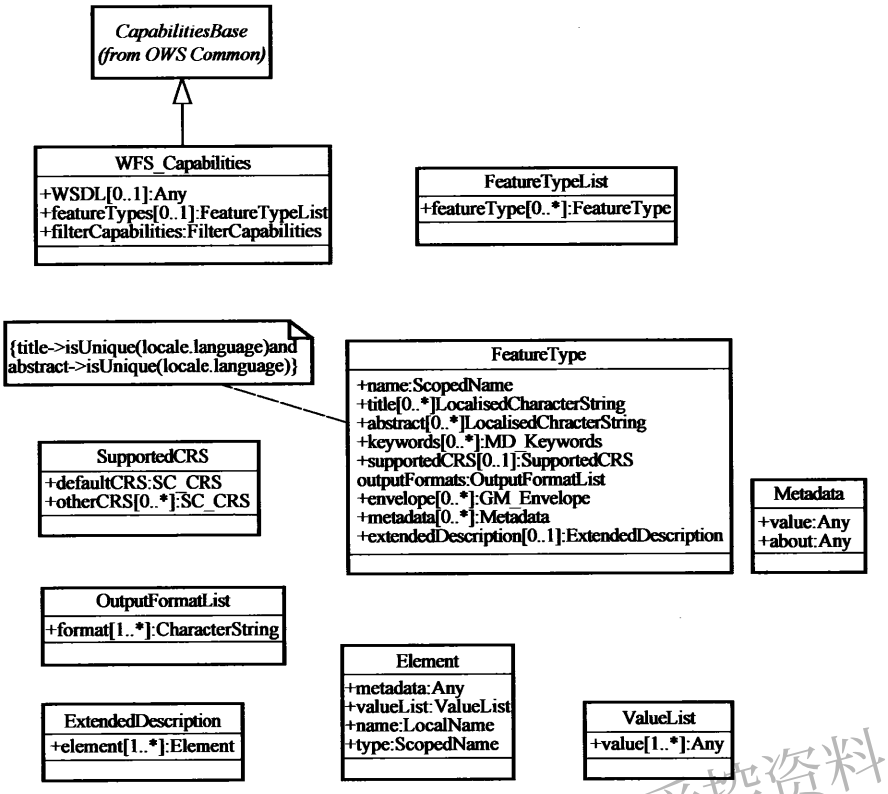

图 13　GetCapabilities 响应

8.3.2　XML 编码

响应 GetCapabilities 请求的根元素是 wfs:WFS_Capabilities 元素，由下列 XML 模式片段所描述：

```
<xsd:element name = "WFS_Capabilites" type = "wfs:WFS_CapabilitiesType"
  <xsd:complexType name = "WFS_CapabilitiesType">
    <xsd:complexContent >
      <xsd:extension base = "ows:CapabilitiesBaseType">
        <xsd:sequence >
          <xsd:element name = "WSDL" minOccurs = "0">
            <xsd:complexType >
              <xsd:complexContent >
                <xsd:restriction base = "xsd:anyType">
                  <xsd:attributeGroup ref = "xlink:simpleLink"/>
                </xsd:restriction >
              </xsd:complexContent >
            </xsd:complexType >
          </xsd:element >
          <xsd:element ref = "wfs:FeatureTypeList" minOccurs = "0"/>
          <xsd:element ref = "fes:Filter_Capabilities" minOccurs = "0"/>
        </xsd:sequence >
      </xsd:extension >
    </xsd:complexContent >
```

```
</xsd:complexType>
```

基本类型 ows:CapabilitiesBaseType 在 OWS 通用实现规范中已有定义(见 OGC 06-121r3:2009,7.2.4)。

元素 ows:ServiceIdentification(服务标识)、ows:ServiceProvider(服务提供者)和 ows:OperationMetadata(操作元数据)继承了基本类型 ows:CapabilitiesBaseType(能力描述文档基本类型)。

8.3.3 服务能力描述文档

除了 OGC 06-121r3:2009 的 7.4 中定义的部分内容外,服务能力响应文档还包括以下部分:

1) WSDL 部分(可选)

这一部分允许服务器引用一个可选的 WSDL 文档以描述服务提供的操作(见附录 E)。这部分内容除了包含在 OperationsMetadata 外,还可以包含在任何需要了解如何使用 WSDL 的工具中。

2) FeatureTypeList(要素类型列表)部分(必选)

这部分定义了 WFS 提供的要素类型的列表,表 11 描述了为每个要素类型定义的概要元数据。

3) Filter(过滤)能力部分(必选)

过滤能力部分的模式定义见 ISO 19143 :2010,7.13,用于形成查询谓词的表达式。

注:在这个模式中,wfs:FeatureTypeList 和 fes:Filter_Capabilities 元素指定为可选元素(如 minOccurs="0"),这是为了支持 GetCapabilities 请求的部分参数(见 OGC 06-121r3:2009,7.3.3),它允许请求概要的服务元数据文档。但是,针对完整的服务元数据就应包含 FeatureType 列表和过滤器能力部分。

8.3.4 FeatureTypeList 部分

wfs:FeatureTypeList 元素应包含一个要素类型列表,是 WFS 提供的 gml:AbstractFeatureType(抽象要素类型)的子类型。

下列 XML 模式片段定义了 wfs:FeatureTypeList 元素:

```
<xsd:element name = "FeatureTypeList" type = "wfs:FeatureTypeListType"/>
  <xsd:complexType name = "FeatureTypeListType">
    <xsd:sequence>
      <xsd:element name = "FeatureType"
                   type = "wfs:FeatureTypeType"
                   maxOccurs = "unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name = "FeatureTypeType">
    <xsd:sequence>
      <xsd:element name = "Name" type = "xsd:QName"/>
      <xsd:element ref = "wfs:Title"    minOccurs = "0" maxOccurs = "unbounded"/>
      <xsd:element ref = "wfs:Abstract" minOccurs = "0" maxOccurs = "unbounded"/>
      <xsd:element ref = "ows:Keywords" minOccurs = "0" maxOccurs = "unbounded"/>
      <xsd:choice>
        <xsd:sequence>
          <xsd:element name = "DefaultCRS"
                       type = "xsd:anyURI"/>
          <xsd:element name = "OtherCRS"
                       type = "xsd:anyURI"
                       minOccurs = "0" maxOccurs = "unbounded"/>
```

```
    </xsd:sequence>
          <xsd:element name = "NoCRS">
           <xsd:complexType/>
          </xsd:element>
        </xsd:choice>
        <xsd:element name = "OutputFormats"
                type = "wfs:OutputFormatListType"minOccurs = "0"/>
        <xsd:element ref = "ows:WGS84BoundingBox"
                minOccurs = "0" maxOccurs = "unbounded"/>
        <xsd:element name = "MetadataURL"
                type = "wfs:MetadataURLType"
                minOccurs = "0" maxOccurs = "unbounded"/>
  <xsd:element name = "ExtendedDescription"
                type = "wfs:ExtendedDescriptionType"
                minOccurs = "0"
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name = "OperationsType">
      <xsd:sequence>
        <xsd:element name = "Operation"
                type = "wfs:OperationType"
                maxOccurs = "unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name = "OutputFormatListType">
      <xsd:sequence maxOccurs = "unbounded">
        <xsd:element name = "Format" type = "xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name = "MetadataURLType">
      <xsd:attributeGroup ref = "xlink:simpleLink"/>
      <xsd:attribute name = "about" type = "xsd:anyURI" />
    </xsd:complexType>
    <xsd:complexType name = "ExtendedDescriptionType">
      <xsd:sequence>
  <xsd:element ref = "wfs:Element" maxOccurs = "unbounded"/>
      <xsd:sequence />
    </xsd:complexType>
  <xsd:element name = "Element" type = "wfs:ElementType"/>
  <xsd:complexType name = "ElementType">
      <xsd:sequence>
  <xsd:element ref = "ows:Metadata"/>
  <xsd:element ref = "wfs:ValueList"/>
      </xsd:sequence>
```

```
<xsd:attribute name = "name" type = "xsd:string" use = "required"/>
<xsd:attribute name = "type" type = "xsd:QName" use = "required"/>
  </xsd:complexType >
  <xsd:element name = "ValueList" type = "wfs:ValueListType"/>
  <xsd:complexType name = "ValueListType">
<xsd:sequence maxOccurs = "unbounded">
<xsd:element ref = "wfs:Value"/>
</xsd:sequence >
  </xsd:complexType >
  <xsd:element name = "Value" type = "xsd:anyType"/>
```

wfs:FeatureTypeList 元素应为服务所提供的每一个要素类型包含一个 wfs:FeatureType 元素，该元素包含了要素类型的元数据。

表 11 列举了用于描述 wfs:FeatureTypeList 中每个要素类型的元素。

表 11 描述要素类型的元素

元素名称	描　述
Name（名称）	由命名空间指定的要素类型的名称，属于必选要素
Title(标题)	无顺序的可阅读的 0 个或多个标题，用于在菜单中简明标识要素类型。xml:lang 属性可用于指定标题的语言。如果有不止一个 wfs:Title 元素，每个标题都应由 xml:lang 属性指定相应的语言类型
Abstract(摘要)	一列无序的 0 个或多个 wfs:Abstract 元素。每一个 wfs:Abstract 元素是对要素类型更多信息的描述。xml:lang 属性可用于为摘要指定语言。如果有不止一个 wfs:Abstract 元素，每个摘要都应由 xml:lang 属性指定相应的语言类型
Keywords(关键字)	wfs:Keywords 元素包含辅助目录搜索的简短单词
DefaultCRS(缺省空间参照系)	如果在一个查询或事务请求中此元素没有明确指定，那么 wf:DefaultCRS 元素则指定 WFS 中用于表达一个空间要素状态的坐标参照系。例如，如果一个 GetFeature 请求没有为 wfs:Query 的 srsName 属性设定 CRS 值，那么满足请求的要素数据的任何空间特性都应使用 wf:DefaultCRS 值表达。CRS 可以使用 OGC 命名空间(参见 OGC 07-092r2)的 URNs 中所定义的 URL 格式(URN) [OGC 05-010]来编码。wf:DefaultCRS 不必是用于要素数据的内部存储 CRS，因此不必像内部存储 SRS 那样被解析。如果 wf:DefaultCRS 不同于内部存储的 CRS，那么 WFS 应支持 wf:DefaultCRS 和内部存储 CRS 之间的转化。当确定和声明要确保数据的精确性时，应考虑转换所造成的影响
OtherCRS(其他空间参照系)	OtherCRS 元素用于指明在查询事务和查询请求中其他被支持的 CRSs。一个“支持的 CRS”意味着 WFS 支持 OtherCRS 和内部存储 CRS 之间的空间特性转换。当确定和声明要确保数据的精确性时，应考虑转换所造成的影响
NoCRS(没有坐标参照系)	NoCRS 元素用于没有空间特性的要素类型，因此无论怎样没有 CRS。对于具有空间特性的要素和要素集此参数不是必备条件。NoCRS 元素从不暗示，因此它不能用于“不知道的 CRS”的语义中。此元素仅仅用于一个可识别的标签，没有元素和属性内容
OutputFormats（输出格式）	OutputFormats 元素是一列 MIME 类型，此类型指明可以给一个要素类型产生输出格式。如果这个可选元素没被指定，那么所有 GetFeature 操作所列出的结果格式应假定被支持

表 11（续）

元素名称	描　述
WGS84BoundingBox（WGS84 外接矩形）	WGS84BoundingBox 元素用于表达一个封闭的矩形边界，此边界是用 WGS84 中的经度和纬度的十进制数表示。目的在于通过表明特殊要素类型存在的位置方便进行地理搜索。由于可以指定多个 ows:WGS84BoundingBox 元素，WFS 可以指出各种数据存在哪里。这个信息使得客户端能够迅速地查询到他们所需要的数据以方便使用
MetadataURL（元数据资源标识地址）	WFS 可以使用 0 个或多个 MetadataURL 元素提供特定要素类型详细的元数据。xlink:href 元素可供任何标准化的元数据进行参考。可选的 about 属性可供包括 MetadataURL 在内的元素的某方面参考，并提供了更多的关于 MetadataURL 元素的信息
ExtendedDescription（扩展的描述）	WFS 使用 wfs：ExtendedDescription 元素可以选择性地把元素加进要素类型的描述中而不必重新定义能力模式。wfs：ExtendedDescription 元素包含一个或多个 wfs：Element 元素。wfs：Element 元素包含一个 name 属性，一个 type 属性和一个值列表，该列表枚举了一个或多个指定的可扩展描述元素。name 属性用于指明扩展的描述性元素的名称。type 属性用于指明扩展的描述性元素值列表中的中值类型。类型应从以 XML 模式[W3C XML Schema 第 2 部分]定义的构建类型列中取出。wfs：Element 元素也包括一个 ows：Metadata 元素，该元素可用于引用描述 wfs：Element 的元数据。wfs:ExtendedDescription 元素可以让使用者用于依据特殊目的对一个要素类型设置用户化描述，或者供提供商把其指定的描述信息加到能力描述文档的要素类型的描述中。 在所有示例中，客户可以安全地忽略全部或一部分扩展的描述性元素。每个添加到要素类型描述中的可扩展的描述性 wfs：Element 都应通过由提供添加元素的描述性元数据的 ows：Metadata 来完成

8.3.5　参数的范围和限制

8.3.5.1　概述

ows:Parameter 和 ows:Constraint 元素在 OWS 通用实现规范（见 OGC 06-121r3:2009，表 13）中已被定义，并且允许为 WFS 提供的所有操作定义全局有效的值域和限制，或者为特定操作定义局部有效的值域和限制。

8.3.5.2　参数值域

表 12 定义了可以在 WFS 能力描述文档值域中定义的参数值域。

表 12　WFS 操作的参数值域

操作名称	参数名称	期望值的类型	描述/可能值
所有操作（除了 GetCapabilities）	version（版本）	字符型	应包括值“2.0.0”，可以包括值“1.1.0”，“1.0.0”或者其他提供商指定的版本号
GetFeature GetFeatureWithLock Transaction	srsName（空间参照系名称）	字符型，URI 或 MIME 类型	WFS 能够处理的 CRS 的列表
GetCapabilities	acceptVersions（可接受的版本）	字符型	应包括值“2.0.0”，可以包括值“1.1.0”，“1.0.0”或者其他提供商指定的版本号

表 12（续）

操作名称	参数名称	期望值的类型	描述/可能值
GetCapabilities	acceptFormats（可接受的格式）	MIME 类型	应包括值"text/xml"，可以包括如"text/html" 或"text/plain"之类的其他 MIME 类型或服务器能够产生的任何其他提供商支持的 MIME 类型
GetCapabilities	Sections（节）	字符型	下面列表中 0 个或多个值："ServiceIdentification"（服务标识），"ServiceProvider"（服务提供者），"OperationsMetadata"（操作元数据），"FeatureTypeList"（要素类型列表），"Filter_Capabilities"（过滤器_能力）
DescribeFeatureType	outputFormat（输出格式）	字符型或 MIME 类型	应包括值"application/gml+xml，version=3.2"，可以包括服务支持的任何其他字符或 MIME 类型，包括 GML 之前的版本
GetPropertyValue	outputFormat（输出格式）	字符型	应包括值"application/gml+xml，version=3.2"。可以包括服务支持的任何其他字符或 MIME 类型，包括 GML 之前的版本
GetPropertyValue	Resolve（解析）	字符型	应包括"none"（无）和"local"（本地），也可以包括"remote"（远程）和"all"（全部），以表明服务器能够解析远程资源引用
GetFeature	outputFormat（输出格式）	字符型或 MIME 类型	应包括值"application/gml+xml，version=3.2"。可以包括服务支持的任何其他字符或 MIME 类型，包括 GML 之前的版本
GetFeature	resolve（解析）	字符型	应包括"none"（无）和"local"（本地），也可以包括"remote"（远程）和"all"（全部），以表明服务器能够解析远程资源引用
GetFeatureWithLock	outputFormat（输出格式）	字符型或 MIME 类型	应包括值"application/gml+xml，version=3.2"。可以包括服务支持的任何其他字符或 MIME 类型，包括 GML 之前的版本
GetFeatureWithLock	Resolve（解析）	字符型	应包括"none"（无）和"local"（本地），也可以包括"remote"（远程）和"all"（全部），以表明服务器能够解析远程资源引用
CreateStoredQuery（创建存储的查询）	Language（语言）	anyURI（任何资源标识符）	应包含 urn-x：wfs：StoredQueryLanguage：WFS_QueryExpression，也可以包含其他支持的语言值。本标准不指定其他可能语言值的语意，也不描述任何额外的值
Transaction	inputFormat（输入格式）	字符型或 MIME 类型	应包括值"application/gml+xml，version=3.2"。可以包括服务支持的任何其他字符或 MIME 类型，包括 GML 之前的版本
Transaction	vendorId（提供商的标识符）	字符型	任何用来作为 wfs：Native 元素指定提供商标识符的字符

不管是局部地针对每个操作，还是全局地针对整个服务，服务器在能力描述文档中声明支持的所有操作都要遵循表 12 的参数值域。

通常，每个 WFS 的实现都有指定的参数值域。

示例 1：srsName 参数的允许值是由 WFS 支持的特定变换来指定的。

然而，在某些情况下一个参数值域是由本标准定义的。

示例 2：在本标准中定义的 resolve 参数（见 7.6.4）值域包含"local"，"remote"，"all"或"none"。

在这些情况下，WFS 可以仅仅限制此值域。如果支持所有值域范围，那么在能力描述文档中就不必列举出参数值域。

8.3.5.3 服务和操作的约束

表 13 列举的所有服务约束应在服务器的能力描述文档（见 8.3.3）中指定，并为所有服务约束参数设置适当的值，以表明服务器是否遵守相应的一致性类。

表 13 服务约束

约束名称	可能的值或/和值类型	描 述
ImplementsBasicWFS （实现基础 WFS）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了基础 WFS 一致性类
ImplementsTransactionalWFS （实现事务 WFS）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了事务 WFS 一致性类
ImplementsLockingWFS （实现锁定 WFS）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了锁定 WFS 一致性类
KVPEncoding （KVP 编码）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了 HTTP GET 一致性类
XMLEncoding （XML 编码）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了 HTTP POST 一致性类
SOAPEncoding （SOAP 编码）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了 SOAP 一致性类
ImplementsInheritance （实现继承）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了 Inheritance 一致性类
ImplementsRemoteResolve （实现远程解析）	布尔值。“TRUE”或“FALSE”	表明此约束指定的操作是否能够解析远程引用。如果此约束没有被指定，那么该值就假定为 FALSE
ImplementsResultPaging （实现结果分页）	布尔值。“TRUE”或“FALSE”	表明服务器是否支持每次要素结果集进行分页的功能
ImplementsStandardJoins （实现标准连接）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了 Standard Joins（标准连接）一致性类
ImplementsSpatialJoins （实现空间连接）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了 Spatial Joins（空间连接）一致性类
ImplementsTemporalJoins （实现时间连接）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了 Temporal Joins（时间连接）一致性类
ImplementsFeatureVersioning （实现要素版本）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了 FeatureVersioning（要素版本）一致性类
ManageStoredQueres （管理存储的查询）	布尔值。“TRUE”或“FALSE”	表明服务器是否实现了 Manage Stored Queries（管理存储的查询）一致性类

服务器可以在能力描述文档中（见 8.3.3）选择性地实现表 14 定义的一个或多个约束。

表 14 操作约束

约束名称	可能的值或(和)值类型	默认值	WFS操作	描述
AutomaticData Locking(自动数据锁定)	布尔值。“TRUE”或“FALSE”	FALSE	Transaction	表明事务操作是否自动锁定数据以维护数据的一致性,这样客户就不必使用LockFeature或GetFeatureWithLock操作锁定将修改的要素
PreservesSibling Order(保持并列顺序)	布尔值。“TRUE”或“FALSE”	FALSE	Transaction	指定服务器对于基数大于1的特性是否保存并列的顺序。如果值为“true”,那么服务器将保持并列的顺序,否则不保存并列的顺序
PagingIsTransac-tionSafe(事务安全分页)	布尔值。“TRUE”或“FALSE”	FALSE	GetFeature GetFeatureWithLock GetPropertyValue	指定服务器是否维持分页码循环之间的事务一致性
CountDefault (计数缺省值)	大于或等于0的整数值		GetFeature GetFeatureWithLock GetPropertyValue	指定count参数的默认值。如果没有指定此约束值,并且响应分页即不被支持也不被请求所触发,那么整个结果应在一个响应中返回。 服务器宜列举CountDefault值作为自我防护的一种方式,以便请求不会影响服务器
ResolveTimeout Default(解析终止缺省值)	大于或等于0的整数值		GetFeature GetFeatureWithLock GetPropertyValue	定义服务器解析资源引用时,在接到响应之前应等待的最大秒数。 如果此元素没有指定,那么服务器在解析远程资源时会无限期地等待
SortLevelLimit (排序级别限定)	大于或等于0的整数值		GetFeature GetFeatureWithLock	该约束定义了可同时排序的特性的最大值。如果对于一个具体服务,一个请求包含太多的fes:SortProperty元素(即超出SortLevelLimit约束),那么服务器将响应一个7.5指定的异常。 如果没有指定该约束,那么对于可能指定的排序特性的数量没有限制
ResolveLocalScope (解析本地资源范围)	大于0的整数值或者“*”字符	*	GetFeature GetFeatureWithLock GetPropertyValue	当解析作为服务器本地数据存储的资源的引用时,定义解析层次的最小和最大值。值“*”表示尽可能多的层次。如果没有指定此约束,那么假定默认值为“*”
ResolveRemoteScope (解析远程资源范围)	大于0的整数值或者“*”字符	*	GetFeature GetFeatureWithLock GetPropertyValue	当解析作为远程资源的引用时,定义解析层次的最小和最大值。值“*”表示尽可能多的层次。如果没有指定此约束,那么假定默认值为“*”

表 14（续）

约束名称	可能的值或（和）值类型	默认值	WFS 操作	描 述
ResponseCache-Timeout（响应缓冲终止时间）	大于 0 的整数		GetFeature GetFeatureWithLock GetPropertyValue	为支持分页（见 7.7.4.4）定义缓冲响应的时间长度，以秒计算。 如果没有指定此约束，那么响应缓存不会终止
QueryExpressions（查询表达式）	QName；wfs:StoredQuery 中的一个 wfs:Query		GetFeature GetFeatureWithLock GetPropertyValue LockFeature	支持的查询表达式元素的名称
注：这些约束可以在对应操作上指定。如果列举了多个操作，那么可对每个操作分别指定约束（可能每个操作有不同的值）或者在服务级别上指定的约束适用于所有列举的操作。				

8.4 扩展点

扩展点允许服务器给现有参数赋予额外的值，在描述要素类型的元数据中动态添加元素，或者执行本标准没有描述的操作。本标准包含下面扩展点：

- DescribeFeatureType/@outputFormat（见 9.2.4.2）
- GetFeature/@outputFormat（见 11.2.4.1）
- fes:AbstractQueryExpression（见 10.2.4.4，11.2.4.3，12.2.4.1）
- GetFeatureWithLock/@outputFormat（见 13.1）
- Transaction/Insert/@inputFormat（见 15.2.4.2）
- Transaction/Update/@inputFormat（见 15.2.5.2.4）
- Transaction/Replace/@inputFormat（见 15.2.6.2.2）
- wfs:ExtendedDescription 元素（见表 11）
- ows:ExtendedCapabilities（见 OGC 06-121r3:2009，7.4.6）
- wfs:Native（见 15.2.8）
- wfs:AbstractTransactionAction（见 15.2.2）

本标准没有赋予在服务能力描述文档中声明的这些扩展点的值任何含义或操作。但是，如果在服务器的能力描述文档中声明了新增的值或操作，那么需要增加与之相应的元数据来描述它们的含义。

元数据可以通过使用 ows:Metadata 元素（见 OGC 06-121r3:2009，表 32）与包含在 wfs:ExtendedDescription 元素中的每个 wfs:Element 元素关联在一起。

OGC 网络服务通用实现规范（见 OGC06-121r3:2009，7.4.6）包含了如何将元数据和能力描述文档中的参数值或操作关联在一起的完整描述。下面的示例说明了这种用法：

示例 1：下面 XML 片段描述了怎样引用新增的 GetFeature 输出格式值的元数据。

```
<Operation name = "GetFeature">
    <DCP>
      <HTTP>
        <Get xlink:href = "http://www.someserver.com/wfs.cgi?"/>
      </HTTP>
    </DCP>
    <Parameter name = "outputFormat">
```

```
        <AllowedValues>
          <Value>application/gml+xml; version=3.2</Value>
          <Value>application/gml+xml; version=3.1</Value>
          <Value>application/gml+xml; version=2.1</Value>
        </AllowedValues>
      </Parameter>
      <Metadata about="application/gml+xml; version=3.1"
          xlink:href="http://portal.opengeospatial.org/files/?artifact_id=4700"/>
      <Metadata about=" application/gml+xml; version=2.1"
          xlink:href="http://portal.opengeospatial.org/files/?artifact_id=11339"/>
  </Operation>
```

示例 2：下面 XML 片段说明，提供商的专用操作如何使用 wfs:Native 元素实现元数据的引用。

```
  <Operation name="Transaction">
      <DCP>
        <HTTP>
          <Get xlink:href="http://www.someotherserver.com/transform?"/>
        </HTTP>
      </DCP>
      <Parameter name="Native">
        <AllowedValues>
          <Value>ALTER SESSION ENABLE PARALLEL DML</Value>
          <Value>MySecretAction</Value>
        </AllowedValues>
      </Parameter>
      <Metadata about="ALTER SESSION ENABLE PARALLEL DML"
        xlink:href="http://downloaduk.
        oracle.com/docs/cd/A97630_01/server.920/a96540/statements_22a.htm#2053493"/>
      <Metadata about="MySecretAction" xlink:href=http://www.yas.com/MySecretActionDescription.html/>
      </Operation>
```

示例 3：下面 XML 片段说明在服务能力描述文档中，如何使用 ows:ExtendedCapabilities 元素声明提供商专用的操作。

```
  <ExtendedCapabilities>
      <Operation name="MySecretOperation">
        <DCP>
          <HTTP>
            <Get xlink:href="http://www.someserver.com/wfs.cgi?"/>
          </HTTP>
        </DCP>
        <Metadata about="MySecretOperation"
  xlink:href="http://www.myserver.com/Operations/MySecretOperation.html"/>
      </Operation>
        </ExtendedCapabilities>
```

8.5 异常

当 WFS 遇到解析 GetCapabilities 请求所产生的错误时，应抛出一个如 7.5 所示的 OperationParsingFailed 异常。

当 WFS 遇到处理 GetCapabilities 请求所产生的错误时，应抛出一个如 7.5 所示的 OperationProcessingFailed 异常。

9 DescribeFeatureType 操作

9.1 概述

必选操作 DescribeFeatureType 返回 WFS 服务实例能够提供的要素类型的模式描述。模式描述定义了 WFS 的要素示例如何在输入(通过插入、更新和替换操作)和输出(GetFeature、GetFeatureWithLock 和 GetGmlObject 的响应)时进行编码。

9.2 请求

9.2.1 请求语义

图 14 描述了 DescribeFeatureType 请求的模式。

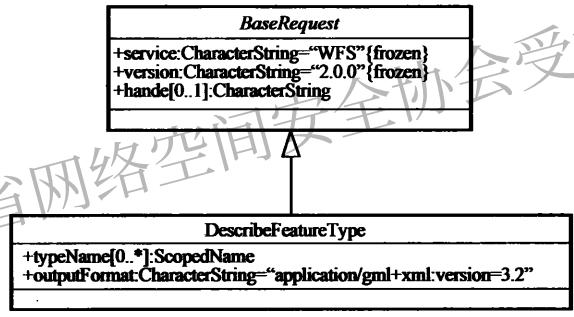

图 14 DescribeFeatureType 请求

9.2.2 XML 编码

下面的 XML schema 片段定义了 DescribeFeatureType 请求的 XML 编码。

```
<xsd:element name = "DescribeFeatureType" type = "wfs:DescribeFeatureTypeType"/>
  <xsd:complexType name = "DescribeFeatureTypeType">
    <xsd:complexContent >
      <xsd:extension base = "wfs:BaseRequestType">
        <xsd:sequence >
          <xsd:element name = "TypeName" type = "xsd:QName"
                       minOccurs = "0" maxOccurs = "unbounded"/>
        </xsd:sequence >
        <xsd:attribute name = "outputFormat" type = "xsd:string" default = " application/
gml + xml; version = 3.2"/>
      </xsd:extension >
```

```
    </xsd:complexContent>
  </xsd:complexType>
```

9.2.3 KVP 编码

表 15 定义了 DescribeFeatureType 请求的 KVP 编码。

表 15 DescribeFeatureType 的 KVP 编码

URL 关键字	O/M[a]	描述
公共关键字 (REQUEST=DescribeFeatureType)		见表 7(只包括所有操作或者 DescribeFeatureType 操作的关键字)
TYPENAME (类型名称)	O	用一列由逗号分开的要素类型进行描述。如果没有指定任何值,那么由服务器提供描述要素类型的完整应用模式
OUTPUTFORMAT (输出格式)	O	应支持"application/gml+xml; version=3.2"值,表明可以产生一个 GML(见 GB/T 23708—2009)应用模式。服务器可能支持本标准未指定意义的其他值

[a] O = Optional(可选的);M = Mandatory(必选的)。

9.2.4 参数讨论

9.2.4.1 typeName(类型名称)参数

在 XML 编码的请求中,typeName 参数编码为 wfs:DescribeFeatureType 元素中的 typeName 属性。

在 KVP 编码的请求中,typeName 参数编码为 TYPENAMES 关键字。

typeName 参数编码 DescribeFeatureType 操作描述的一个或多个要素类型的名称。

typeName 参数的允许值集为服务器能力描述文档中列举的要素类型名称的集合。

如果 typeName 参数缺省,那么响应将返回服务器支持的完整应用模式。

9.2.4.2 outputFormat(输出格式)参数

在 XML 编码的请求中,outputFormat 参数编码为 wfs:DescibeFeatureType 元素的 outputFormat 属性。

在 KVP 编码的请求中,outputFormat 参数编码为 OUTPUTFORMAT 关键字(见 7.6.3.3)。

outputFormat 参数用来表示描述要素类型的模式描述语言。

默认值"application/gml+xml; version=3.2" 表示使用 GML(见 GB/T 23708—2009)应用模式描述要素类型。所有遵循本标准的 WFS 都支持这个默认值。

服务器可以在能力描述文档(见表 12)中声明 outputFormat 属性的其他值,这表明服务器支持多个输出格式,包括 GML 以前的版本。但是本标准只说明了默认值"application/gml+xml; version=3.2"的含义。

如果能力描述文档中列举了任何其他的值,本标准建议在能力描述文档(见 8.3)中添加描述其他值含义的元数据(见 OGC 06-121r3:2009,表 32)。

注:因为本标准以前的 OGC 版本(见 OGC 02-058 和 OGC 04-094)与 GML 具体版本是绑定的,所以服务器可以在能力描述文档中(见表 12)声明支持本标准以前的 OGC 版本,以表明支持 GML 以前的版本。

9.3 响应

9.3.1 概述

在响应一个 DescribeFeatureType 请求时，如果其 outputFormat 属性值已经设为 application/gml+xml; version=3.2，那么 WFS 实现应能提供一个完整并且有效的 GML(参见 GB/T 23708—2009)应用模式，模式中包含请求列举的要素类型的定义。DescribeFeatureType 请求返回的文档可以用来验证 WFS 生成的以要素集合形式表现的要素实例的有效性，或者用来验证作为事务操作输入的要素实例的有效性。

9.3.2 支持多命名空间

一个 XML schema 文档只能描述属于单一命名空间(见 W3C XML Schema 第 2 部分)的元素。当 DescribeFeatureType 操作请求多命名空间的要素类型时，服务器应生成请求的命名空间的完整模式，并且导入余下的命名空间。本标准不强制要求生成和导入命名空间的具体模式。

示例：请看下面描述多命名空间的要素类型的请求：

```
<? xml version = "1.0" ? >
< DescribeFeatureType
  version = "2.0.0"
  service = "WFS"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:ns01 = "http://www.server01.com/ns01"
  xmlns:ns02 = "http://www.server02.com/ns02"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
    http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
  < TypeName > ns01:TreesA_1M </TypeName >
  < TypeName > ns02:RoadL_1M </TypeName >
</DescribeFeatureType >
```

对于这个请求，WFS 可都会生成下面的响应：

```
<? xml version = "1.0" ? >
< xsd:schema
    targetNamespace = "http://www.server01.com/ns01"
    xmlns:ns01 = "http://www.server01.com/ns01"
    xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
    elementFormDefault = "qualified"
    attributeFormDefault = "unqualified">
    < import namespace = "http://www.server02.com/ns02"
        schemaLocation = "http://www.yourserver.com/wfs.cgi?
        REQUEST = DescribeFeatureType&amp;TYPENAMES = ns02:RoadL_1M"/>
    < xsd:element name = "TREESA_1M" type = "ns01:TREESA_1MType" substitutionGroup = "
gml:_Feature"/>
    < xsd:complexType name = "TREESA_1MType">
        < xsd:complexContent >
        < xsd:extension base = "gml:AbstractFeatureType">
            <! — list property declarations for feature type TREESA_1M —>
```

```
        </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <! – other declarations that are part of this schema –>
</xsd:schema>
```

9.4 异常

当 WFS 解析一个 DescribeFeatureType 请求遇到的错误时，将抛出一个如 7.5 中描述的 OperationParsingFailed 异常。

当 WFS 处理一个 DescribeFeatureType 请求遇到的错误时，将抛出一个如 7.5 中描述的 OperationProcessingFailed 异常。

10 GetPropertyValue 操作

10.1 概述

GetPorpertyValue 操作允许从数据存储器中通过查询表达式提取一组指定要素中某个要素的特性值或者一个复杂要素特性值的某个部分(参见 7.9)。

10.2 请求

10.2.1 请求语义

图 15 描述了 GetPropertyValue 的请求模式。

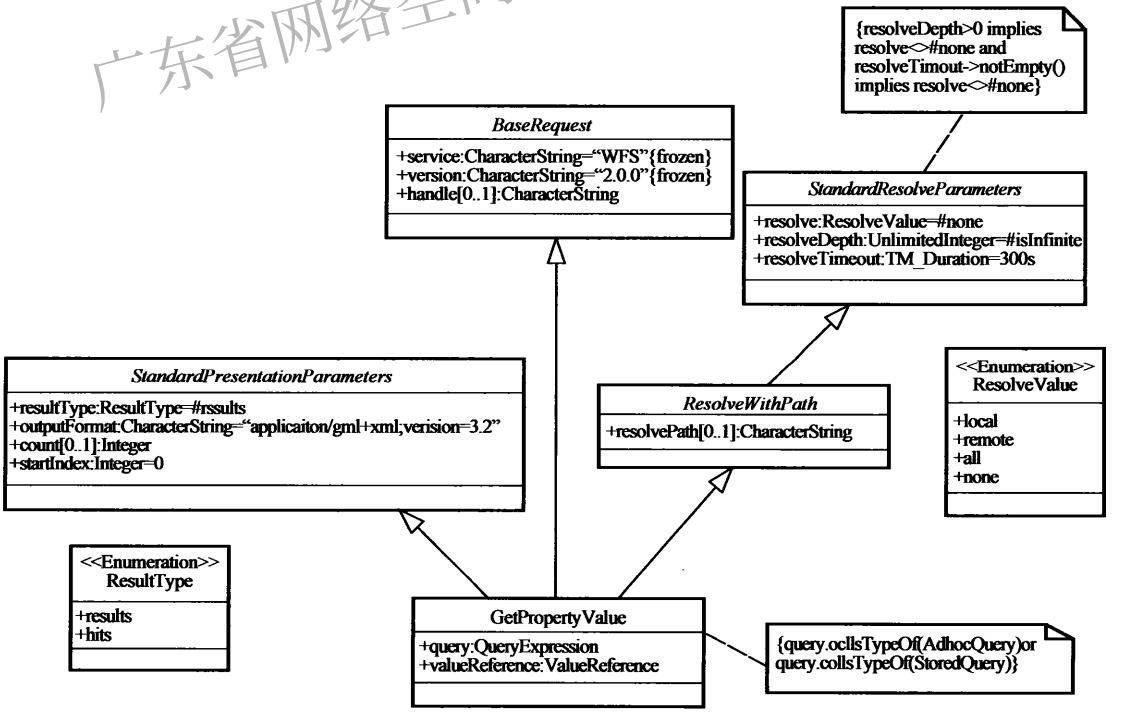

图 15 GetPropertyValue 请求

10.2.2 XML 编码

下列 XML 片段定义了 GetPropertyValue 操作的 XML 编码。

```
<xsd:element name = "GetPropertyValue" type = "wfs:GetPropertyValueType"/>
<xsd:complexType name = "GetPropertyValueType">
  <xsd:complexContent >
    <xsd:extension base = "wfs:BaseRequestType">
      <xsd: sequence >
        <xsd:element ref = "fes:AbstractQueryExpression"/>
      </xsd: sequence >
      <xsd:attribute name = "valueReference" type = "xsd:string" use = "required"/>
      <xsd:attribute name = "resolvePath" type = "xsd:string"/>
      <xsd:attributeGroup ref = "wfs:StandardPresentationParameters"/>
      <xsd:attributeGroup ref = "wfs:StandardResolveParameters"/>
    </xsd:extension >
  </xsd:complexContent >
</xsd:complexType >
```

10.2.3 KVP 编码

定义了 GetPropertyValue 操作的 KVP 编码。

表 16 GetPropertyValue 的 KVP 编码关键字

URL 关键字	O/ M[a](可选/必选)	描　　述
公共关键字 (REQUEST=GetPropertyValue)		见表 7(只包括所有操作或者 GetProperty-Value 操作的关键字)
即席的查询关键字(与存储的查询关键字互斥)		见表 8
存储的查询关键字(与即席的查询关键字互斥)		见表 10
VALUEREFERENCE (值引用)	M	见 10.2.4.3
RESOVEPATH (解析路径)	O	见 7.9.2.4.7
[a] O = Optional(可选的);M = Mandatory(必选的)。		

10.2.4 参数讨论

10.2.4.1 StandardPresentationParameters

见 7.6.3。

10.2.4.2 StandardResovleParameters

见 7.6.4。

10.2.4.3 valueReference (值引用)参数

valueReference(值引用)参数是一个 XPath 表达式(见 ISO 19143:2010,7.4.4),用以指向和引用某个要素特性值的一个节点或者某个属性节点的子节点,该要素特性值应从某个服务器的存储器上获取,

并在响应文档中说明。

该参数的响应应是一个文本节点或者由 valueReference 参数指向的一组元素节点的列表。当该参数值由远程资源提供时，则会用 valueOf()访问函数来获取远程资源上的值。

注：valueOf() XPath 操作(见 7.3.2)的 valueReference 参数与标准 resolve 参数的控制是有区别的。valueReference 是用来表达查询以确定结果集中的要素/值，而标准 resolve 参数(见 7.6.4)是用于确定选择的要素/值中的特性值在响应文档中是如何编码或表现的。

示例：下列 XML 片断描述了要素类型 myns:Person 的一个实例：

```
<myns:Person gml:id = "p4467"
<gml:identifier
    codespace = http://www.canadaSIN.com > 424679360 </gml:identifier >
  <myns:lastName > Smith </myns:lastName >
  <myns:firstName > Mary </myns:firstName >
  <myns:age > 18 </myns:age >
  <myns:sex > Female </myns:sex >
  <myns:spouse xlink:href = "#p4456"/>
  <myns:location xlink:href = "#p101" />
  <myns:mailAddress >
    <myns:Address gml:id = "a201">
      <myns:streetName > Main St.</myns:streetName >
      <myns:streetNumber > 4 </myns:streetNumber >
      <myns:city > SomeCity </myns:city >
      <myns:province > Someprovince </myns:province >
      <myns:postalCode > X1X 1X1 </myns:postalCode >
      <myns:country > Canada </myns:country >
    </myns:Address >
  </myns:mailAddress >
  <myns:phone > 416 - 123 - 4567 </myns:phone >
  <myns:phone > 416 - 890 - 1234 </myns:phone >
  <myns:livesIn xlink:href = "#h32"/>
  <myns:isDriving xlink:href = "r1432"/>
</myns:Person >
```

值的引用 valueReference＝”lastName”用以指定这个人的姓，如本例响应文档中的 Smith。valueReference＝”myns:mailAddress/myns:Address/myns:city”用以指定 Mary Smith 居住城市的名称，如本例中的 SomeCity。valueReference＝”valueOf(myns:location)”通过解析值的引用”#p101”指定 Mary Smiths 所在的位置。

10.2.4.4 AbstractQueryExpression(抽象查询表达式)参数

GetPropertyValue 请求是针对一组通过一个查询表达式查询到的要素进行操作的。

本标准定义了 wfs:Query(见 7.9.2.2) 和 wfs:StoredQuery (见 7.9.3.2)两个元素，用作 GetPropertyValue 操作中的查询表达式，取代 fes:AbstractQueryExpression (见 ISO 19143:2010，6.2)

可以定义其他的元素来替代 fes:AbstractQueryExpression，但是本标准只定义了 wfs:Query 和 wfs:StoredQuery 两个元素的含义。

7.9.2.3 中定义了特定的查询(ad hoc Query)表达式的 KVP 编码。

7.9.3.3 中定义了存储的查询(Stored Query)表达式的 KVP 编码。

KVP 编码的 GetPropertyValue 请求应只包含一种类型的查询表达式，即要么是即席查询表达式，要么是存储的查询表达式。

10.3 响应

10.3.1 响应语义

图 16 描述 GetPropertyValue 的响应语义。

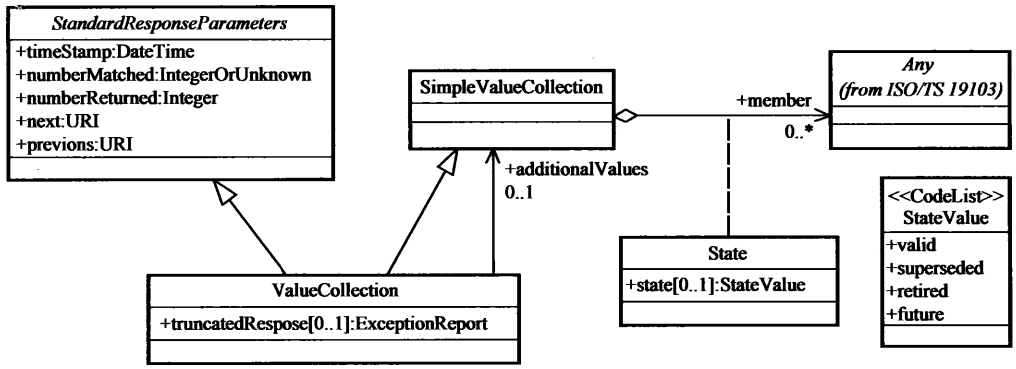

图 16 GetPropertyValue 响应

10.3.2 XML 编码

下列 XML 模式片段定义了 GetPropertyValue 操作的响应。

```
<xsd:element name = "ValueCollection" type = "wfs:ValueCollectionType"/>
  <xsd:complexType name = "ValueCollectionType">
    <xsd:sequence>
<xsd:element ref = "wfs:member" minOccurs = "0" maxOccurs = "unbounded"/>
<xsd:element ref = "wfs:additionalValues" minOccurs = "0"/>
      <xsd:element ref = "wfs:truncatedResponse" minOccurs = "0"/>
    </xsd:sequence>
<xsd:attributeGroup ref = "wfs:StandardResponseParameters"/>
  </xsd:complexType>
<xsd:element name = "member" type = "wfs:MemberPropertyType"/>
<xsd:complexType name = "MemberPropertyType" mixed = "true">
  <xsd: choice minOccurs = "0">
    <xsd:any processContents = "lax" namespace = "##other" minOccurs = "0"/>
<xsd:element ref = "wfs:Tuple"/>
<xsd:element ref = "wfs:SimpleFeatureCollection"/>
  </xsd: choice>
  <xsd:attribute name = "state" type = "wfs:StateValueType"/>
<xsd:attributeGroup ref = "xlink:simpleLink"/>
</xsd:complexType>
<xsd:element name = "Tuple" type = "wfs:TupleType"/>
<xsd:complexType name = "TupleType">
  <xsd:sequence>
```

```
      <xsd:element ref="wfs:member" minOccurs="2" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="additionalValues">
  <xsd:complexType>
  <xsd:choice>
  <xsd:element ref="wfs:ValueCollection"/>
     <xsd:element ref="wfs:SimpleFeatureCollection"/>
  </xsd:choice>
  </xsd:complexType>
  </xsd:element>
  <xsd:element name="truncatedResponse">
  <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ows:ExceptionReport"/>
      </xsd:sequence>
  </xsd:complexType>
  </xsd:element>
  <xsd:simpleType name="StateValueType">
    <xsd:union>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="valid"/>
          <xsd:enumeration value="superseded"/>
          <xsd:enumeration value="retired"/>
          <xsd:enumeration value="future"/>
        </xsd:restriction>
      </xsd:simpleType>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:pattern value="other:\w{2,}"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:union>
  </xsd:simpleType>
```

wfs:ValueCollection 包含零个或多个 wfs:member 要素，该要素包含 GetPropertyValue 操作中通过查询表达式指定的要请求的值。值可能是 wfs:member 要素的嵌入式编码或通过 xlink:href 属性引用的值。xlink:href 属性申明为 wfs:member 要素的 xlink:simpleLink 属性组的一部分。

如果需要在响应文档中包含额外对象来满足引用远程资源的需要(见 7.6.4)，这些额外对象则应用 wfs:AdditionalValues 元素来指定，并且在返回值中的资源引用应重新定位，指向远程引用资源在本地的备份。同 7.6.4.5 中描述的一样，该服务器也应对所有未解析的嵌套资源引用进行适宜的重新定位，指向所准备好的资源。

如果在服务器开始将响应文档传送到客户端过程中遇到异常，该服务器应通过 wfs:TruncatedResponse 元素中断响应文档的传输，该元素中包含的 ows:ExceptionReport 元素将报告异常信息(见

7.5)。如果服务器成功地完成了整个响应文档的传输，wfs:TruncatedResponse 元素则不必包含在响应文档中。

10.3.3 state(状态)参数

不支持要素版本一致性类(见表 1)的服务器不必使用 state 参数。

支持要素版本一致性类的服务器应利用 state 参数在响应文档中报告具有版本变化特征的值的状态。state 参数的取值包括字符串"retired"(过期)、"superseded"(被替代)、"valid"(有效)、"future"(将来)和带有前缀"other:"(其他)的其他字符串。"retired"指该值已经从所查询的数据存储器上删除了；"superseded"表明所查询的数据存储器上特定版本的值已经被较新版本的值替代了；"valid"是指所查询的数据存储器上的当前版本。"future"表明尚未有效的值，合理的原因如它可能用于发布某个指定的值。

任何其他带有"other:"标记的字符串可能用来报告特定服务器的版本状态，本标准没有为带有"other:"标记的值赋予任何意义。

10.3.4 标准响应参数

有关标准响应参数的讨论见 7.7。

10.4 异常

如果 WFS 服务在解析一个 GetPropertyValue 请求时遇到错误，就会抛出如 7.5 描述的 OperationParsingFailed 异常信息。

如果 WFS 服务在处理一个 GetPropertyValue 请求时遇到错误，就会抛出如 7.5 描述的 OperationProcessingFailed 异常信息。

11 GetFeature 操作

11.1 概述

GetFeature 操作返回一个数据请求中所选择的要素。WFS 处理一个 GetFeature 请求并向客户端返回一个响应文档，该响应文档中包含了满足请求中指定查询条件的 0 个或多个要素实例。

采用 GML(见 GB/T 23708—2009)对要素进行规范表达，GetFeature 请求格式是在结果集表达基础上进行建模的。为此，有必要清楚地理解通用要素模型(见 ISO 19109)是如何映射为 GML 表达的。GML 的完整描述见 GB/T 23708—2009，这里只是对重要的部分做了总结。

在 GML 中，一个要素表示为一个 XML 元素。该元素的名称表明了要素的类型(Feature Type)，通常以开头字母大写(UpperCamelCase)给出，例如，*xmml:BoreHole* 或 *myns:SecondaryCollege* 。

该元素的内容由一组描述要素特性值的元素组成，每个特性是该要素元素的一个子元素。特性元素的名称代表特性的含义，通常以开头字母小写(lowerCamelCase)给出，例如，*gml:boundedBy* 或 *xmml:collarLocation*。

特性值嵌入在特性元素的内容中，或者由作为特性元素的 XML 属性的链接所标识的资源值的引用给出。如果使用了嵌入方式，那么内容可能是文字(数字、文本等)，或者通过 XML 元素来构成，但是不能对特性值结构进行假定。在某些情况下，要素的一个特性值可能是另外一个要素，例如，myns:School 可能拥有 myns:frontsOn 特性，而 myns:frontsOn 特性的值是 myns:Road，同时，myns:Road 自身又有其他特性等。

11.2 请求

11.2.1 请求语义

图 17 描述了 GetFeature 请求的模式。

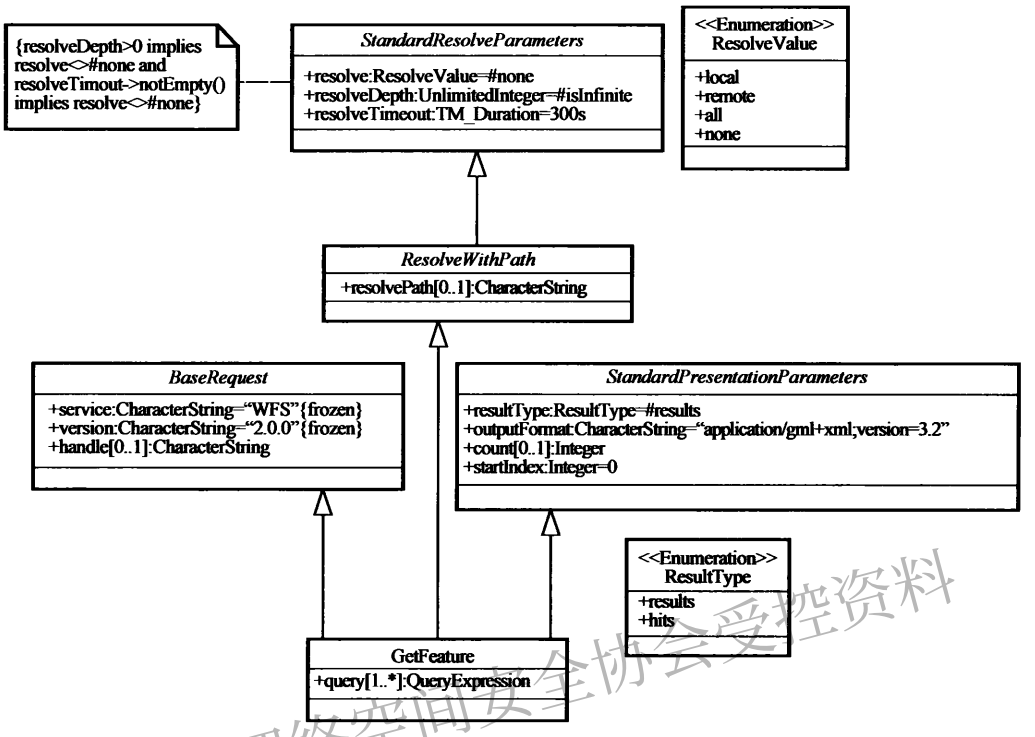

图 17 GetFeature 请求

11.2.2 XML 编码

下面一段 XML 模式片段定义了 XML 中 GetFeature 请求的编码：

```
<xsd:element name = "GetFeature" type = "wfs:GetFeatureType"/>
  <xsd:complexType name = "GetFeatureType">
    <xsd:complexContent>
      <xsd:extension base = "wfs:BaseRequestType">
        <xsd:sequence>
          <xsd:element ref = "fes:AbstractQueryExpression"
                       maxOccurs = "unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref = "wfs:StandardPresentationParameters"/>
        <xsd:attributeGroup ref = "wfs:StandardResolveParameters"/>
      </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

GetFeature 请求可以包含一个或多个查询表达式。一个查询表达式确定一组在响应文档中返回给客户端的要素实例。GetFeature 请求中的查询表达式之间是相互独立的，可以以任意顺序出现，但是结果集按照 GetFeature 请求中查询表达式的顺序来进行排序。

11.2.3 KVP 编码

表 17 定义了 GetFeature 请求的 KVP 编码。

表 17 GeFeature 操作的 KVP 编码关键字

URL 关键字	描述
公共关键字 (REQUEST=GetFeature)	对于可以用在 KVP 编码的 GetFeature 请求的其他参数，见表 7
StandardPresentationParameters(标准表达参数)	见表 5
StandardResolveParameter (标准解析参数)	见表 6
即席查询关键字(与存储的查询关键字相互排斥)	见表 8
存储的查询关键字(与即席查询关键字相互排斥)	见表 10

11.2.4 参数讨论

11.2.4.1 StandardPresentationParameters

见 7.6.3。

11.2.4.2 StandardResovleParameters

见 7.6.4。

11.2.4.3 AbstractQueryExpression(抽象查询表达式)参数

GetFeature 请求在一组可识别的要素集上使用一个或多个查询表达式进行操作。

对于 XML 编码的请求，本标准定义了 wfs:Query (见 7.9.2.2)和 wfs:StoredQuery (见 7.9.3.2)两个元素。这两个元素可以在 GetFeature 请求中用作查询表达式，并且替代 fes:AbstractQueryExpression (见 ISO 19143:2010，6.2)。一个 GetFeature 请求可以由多个 wfs:Query 与(或者)wfs:StoredQuery 元素编码而成。

fes:AbstractQueryExpression 也可以由其他的元素取代，但是本标准只为 wfs:Query 和 wfs:StoredQuery 两个元素定义了具体含义。

即席查询表达式的 KVP 编码在 7.9.2.3 中已有定义，一个 KVP 编码的请求可以由多个即席查询表达式编码而成。

存储的查询表达式的 KVP 编码在 7.9.3.3 中已有定义。一个 KVP 编码的请求中只能包含一个存储的查询表达式。

KVP 编码的请求只包含一种类型的查询表达式，要么为即席查询表达式，要么为存储的查询表达式。

11.3 响应

11.3.1 响应语义

图 18 描述了 GetFeature 响应的模式。

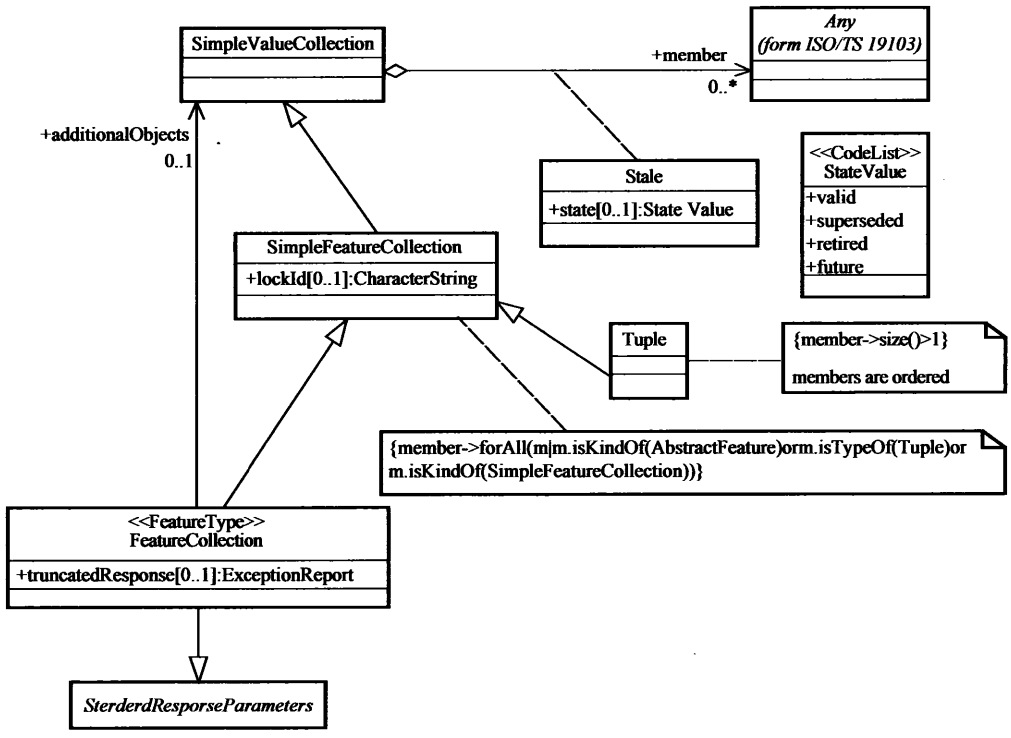

图 18　GetFeature 响应

11.3.2　XML 编码

GetFeature 请求的响应是一个带有根元素< wfs:FeatureCollection >(要素集合) 的 XML 文档,其 XML schema 片段如下:

```
<xsd:element name = "FeatureCollection"
             type = "wfs:FeatureCollectionType"
             substitutionGroup = "wfs:SimpleFeatureCollection"/>
<xsd:complexType name = "FeatureCollectionType">
  <xsd:complexContent >
    <xsd:extension base = "wfs:SimpleFeatureCollectionType">
      <xsd:sequence >
        <xsd:element ref = "wfs:additionalObjects" minOccurs = "0"/>
        <xsd:element ref = "wfs:truncatedResponse" minOccurs = "0"/>
      </xsd:sequence >
      <xsd:attributeGroup ref = "wfs:StandardResponseParameters"/>
      <xsd:attribute name = "lockId" type = "xsd:string"/>
    </xsd:extension >
  </xsd:complexContent >
</xsd:complexType >
<xsd:element name = "additionalObjects">
  <xsd:complexType >
    <xsd:choice >
      <xsd:element ref = "wfs:ValueCollection"/>
```

```
        <xsd:element ref = "wfs:SimpleFeatureCollection"/>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name = "SimpleFeatureCollection"
               type = "wfs:SimpleFeatureCollectionType"/>
  <xsd:complexType name = "SimpleFeatureCollectionType">
    <xsd:sequence>
      <xsd:element ref = "wfs:member" minOccurs = "0" maxOccurs = "unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name = "boundedBy" type = "wfs:EnvelopePropertyType"/>
  <xsd:complexType name = "EnvelopePropertyType">
    <xsd:sequence>
      <xsd:any namespace = "##other"/>
    </xsd:sequence>
  </xsd:complexType>
```

11.3.3 参数讨论

11.3.3.1 lockId(锁标识符)参数

lockId 参数不包含在 GetFeature 操作的响应中。

见 13.3.2 对 lockId 参数的讨论。

11.3.3.2 state(状态)参数

10.3.2 中的 XML 片段声明了包含 state 属性的 wfs:member(成员)要素。

不支持要素版本一致性类(见表 1)的服务器不使用 state 参数。

支持要素版本一致性类的服务器在响应文档中使用 state 参数报告最新版本的要素的状态。state 参数的取值包括字符串“retired”(过期)、“superseded”(被替代)、“valid”(有效)、“future”(将来)和带有前缀“other:”的其他字符串。“retired”指该值已经从所查询的数据存储器上删除了;“superseded”表明所查询的数据存储器上特定版本的值已经被较新版本的值替代了;“valid”是指所查询的数据存储器上的当前版本。“future”表明尚未有效的值,合理的原因如它可能用于发布某个指定的值。

其他任何带有前缀“other”的字符串可以用来表示服务器具体的新版本要素的状态。本标准没有解析带有前缀“other”的值的含义。

11.3.3.3 标准响应参数

见 7.7 中对标准响应参数的描述。

11.3.3.4 单一查询响应

如果 GetFeature 操作包含一个单一查询表达式,那么服务器返回一个包含 0 个或多个 wfs:member 元素的 wfs:FeatureCollection 元素,每个 wfs:member 元素包含或引用 GetFeature 操作的结果集中的一个要素。

11.3.3.5 多元查询响应

如果 GetFeature 操作包含多元查询表达式，那么服务器返回一个包含一个或者多个 wfs:collection 元素的 wfs:FeatureCollection 元素。每个 wfs:member 元素包含一个 wfs:FeatureCollection 元素，存放请求中一个查询表达式的结果。

组件要素集可以被直接编码到响应文件，也可以被相应文件引用，也可以两者兼而有之。

示例 1：下面片段举例说明了一个对 GetFeature 操作的响应，该操作包含组件要素集被内联编码的多元查询表达式。

```
<? xml version = "1.0" encoding = "UTF - 8"? >
< wfs:FeatureCollection >
  timeStamp = "2008 - 08 - 01T22:47:02"
  numberMatched = "349" numberReturned = "300"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema - instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.opengis.net/gml/3.2
                http://schemas.opengis.net/gml/3.2.1/gml.xsd">
< wfs: member >
< wfs:FeatureCollection
      timeStamp = "2008 - 08 - 01T22:47:02"
      numberMatched = "15" numberReturned = "15">
  ...
</wfs:FeatureCollection >
</wfs:member >
< wfs:member >
< wfs:FeatureCollection
  timeStamp = "2008 - 08 - 01T22:47:04"
  numberMatched = "257" numberReturned = "257">
  ...
</wfs:FeatureCollection >
</wfs:member >
< wfs:member >
< wfs:FeatureCollection
  timeStamp = "2008 - 08 - 01T22:47:06"
  numberMatched = "77" numberReturned = "28">
  ...
</wfs:FeatureCollection >
</wfs:member >
</wfs:FeatureCollection >
```

示例 2：下面片段举例说明了包含多元查询表达式的 GetFeature 操作的响应，其中要素集以嵌入式编码形式引用内嵌要素集。

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

```
    timeStamp = "2008 - 08 - 01T22:47:02"
    numberMatched = "349" numberReturned = "300"
    xmlns:wfs = "http://www.opengis.net/wfs/2.0"
    xmlns:gml = "http://www.opengis.net/gml/3.2"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema - instance"
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.opengis.net/gml/3.2
                http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    < wfs:member xlink:href = "..."/>
    ...
    < wfs:member xlink:href = "..."/>
  </wfs:FeatureCollection >
```

符合一个 GetFeature 操作中多个查询条件的要素集用一个要素集合表示，并且由其他所有的要素集合所引用。

虽然服务器可以按照任意的顺序处理查询表达式，但是响应文档中要素集合的顺序与 GetFeature 操作中相应的查询表达式的顺序应相同。

11.3.3.6 连接查询响应

如果 GetFeature 操作包含实现连接查询的查询表达式，那么服务器使用 wfs:Tuple(元组)(见 10.3.2)元素返回满足这一连接查询表达式的所有要素元组。

示例 1：下面 XML 片段为对三种要素类型 ns1:FeatureTypeOne(要素类型一)，ns1:FeatureTypeTwo(要素类型二)和 ns1:FeatureTypeThree(要素类型三)进行连接查询的结果。

```
<? xml version = "1.0" encoding = "UTF - 8"? >
< wfs:FeatureCollection
timeStamp = "2008 - 08 - 01T22:47:02"
  numberMatched = "200"
  numberReturned = "200"
  xmlns:ns1 = "http://www.someserver.com/ns1"
  xmlns:ns2 = "http://www.someserver.com/ns2"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema - instance"
  xsi:schemaLocation = "http://www.someserver.com/ns1 ./ns1.xsd
            http://www.xomeserver.com/ns2 ./ns2.xsd
            http://www.opengis.net/wfs/2.0
            http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
            http://www.opengis.net/gml/3.2
            http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  < wfs:boundedBy >
    < gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
      < gml:lowerCorner > - 32.7638053894043 104.1429748535156 </gml:lowerCorner >
        < gml:upperCorner > 0 133.3553924560547 </gml:upperCorner >
    </gml:Envelope >
  </wfs:boundedBy >
  < wfs:member >
```

```
    <wfs:Tuple>
      <wfs:member>
        <ns1:FeatureTypeOne>...</ns1:FeatureTypeOne>
      </wfs:member>
      <wfs:member>
        <ns1:FeatureTypeTwo>...</ns1:FeatureTypeTwo>
      </wfs:member>
      <wfs:member>
        <ns2:FeatureTypeThree>...</ns2:FeatureTypeThree>
      </wfs:member>
    </wfs:Tuple>
  </wfs:member>
  <wfs:member>
      <wfs:Tuple>
  <wfs:member>
  <ns1:FeatureTypeOne> ... </ns1:FeatureTypeOne>
  </wfs:member>
  <wfs:member>
  <ns1:FeatureTypeTwo> ...</ns1:FeatureTypeTwo>
  </wfs:member>
  <wfs:member>
  <ns2:FeatureTypeThree>...</ns2:FeatureTypeThree>
  </wfs:member>
  </wfs:Tuple>
    </wfs:member>
    <wfs:member>
  <wfs:Tuple>
  <wfs:member xlink:href="..."/>
  <wfs:member xlink:href="..."/>
  <wfs:member>
  <ns2:FeatureTypeThree> ...</ns2:FeatureTypeThree>
  </wfs:member>
  </wfs:Tuple>
  </wfs:member>
    ...
  </wfs:FeatureCollection>
```

包含连接与非连接查询表达式的 GetFeature 操作返回包含 wfs:member 和 wfs:tuple 元素的混合要素集合。

示例 2：下面 XML 片段为包含连接与非连接查询表达式的 GetFeature 操作返回的响应文档。

```
  <? xml version = "1.0" encoding = "UTF - 8"? >
  <wfs:FeatureCollection
    timeStamp = "2008 - 08 - 01T22:47:02"
    numberMatched = "349"
    numberReturned = "300"
    xmlns = "http://www.someserver.com/myns"
```

```
xmlns:wfs = "http://www.opengis.net/wfs/2.0"
xmlns:gml = "http://www.opengis.net/gml/3.2"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema - instance"
xsi:schemaLocation = "http://www.someserver.com/myns ./RoadRail.xsd
              http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.opengis.net/gml/3.2
              http://schemas.opengis.net/gml/3.2.1/gml.xsd">
<wfs:boundedBy>
  <gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
    <gml:lowerCorner> - 32.7638053894043 104.1429748535156 </gml:lowerCorner>
    <gml:upperCorner> 0 133.3553924560547 </gml:upperCorner>
  </gml:Envelope>
</wfs:boundedBy>
<wfs:member>
  <wfs:FeatureCollection
    timeStamp = "2008 - 08 - 01T22:47:02"
    numberMatched = "15"
    numberReturned = "15">
    <wfs:member>...</wfs:member>
    <wfs:member>...</wfs:member>
    ...
  </wfs:FeatureCollection>
</wfs:member>
<wfs:member>
  <wfs:FeatureCollection
    timeStamp = "2008 - 08 - 01T22:47:04"
    numberMatched = "257"
    numberReturned = "257">
    <wfs:member>
      <wfs:Tuple>
        <wfs:member>...</wfs:member>
        <wfs:member>...</wfs:member>
        ...
      </wfs:Tuple>
    </wfs:member>
    <wfs:member>
      <wfs:Tuple>
        <wfs:member>...</wfs:member>
        <wfs:member>...</wfs:member>
        ...
      </wfs:Tuple>
    </wfs:member>
    ...
  </wfs:FeatureCollection>
```

```
    </wfs:member>
    <wfs:member>
      <wfs:FeatureCollection
        timeStamp="2008-08-01T22:47:06"
        numberMatched="77"
        numberReturned="28">
        <wfs:member>...</wfs:member>
        <wfs:member>...</wfs:member>
        ...
      </wfs:FeatureCollection>
    </wfs:member>
  </wfs:FeatureCollection>
```

11.3.4 附加对象

如果在响应文档中需要包含附加对象以满足对被引用资源(见 7.6.4)的解析,这些附加对象需要放在 wfs:AdditionalObjects(附加对象)元素中,并且在返回的要素中资源引用要重新定位到这些被引用资源的本地副本上。如 7.6.4.5 中所述,服务器同时要做出调整,将所有没有解析的嵌套的资源引用重新定位以指向这些准备好的资源。

11.3.5 GetFeatureById 响应

实现包含 GetFeatureById 存储的查询(见 7.9.3.6)的 GetFeature 操作,产生一个响应,该响应由依据 outputFormat 参数(见 7.6.3.7)值编码的单个要素组成。该响应只包含 XML 要素,不包含其他由于执行 GetFeature 操作(见 11.3.3.4,11.3.3.5 和 11.3.3.6)所产生的要素。

11.4 异常

如果 WFS 在解析 GetFeature 请求时遇到错误,那么将抛出一个如 7.5 中所描述的 OperationParsingFailed 异常。

如果 WFS 在处理 GetFeature 请求时遇到错误,那么将抛出一个如 7.5 中所描述的 OperationProcessingFailed 异常。

12 LockFeature 操作

12.1 概述

网络连接本身是无状态的,因此,序列化事务的语义是不被保存的。为了理解这一点,以更新操作为例。

一个客户获取了一个要素实例,接着这个要素在客户端被修改,并且通过 Transaction 请求向 WFS 提交更新命令。由于没有任何事务可以保证要素在客户端被修改时,没有其他客户同时更新数据库中的同一要素,所以不存在连续性。

保证连续性的一个方法是要求访问数据的操作是互相排斥的,即当一个事务访问一个数据项时,其他事务不能修改这个数据项。这可以通过使用控制数据访问的锁来实现。

LockFeature 操作的目的是使用长期要素锁机制来保证一致性。这个锁被认为是长期的,因为网络的延迟使要素锁要比本机的商用数据库中的锁时间长很多。

如果 WFS 实现了 LockFeature 操作,那么应在它的能力描述文档(见 8.3)中声明此事实。

12.2 请求

12.2.1 请求语义

图 19 描述了 LockFeature 请求的模式。

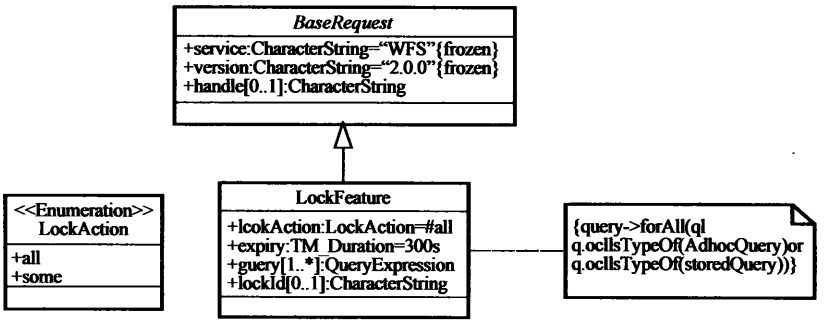

图 19 LockFeature 请求

12.2.2 XML 编码

下面的 XML 模式片段定义了 LockFeature 请求的 XML 编码：

```
<xsd:element name = "LockFeature" type = "wfs:LockFeatureType"/>
<xsd:complexType name = "LockFeatureType">
  <xsd:complexContent>
    <xsd:extension base = "wfs:BaseRequestType">
      <xsd:sequence>
        <xsd:AbstractQueryExpression maxOccurs = "unbounded"/>
      </xsd:sequence>
      <xsd:attribute name = "lockId" type = "xsd:string"/>
      <xsd:attribute name = "expiry" type = "xsd:positiveInteger" default = "300"/>
      <xsd:attribute name = "lockAction" type = "wfs:AllSomeType" default = "ALL"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:simpleType name = "AllSomeType">
  <xsd:restriction base = "xsd:string">
    <xsd:enumeration value = "ALL"/>
    <xsd:enumeration value = "SOME"/>
  </xsd:restriction>
</xsd:simpleType>
```


12.2.3 KVP 编码

表 18 定义了 LockFeature 操作的 KVP 编码。

表 18 LockFeature 操作的 KVP 编码关键字

URL 关键字	O/M[a]	默认值	描述
公共关键字 (REQUEST=LockFeature)			见表 7(只包括所有操作或者 LockFeature 操作的关键字)
即席的查询关键字(与存储的查询表达式和 LOCKID 关键字相互排斥)			见表 8
存储的查询表达式(与即席查询表达式和 LOCKID 关键字相互排斥)			见表 10
LOCKID (锁标识符)	O		指定一个存在的锁标识符,以重新设置锁期限
EXPIRY (期限)	O	300	如果没有收到解锁的 Transaction 请求,此参数值表示锁在被清除之前坚持的秒数
LOCKACTION (锁定动作)	O	ALL	指定锁请求的方式。值 ALL 表示操作成功则所有的要素被锁定,否则操作抛出一个 CannotLockAllFeatures 异常。值 SOME 表示服务器锁定尽可能多的要素。尽管返回的 lockId 可能没有锁住任何要素,但是操作依然是成功的
[a] O = Optional(可选的);M = Mandatory(必选的)。			

12.2.4 参数讨论

12.2.4.1 AbstractQueryExpression 参数

LockFeature 请求在一组可识别的要素集上使用一个或多个查询表达式进行操作。

对于 XML 编码的请求,本标准定义了 wfs:Query (见 7.9.2.2)和 wfs:StoredQuery (见 7.9.3.2)两个元素。这两个元素可以在 LockFeature 请求中用作查询表达式,并且替代 fes:AbstractQueryExpression (见 ISO 19143:2010, 6.2)。

可以定义其他的元素来取代 fes:AbstractQueryExpression,但是,本标准只赋予了 wfs:Query 和 wfs:StoredQuery 两个元素的含义。

即席查询表达式的 KVP 编码在 7.9.2.3 中已有定义。多个即席查询表达式可以编码成一个 KVP 编码的请求。

存储的查询表达式的 KVP 编码在 7.9.3.3 中已有定义。一个 KVP 编码的请求中只能包含一个存储的查询表达式。

KVP 编码的请求只包含一种类型的查询表达式,要么为即席查询表达式,要么为存储的查询表达式。

12.2.4.2 lockId 参数

LockFeature 操作中的 lockId 参数用来重新设置一个现有锁的终止时间。

示例 1:下面 XML 编码的请求将指定的锁的终止时间重新设置为 600 s。

```
<wfs:LockFeature lockId="1013" expiry="600"/>
```

这个 LockFeature 操作应在指定的锁终止之前实现;否则服务器将抛出 LockHasExpired 异常(见表 3)。

终止时间应根据获取的初始锁定时间来重新设置。

示例 2：如果一个锁最初要求 300 s，50 s 后锁的终止时间重新设置为 600 s，那么这个锁还能够持续 550 s 的时间。

12.2.4.3 expiry 参数

如果事务没有声明会终止这个锁，那么 expiry 参数用来设定 WFS 对要素实例锁定的期限。

Expiry 期限用秒来表示。

一旦整个锁定请求被处理，锁的期限计时就开始了，然后锁定的响应可以完整地传到客户端。

一旦指定的时间段过去了，WFS 将释放这个要素上现存的锁。与相同的标识符的锁有关的进一步事务将会失败，服务将抛出一个如 7.5 所描述的 InvalidParameterValue 异常。

如果 expiry 参数没有赋值，那么默认的期限值为 300 s。

如同 12.2.4.2 中所描述，一个现存的锁的终止时间可以延长。

12.2.4.4 lockAction 参数

可选参数 lockAction 是用来控制要素如何被锁定的。

一个为 ALL 的锁定操作表明 WFS 锁定了所有请求的要素实例。如果所有请求的要素实例不能被锁住，那么操作失效，服务应抛出一个如 7.5 所示的 CannotLockAllFeatures 异常，并且没有要素实例仍被锁定。

值为 SOME 的锁定操作表示 WFS 应请求一个可以锁定能够请求的要素实例的锁。

默认的锁定操作值应是 ALL。

图 20 描述了 LockFeature 操作的状态机制。

12.2.5 WFS 锁定的状态机制

本条定义了提供 WFS 接口的服务器锁定状态的状态机制。状态图显示了状态间允许的转变。所有其他的状态转换是不允许的，并且如果进行了其他的状态转换，服务器将报告错误。

一个服务器支持对多组要素进行锁定，其中每个要素只能被一个锁锁定。WFS 规范中定义的每个锁都是独立的。

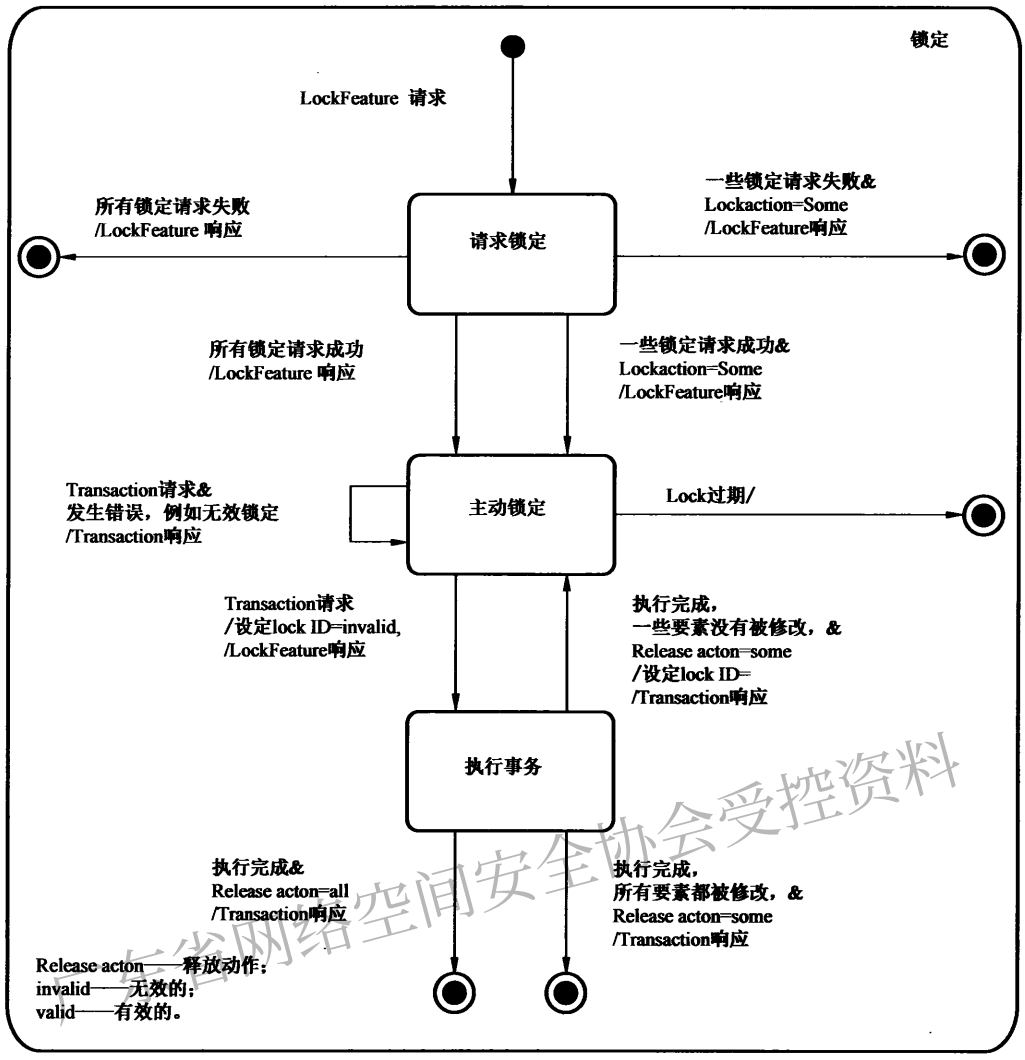

图 20　WFS 锁的状态图

12.3　响应

12.3.1　响应语义

图 21 描述了 LockFeature 响应的模式。

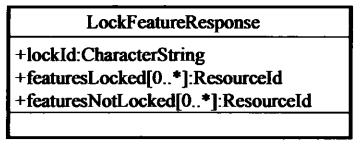

图 21　LockFeature 响应

12.3.2　XML 编码

下面的 XML 模式片段定义了 LockFeature 响应的 XML 编码：

```
< xsd:element name = "LockFeatureResponse" type = "wfs:LockFeatureResponseType"/>
```

```
<xsd:complexType name = "LockFeatureResponseType">
    <xsd:sequence>
    <xsd:element name = "FeaturesLocked"
                type = "wfs:FeaturesLockedType" minOccurs = "0"/>
    <xsd:element name = "FeaturesNotLocked"
                type = "wfs:FeaturesNotLockedType" minOccurs = "0"/>
    </xsd:sequence>
    <xsd:attribute name = "lockId" type = "xsd:string"/>
</xsd:complexType>
<xsd:complexType name = "FeaturesLockedType">
    <xsd:sequence maxOccurs = "unbounded">
      <xsd:element ref = "fes:ResourceId"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name = "FeaturesNotLockedType">
    <xsd:sequence maxOccurs = "unbounded">
      <xsd:element ref = "fes:ResourceId"/>
    </xsd:sequence>
</xsd:complexType>
```

为了响应 LockFeature 请求，WFS 应生成一个 wfs:LockFeatureResponse 元素。该文档应包含一个锁标识符，客户端应用程序可以在后继 WFS 操作中使用这个锁标识符来操作被锁的要素实例集。响应中可以包括可选的元素 wfs:FeaturesLocked 和 wfs:FeaturesNotLocked 元素，具体取值依赖于 lockAction 属性值。

如果锁动作指定为 ALL 并且所有标识的要素实例都被成功锁定，WFS 应返回一个 wfs:WFS_LockFeatureResponse 元素，其中包含 wfs:FeaturesLocked 元素而不包含 wfs:FeatureNotLocked 元素，因此所有标识的要素实例要么全部被锁定，要么都不能被锁定。如果一些或所有要素实例不能被锁，那么 WFS 应返回一个指明锁定请求失败的异常，因为一些或者所有要素实例被其他客户端锁定了。

如果锁动作指定为 SOME，那么 wfs:WFS_FeatureResponse 元素应包含 wfs:FeaturesLocked 和 wfs:FeaturesNotLocked 元素。wfs:FeaturesLocked 元素列出 LockFeature 请求锁定的所有要素实例的要素标识符，wfs:FeaturesNotLocked 元素包含不能被 WFS 锁定的要素实例的要素标识符列表(可能因为他们已被其他客户端锁定)。如果一个锁请求没有成功锁定要素，那么 WFS 返回一个 wfs:WFS_LockFeatureResponse 文档，包括 lockId 属性值，但是不包括 wfs:FeaturesLocked 元素和 wfs:FeatureNotLocked 元素。换句话说，这是一个空响应。因为没有资源被锁，此请求完成以后，lockId 应立即释放。因为 lockId 值已经不存在，所以如果同样的 lockId 用于后续的事务中，会抛出一个如 7.5 所述的 InvalidLockId 异常。

本标准没有对锁标识符的格式做任何假定，唯一的要求是该标识符可以用事务请求的字符集表达。

12.4 异常

如果 WFS 没有实现 LockFeature 操作，那么它将生成一个 OperationNotSupported 异常，表示如果遇到这样一个请求，这个操作不被支持。

如果 WFS 支持 LockFeature 操作，解析请求时遇到错误，那么将抛出如 7.5 所述的 OperationParsingFailed 异常。

如果 WFS 支持 LockFeature 操作，处理请求时遇到错误，那么将抛出如 7.5 所述的 OperationProcessingFailed 异常。

13 GetFeatureWithLock 操作

13.1 概述

GetFeatureWithLock 操作和 GetFeature 操作（见第 11 章）在功能上是相似的，唯一不同的是在 GetFeatureWithLock 操作的响应中，WFS 不仅应生成与 GetFeature 操作相似的响应文档，而且应锁定结果集中的要素；甚至可能在后续的 Transaction 操作（见第 15 章）中更新这些要素。

13.2 请求

13.2.1 请求语义

图 22 描述了 GetFeatureWithLock 请求的模式。

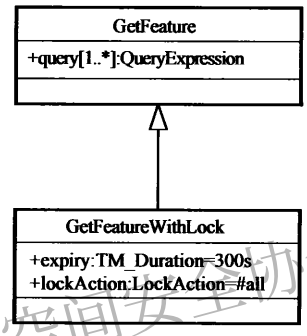

图 22 GetFeatureWithLock 请求

13.2.2 XML 编码

以下 XML 模式片段定义了 GetFeatureWithLock 操作的 XML 编码：

```
<xsd:element name = "GetFeatureWithLock" type = "wfs:GetFeatureWithLockType"/>
<xsd:complexType name = "GetFeatureWithLockType">
  <xsd:complexContent>
    <xsd:extension base = "wfs:GetFeatureType">
      <xsd:attribute name = "expiry" type = "xsd:positiveInteger" default = "300"/>
      <xsd:attribute name = "lockAction" type = "wfs:AllSomeType" default = "ALL"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

13.2.3 KVP 编码

GetFeatureWithLock 操作的 KVP 编码与 GetFeature 操作（见表 17）相似。表 19 列举了 GetFeatureWithLock 操作中可用的额外 KVP 参数。

表 19 GetFeatureWithLock 操作的 KVP 编码的额外关键字

URL 关键字	O/M[a]	默认值	描述
EXPIRY (期限)	O	300	只有当请求为 GetFeatureWithLock 时,此参数才被指定。此参数值为一个正整数,值指定为秒数,表示结果集中要素被锁定的时间长度。如果没有指定此参数,那么锁定的默认期限为 300 s
LOCKACTION (锁定动作)	O	ALL	指定锁请求的方式。值 ALL 表示操作成功则所有的要素被锁定,否则操作抛出一个 CannotLockAllFeatures 异常。值 SOME 表示服务器应锁定尽可能多的要素,尽管返回的 lockId 可能没有锁住任何要素,但是操作依然是成功的
[a] O = Optional(可选的);M = Mandatory(必选的)。			

13.2.4 参数讨论

13.2.4.1 expiry 参数

在没有接收到 Transaction 请求来解锁的情况下,expiry 参数用来设置 WFS 保持锁定状态的时间期限。默认锁定的持续时间为 300 s 或 5 min。

13.2.4.2 lockAction 参数

lockAction 参数控制 WFS 在结果集中对要素锁定的方式。

参数值 ALL(默认值)表示 WFS 试图对结果集中所有要素进行锁定。如果由于一部分要素已经被锁,导致不能够对所有要素锁定,那么 WFS 将抛出一个如 7.5 所描述的 CannotLockAllFeatures 异常。如果所有的要素都被成功锁定,那么 WFS 返回给客户端一个锁标识符(使用 wfs:FeatureCollection 元素中的 lockId 属性),这个标识符能用在后续对被锁的要素进行操作的 Transaction 操作中,并可用来解锁(见 15.2.3.1.2)。

参数值 SOME 表示 WFS 在结果集中锁定尽可能多的要素。响应文档中只包含锁定成功的要素,并且 WFS 返回给客户端一个锁标识符(使用 wfs:FeatureCollection 元素中的 lockId 属性),这个锁标识符能够用在后续对锁定的要素进行操作的 Transaction 操作中,并可用来解锁(见 15.2.3.1.2)。

13.2.4.3 resultType 参数

resultType 参数在 7.6.3.6 中有所描述。

GetFeatureWithLock 请求的 resultType 属性唯一有效的值为“result”。如果客户端使用了值“hits”,那么服务器将抛出一个 InvalidParameterValue 异常(见 7.5)。

13.3 响应

13.3.1 概述

GetFeatureWithLock 操作的响应与 GetFeature 操作的响应相似。唯一的不同点是 GetFeature-WithLock 操作的响应包含 lockId 参数值。

13.3.2 lockId 参数

对一个 GetFeatureWithLock 请求,WFS 生成一个包含用来在结果集中锁定要素的锁标识符的响应文档。这个锁标识符在响应文档中用 lockId 属性进行编码,lockId 属性在 wfs:FeatureCollection 元

素(见 11.3.2)中有声明。

示例：下面 XML 片段描述了如何在 GetFeatureWithLock 操作的响应中使用 lockId 属性。

```
<wfs:FeatureCollection lockId = "00A01"… >
…
</wfs:FeatureCollection>
```

省略号表示 GetFeatureWithLock 响应中其他所有与 GetFeature 响应的内容一样。

13.4 异常

如果 WFS 没有实现 GetFeatureWithLock 操作，而又收到对此操作的请求，那么它将生成一个 OperationNotSupported 异常，表明不支持这个操作。

如果 WFS 支持 GetFeatureWithLock 操作，并在解析此请求时遇到错误，那么它将抛出一个如 7.5 所描述的 OperationParsingFailed 异常。

如果 WFS 支持 GetFeatureWithLock 操作，并在处理此请求时遇到错误，那么它将抛出一个如 7.5 所描述的 OperationProcessingFailed 异常。

14 存储的查询表达式的管理

14.1 概述

本条阐述了 ListStoredQueries(列举存储的查询)、DescribeStoredQueries(描述存储的查询)、CreateStoredQuery(创建存储的查询)和 DropStoredQuery(终止存储的查询)的各项操作。

所有服务器应支持 ListStoredQueries 和 DescribeStoredQueries 操作，并且实现 GetFeatureById 存储的查询(见表 1 和 7.9.3.6)功能。

实现管理存储的查询一致性类(见表 1)的服务器应在它们的能力描述文档中通过 ManageStoredQueries(管理存储的查询)条件(见表 13)予以申明，并且应实现 CreateStoredQuery 和 DropStoredQuery 操作。

14.2 定义存储的查询

14.2.1 XML 编码

下面的 XML 模式片段定义了描述或定义一个存储的查询表达式的 XML 编码。

```
<xsd:complexType name = "StoredQueryDescriptionType">
  <xsd:sequence>
    <xsd:element ref = "wfs:Title" minOccurs = "0" maxOccurs = "unbounded"/>
    <xsd:element ref = "wfs:Abstract" minOccurs = "0" maxOccurs = "unbounded"/>
    <xsd:element ref = "ows:Metadata" minOccurs = "0" maxOccurs = "unbounded"/>
    <xsd:element name = "Parameter"
                 type = "wfs:ParameterExpressionType"
                 minOccurs = "0" maxOccurs = "unbounded"/>
      <xsd:element  name = "QueryExpressionText" type = "wfs:QueryExpressionTextType"
                 minOccurs = "1" maxOccurs = "unbounded"/>
  </xsd:sequence>
  <xsd:attribute name = "id" type = "xsd:anyURI" use = "required"/>
</xsd:complexType>
<xsd:complexType name = "ParameterExpressionType">
```

```
    <xsd:sequence>
      <xsd:element ref="wfs:Title" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="wfs:Abstract" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="ows:Metadata" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="type" type="xsd:QName" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="QueryExpressionTextType" mixed="true">
    <xsd:choice>
      <xsd:any namespace="##other" processContents="skip"
        minOccurs="0" maxOccurs="unbounded"/>
      <xsd:any namespace="##targetNamespace" processContents="skip"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:attribute name="returnFeatureTypes"
      type="wfs:ReturnFeatureTypesListType" use="required"/>
    <xsd:attribute name="language" type="xsd:anyURI" use="required"/>
    <xsd:attribute name="isPrivate" type="xsd:boolean" default="false"/>
  </xsd:complexType>
  <xsd:simpleType name="ReturnFeatureTypesListType">
    <xsd:list itemType="xsd:QName"/>
  </xsd:simpleType>
```

类型 wfs:StoredQueryDescriptionType(存储的查询描述类型)用作声明 wfs:StoredQueryDescription(存储的查询描述)元素和 wfs:StoredQueryDefinition(存储的查询定义)元素,前者为 DescribeStoredQueries(描述存储的查询)操作(见 14.4)的响应文档提供一个对存储的查询的描述;后者用于 CreateStoredQuery(创建存储的查询)操作(见 14.5)中定义一个存储的查询。

存储的查询的描述和定义包含了描述存储的查询的元数据、一组存储的查询所接受的零个或多个参数,以及当存储的查询被调用时所执行的一个或多个组件查询表达式。

14.2.2 参数讨论

14.2.2.1 Title(标题)参数

Title 参数允许用一种或多种语言为一个存储的查询赋予人类能够理解的标题。如果有多于一个 Title 值,则每个 Title 值对应的 xml:lang 属性的值应不同。

一个 Title 字符串的语言应用两个字符组成的语言代码来声明,缺省值为“en”。两个字符组成的语言代码的描述见 IETF RFC 4646。

14.2.2.2 Abstract(摘要)参数

Abstract 参数可用于把人类能阅读的阐述赋给一个存储的查询。如果有多个 wfs:Abstract 值,则每个 wfs:Abstract 值相应的 xml:lang 属性的值不同。

一个 Abstract 字符串的语言可以用两个字符组成的语言代码来声明,缺省值为“en”。两个字符组成的语言代码的描述见 IETF RFC 4646。

14.2.2.3 Metadata(元数据)参数

Metadata 参数用于内联编码或引用关于一个存储的查询的更加详细元数据。有关 ows:Metadata 元素见 OGC 06-121r3 文档。

14.2.2.4 Parameter(参数)参数

14.2.2.4.1 概述

一个存储的查询表达式可以有零个或多个参数,每一个参数都应在定义存储的查询时通过 wfs:Parameter 元素列举出来。

14.2.2.4.2 Title(标题)、Abstract(摘要) 和 Metadata(元数据)参数

有关 Title,Abstract 和 Metadata 参数的描述见 14.2.2.1、14.2.2.2 和 14.2.2.3。

14.2.2.4.3 name(名称)参数

存储的查询的每个参数的名称应通过 wfs:Parameter 元素的名称属性来进行编码。

14.2.2.4.4 type(类型)参数

存储的查询的每个参数的类型通过 wfs:Parameter 元素的类型属性来进行编码。类型属性的值是 QName,用于识别描述相应参数的内容模型的类型。

示例 1: type="xsd:double"指定一个已存储的查询参数的类型是双精度型。

示例 2: type="gml:PolygonPropertyType"指定一个已存储的查询参数的类型是 GML 的多边形。

14.2.2.5 QueryExpressionText(查询表达式文本)参数

14.2.2.5.1 概述

当一个存储的查询实现被调用时,wfs:QueryExpressionText 元素用于列举一个或多个组件查询表达式。

每一个 wfs:QueryExpressionText 元素可能包含一个 wfs:Query(见 7.9.2.2)元素,或者一个 wfs:StoredQuery (见 7.9.3.2)元素,或者若干个其他特定实现的内容用于指定用另外一种实现语言编码的组件查询表达式 (见 14.2.2.5.3)。

在一个组件查询表达式的文本中,用"${argument_name}"标记来表明在运行时一个存储的查询表达式的参数值应被替代的位置。如果参数值需要在多个地方被替代,"${argument_name}"标记会出现多次。argument_name 标记用于指明存储的查询表达式中一个参数的名称所在的位置。

注: 下列要存储的查询表达式用于查找调用"Features In Polygon"存储的查询时落在用户指定的矩形范围内属于 FeatureType1,FeatureType2 和 FeatureType3 类型的所有地理要素。"${AreaOfInterest}"标记表明在运行时 AreaOfInterest 参数值需要被代替的位置。在这个注中标记在三个地方出现。

```
<? xml version = "1.0"? >
< wfs:CreateStoredQuery
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.org/fes/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema - instance"
  xmlns:myns = "http://www.someserver.com/myns"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
```

```
                http://schemas.opengis.net/wfs/2.0/wfs.xsd"
    service = "WFS"
    version = "2.0.0">
    < wfs:StoredQueryDefinition id = "urn:StoredQueries:FeaturesInPolygon">
      < wfs:Title > Features In Polygon </wfs:Title >
      < wfs:Abstract > Find all the features in a Polygon.</wfs:Abstract >
      < wfs:Parameter name = "AreaOfInterest" type = "gml:PolygonPropertyType"/>
      < wfs:QueryExpressionText
         returnFeatureTypes = "myns:Parks myns:Lakes myns:Rivers"
         language = "urn:ogc:def:queryLanguage:OGC - WFS::WFS_QueryExpression"
         isPrivate = "false">
        < wfs:Query typeNames = "myns:Parks">
          < fes:Filter >
            < fes:Within >
              < fes:ValueReference > geometry </fes:ValueReference >
              $ {AreaOfInterest}
            </fes:Within >
          </fes:Filter >
        </wfs:Query >
        < wfs:Query typeNames = "myns:Lakes">
          < fes:Filter >
            < fes:Within >
              < fes:ValueReference > region </fes:ValueReference >
              $ {AreaOfInterest}
            </fes:Within >
          </fes:Filter >
        </wfs:Query >
        < wfs:Query typeNames = "myns:Rivers">
          < fes:Filter >
            < fes:Within >
              < fes:ValueReference > region </fes:ValueReference >
              $ {AreaOfInterest}
            </fes:Within >
          </fes:Filter >
        </wfs:Query >
      </wfs:QueryExpressionText >
    </wfs:StoredQueryDefinition >
  </wfs:CreateStoredQuery >
```

14.2.2.5.2 声明返回的要素类型

wfs:QueryExpressionText 要素的 ReturnFeatureTypes(返回要素类型)属性用于指定每个组件查询表达式返回的要素类型。

如果不止一个 ReturnFeatureType 列出,表明相应的组件查询表达式返回一组已列出的要素类型。

返回的要素类型的名称应与该服务能力描述文档(见 8.3.4)中列出的要素类型名称相一致。

14.2.2.5.3 实现语言

作为 wfs:QueryExpressionText 要素的属性 language 参数用于指定内嵌查询表达式所采用的实现语言。

遵循本标准的服务器应支持"urn:ogc:def:queryLanguage:OGC-WFS::WFSQueryExpression"属性值,标明组件查询表达式是用 wfs:Query(见 7.9.2.2)或者 wfs:StoredQuery(见 7.9.3.2)元素指定。

服务器也可以支持其他实现语言,但是本标准中没有为其他语言参数值设定任何含义。如果一个服务器支持其他实现语言表明已存储的查询表达式,应在其能力描述文档中声明这些 language 参数值(见表 12)。

注:可以使用服务器在其能力描述文档(见 8.3)中声明所支持的其他语言来实现组件查询表达式文本的编码,包括 SQL、XQuery、XPath 和 SPARQL 语言。事实上,如果服务器支持可运行的编码语言如 Java,查询表达式也可以用该编码语言来表示。

14.2.2.5.4 IsPrivate(是否私有)参数

由 wfs:QueryExpressionText 元素指定的组件查询表达式的实现文本可以是私有的或公有的,私有的意味着实现文本只对存储的查询创建者可见,公有的是指实现文本对任何人都可见。

该参数是在创建存储的查询时设置的,它决定是否要在 DescribeStoredQueries 操作的响应中表现组件查询表达式的实现文本。

如果 IsPrivate 参数的值为"true"(真),服务器则不能将要组件查询表达式在 DescribeStoredQueries 操作的响应中表现出来。

如果 IsPrivate 参数的值为"false"(假),服务器则将要组件查询表达式在 DescribeStoredQueries 操作的响应中表现出来。

14.2.2.6 id 参数

id 参数用于指定一个的标记符,用来重复调用要存储的查询表达式。

14.3 ListStoredQueries 操作

14.3.1 请求语义

ListStoredQueries 操作列举了在一个服务器上已经存储的查询表达式(见图 23)。

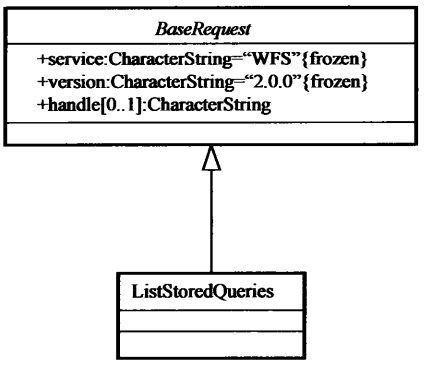

图 23 ListStoredQueries 请求

14.3.2 XML 编码

以下 XML 模式片段定义了 ListStoredQueries 操作的 XML 编码。

```
<xsd:element name = "ListStoredQueries" type = "wfs:ListStoredQueriesType"/>
  <xsd:complexType name = "ListStoredQueriesType">
    <xsd:complexContent >
      <xsd:extension base = "wfs:BaseRequestType"/>
    </xsd:complexContent >
  </xsd:complexType >
```

14.3.3 KVP 编码

表 20 定义了 ListStoredQueries 操作的 KVP 编码。

表 20 ListStoredQueries 的 KVP 编码中的关键字

URL 关键字	描 述
公共关键字(REQUEST=ListStoredQueries)	见表 7(只包括所有操作或者 ListStoredQueries 操作的关键字)

14.3.4 响应

图 24 描述了 ListStoredQueries 操作的响应。

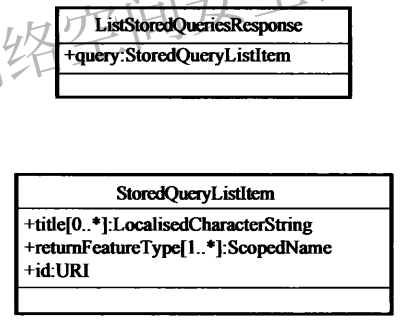

图 24 ListStoredQueriesResponse

以下 XML 模式片段定义了 ListStoredQueries 操作的响应。

```
<xsd:element name = "ListStoredQueriesResponse"
             type = "wfs:ListStoredQueriesResponseType"/>
  <xsd:complexType name = "ListStoredQueriesResponseType">
    <xsd:sequence >
      <xsd:element name = "StoredQuery" type = "wfs:StoredQueryListItemType"
                   minOccurs = "0" maxOccurs = "unbounded"/>
    </xsd:sequence >
  </xsd:complexType >
  <xsd:complexType name = "StoredQueryListItemType">
    <xsd:sequence >
      <xsd:element ref = "wfs:Title" minOccurs = "0" maxOccurs = "unbounded"/>
```

```
    <xsd:element name = "ReturnFeatureType"
                 type = "xsd:QName" maxOccurs = "unbounded"/>
  </xsd:sequence>
  <xsd:attribute name = "id" type = "xsd:anyURI" use = "required"/>
</xsd:complexType>
```

ListStoredQueries 操作的响应根元素是 wfs:ListStoredQueriesResponse 元素，它包含一个或多个 wfs:StoredQuery 元素，每个元素描述一个服务器提供的存储的查询表达式。有关 wfs:Title，wfs: ReturnFeatureType 和 id 属性的描述见 14.2。

14.3.5 异常

当服务器解析 ListStoredQueries 操作遇到错误时，会抛出一个如 7.5 中描述的 OpertationParsingFailed 异常信息。

当服务器处理 ListStoredQueries 操作遇到错误时，会抛出一个如 7.5 中描述的 OpertationProcessingFailed 异常信息。

14.4 DescribeStoredQueries 操作

14.4.1 请求语义

DescribeStoredQueires 操作提供了有关一个服务器提供的每个存储的查询表达式的详细元数据（见图 25）。

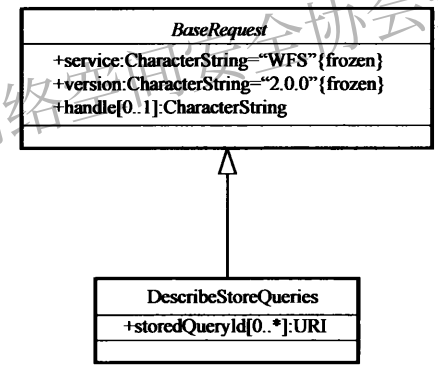

图 25 DescribeStoredQueries 请求

14.4.2 XML 编码

以下 XML 模式片段定义了 DescribeStoredQueires 操作的 XML 编码。

```
<xsd:element name = "DescribeStoredQueries"
             type = "wfs:DescribeStoredQueriesType"/>
<xsd:complexType name = "DescribeStoredQueriesType">
  <xsd:complexContent>
    <xsd:extension base = "wfs:BaseRequestType">
      <xsd:sequence>
        <xsd:element name = "StoredQueryId" type = "xsd:anyURI"
                     minOccurs = "0" maxOccurs = "unbounded"/>
      </xsd:sequence>
```

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

wfs: DescribeStoredQueries 元素包含零个或多个 wfs:StoredQueryId 元素，列举要描述的存储的查询表达式的标识符。如果没有指定 wfs:StoredQueryId 元素，则对一个服务器上提供的所有存储的查询表达式进行描述。

14.4.3 KVP 编码

表 21 定义了 DescribeStoredQueries 操作的 KVP 编码。

表 21 DescribeStoredQueries 的 KVP 编码关键字

URL 关键字	O/M(可选/必选)	描述
公共关键字 (REQUEST=DescribeStoredQueries)		见表 7(只包括所有操作或者 DescribeStoredQueries 操作的关键字)
STOREDQUERY_ID (存储的查询标识符)	O	一组用逗号隔开的存储的查询表达式的标识符。如果没有指定该关键字的值，则列出一个服务器所提供的所有存储的查询表达式
[a] O = Optional(可选的);M = Mandatory(必选的)。		

14.4.4 响应

14.4.4.1 响应语义

图 26 描述了 DescribeStoredQueries 操作的响应语义。

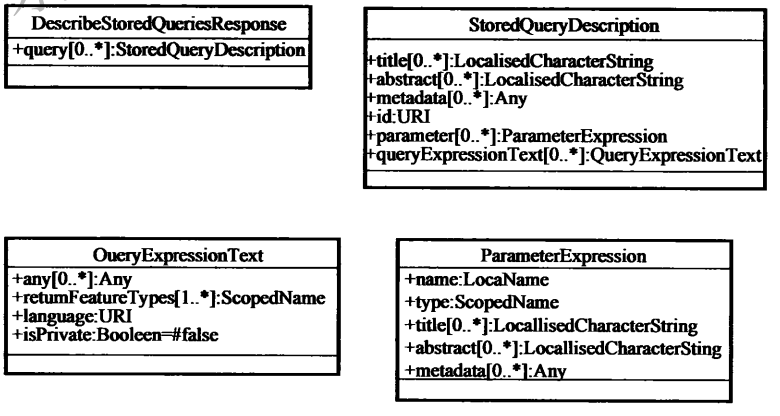

图 26 描述存储的查询的响应语义

14.4.4.2 XML 编码

以下 XML 模式片段定义了 DescribeStoredQueries 操作的响应。

```
<xsd:element name = "DescribeStoredQueriesResponse"
        type = "wfs:DescribeStoredQueriesResponseType"/>
  <xsd:complexType name = "DescribeStoredQueriesResponseType">
    <xsd:sequence>
```

```
    <xsd:element name = "StoredQueryDescription"
                 type = "wfs:StoredQueryDescriptionType"
                 minOccurs = "0" maxOccurs = "unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

该响应包含一个或多个描述存储的查询的 wfs:StoredQueryDescription 要素，wfs:StoredQueryDescriptionType 类型的描述见 14.2。

14.5 CreateStoredQuery 操作

14.5.1 请求语义

CreateStoredQuery 操作允许客户端创建要存储的查询表达式(见图 27)。

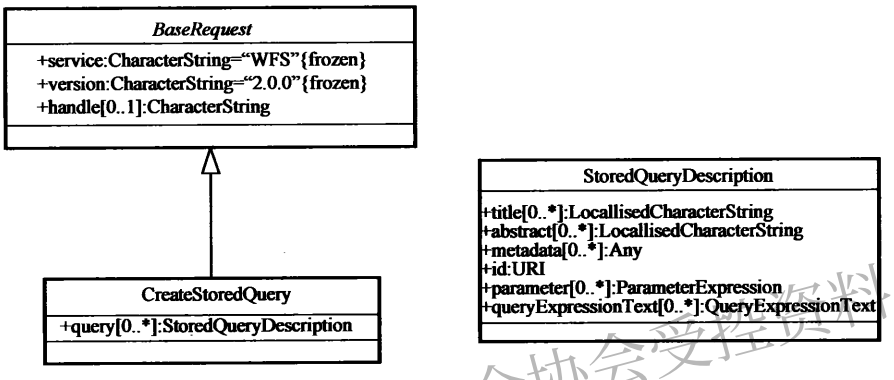

图 27 CreateStoredQuery 请求

并不是服务器所提供的所有存储的查询都需要通过 CreateStoredQuery 操作来创建。服务器可以被提前配置以提供任意数量的以任意方式实现的存储的查询。可以预期，本标准文档定义了符合 WFS 所应提供条件的存储的查询集。

注：可以进一步预期，在很多情况下，为了追求效率和效果，提前配置的存储的查询可以作为可运行代码来实现，针对复杂的查询逻辑，也允许此类实现方式，并且隐藏于一个简单的存储的查询界面中。

14.5.2 XML 编码

以下 XML 模式片段定义了 CreateStoredQuery 操作的 XML 编码。

```
<xsd:element name = "CreateStoredQuery"
  type = "wfs:CreateStoredQueryType"/>
<xsd:complexType name = "CreateStoredQueryType">
  <xsd:complexContent>
    <xsd:extension base = "wfs:BaseRequestType">
      <xsd:sequence>
        <xsd:element name = "StoredQueryDefinition"
          type = "wfs:StoredQueryDescriptionType"
          minOccurs = "0" maxOccurs = "unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

14.5.3 KVP 编码

本标准没有为 CreateStoredQuery 操作定义 KVP 编码。

14.5.4 参数讨论

使用 StoredQueryDefinition 编码的 StoredQueryDefinition 参数应包含一个已存储的查询的定义，多个要存储的查询可以在一个 CreateStoredQuery 请求中创建。

wfs:StoredQueryDescriptionType 类型的描述详见 14.2。

14.5.5 响应

图 28 描述了 CreateStoredQuery 操作的响应语义。

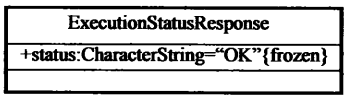

图 28 CreateStoredQuery 响应

以下 XML 模式片段定义了 CreateStoredQuery 操作的响应。

```
< xsd:element name = "CreateStoredQueryResponse"
    type = "wfs:CreateStoredQueryResponseType"/>
< xsd:complexType name = "ExecutionStatusType">
    < xsd:attribute name = "status" type = "xsd:string" fixed = "OK"/>
</xsd:complexType >
< xsd:complexType name = "CreateStoredQueryResponseType">
    < xsd:complexContent >
      < xsd:extension base = "wfs:ExecutionStatusType"/>
    </xsd:complexContent >
</xsd:complexType >
```

该响应的根元素 wfs: CreateStoredQuery 包含唯一的属性“status”(状态)。

该状态属性的唯一有效值是“OK”,标明要存储的查询成功创建,并保存在服务器上,否则该服务器将产生一个异常信息(见 7.5)。

14.6 DropStoredQuery 操作

14.6.1 请求语义

DropStoredQuery 操作允许从系统中终止前面创建的存储的查询(如图 29)。

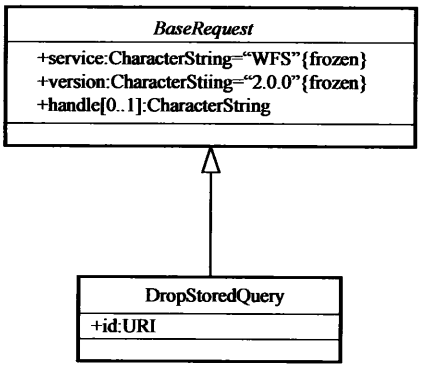

图 29 DropStoredQuery 请求

14.6.2 XML 编码

以下 XML 模式片段定义了 DropStoredQuery 操作的 XML 编码。

```
<xsd:element name = "DropStoredQuery">
  <xsd:complexType >
    <xsd:complexContent >
      <xsd:extension base = "wfs:BaseRequestType">
        <xsd:attribute name = "id" type = "xsd:anyURI" use = "required"/>
          </xsd:extension >
    </xsd:complexContent >
  </xsd:complexType >
</xsd:element >
```

该请求仅接受要终止的存储的查询的标识符，该标识符在 wfs：DropStoredQuery 元素中编码为 id 属性。

14.6.3 KVP 编码

表 22 定义了 DropStoredQuery 操作的 KVP 编码。

表 22 DropStoredQuery 的 KVP 编码关键字

URL 关键字	O/ M[a]（可选/必选）	描述
公共关键字 (REQUEST=DropStoredQuery)		见表 7(只包括所有操作或者 DescribeStoredQueries 操作的关键字)
STOREDQUERY_ID (存储的查询的标识符)	M	一组以逗号隔开的存储的查询。如果没有指定该关键字，则指服务器提供的所有存储的查询

[a] O = Optional(可选的)；M = Mandatory(必选的)。

14.6.4 响应

以下 XML 模式片段定义了 DropStoredQuery 操作的响应。

```
<xsd:element name = "DropStoredQueryResponse" type = "wfs:ExecutionStatusType"/>
```

wfs:ExecutionStatusType 类型的描述见 14.5.5。

14.7 异常

如果 WFS 服务没有实现 CreateStoredQuery 和 DropStoredQuery 操作，当该服务器接收到其中一种请求时，将会产生一个 OperationNotSupported 异常消息，标明服务器不支持该操作。

如果 WFS 服务器的确支持这两个操作，但是在解析请求过程中遇到错误时，服务器将会抛出 OperationParsingFailed 异常消息(见 7.5)。

如果 WFS 服务器的确支持这两个操作，但是在处理请求过程中遇到错误时，服务器将会抛出 OperationProcessingFailed 异常消息(见 7.5)。

15 Transaction 操作

15.1 概述

可选操作 Transaction 用来描述在 WFS 控制下应用于要素实例的数据事务操作。客户端可以使用 Transaction 操作创建、修改、替换和删除 WFS 提供的数据存储中的要素。GML(见 GB/T 23708—2009)被用作要素的规范表达，把规范的 GML 描述方式转换为用于数据存储的内部描述方式要通过特定的 WFS 实现来完成(内部描述方式也可能是 GML 形式，在这种情况下不需要转换)。

当事务实现完毕，WFS 应生成如 15.3 所描述的 XML 报告文档，来指明操作的完成状态。

支持可选 Transction 操作的 WFS 应在 8.3 所描述的能力描述文档中声明这一事实。

15.2 请求

15.2.1 请求语义

图 30 描述了 Transaction 请求的模式。

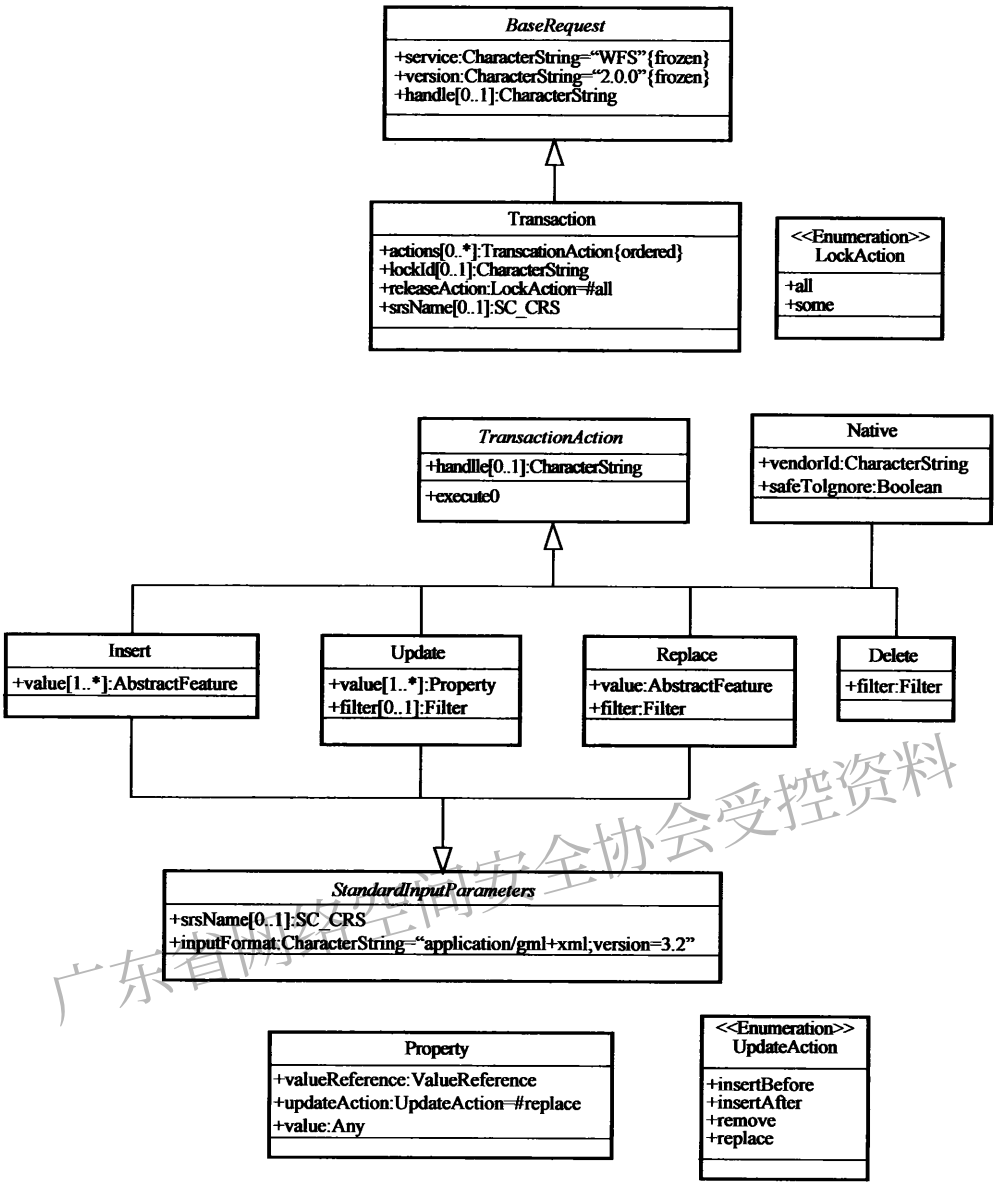

图 30　Transaction 请求

15.2.2　XML 编码

以下 XML 模式片段定义了 Transaction 请求的 XML 编码：

```
<xsd:element name = "Transaction" type = "wfs:TransactionType"/>
<xsd:complexType name = "TransactionType">
    <xsd:complexContent >
      <xsd:extension base = "wfs:BaseRequestType">
        <xsd:sequence >
          <xsd:sequence minOccurs = "0" maxOccurs = "unbounded">
            <xsd:element ref = "wfs:AbstractTransactionAction"/>
          </xsd:sequence >
        </xsd:sequence >
```

```
            <xsd:attribute name = "lockId" type = "xsd:string"/>
            <xsd:attribute name = "releaseAction" type = "wfs:AllSomeType" default = "ALL"/>
            <xsd:attribute name = "srsName" type = "xsd:string"/>
          </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name = "AbstractTransactionAction" type = "wfs:AbstractTransactionActionT-
ype"
          abstract = "true"/>
    <xsd:complexType name = "AbstractTransactionActionType" abstract = "true">
          <xsd:attribute name = "handle" type = "xsd:string"/>
    </xsd:complexType>
```

一个 wfs:Transaction 应包含 0 个或多个可替换的 wfs:AbstractTransactionAction 元素。

wfs:AbstractTransactionAction 元素包含属性句柄，用于标记由特定名称指定的客户端的事务动作，以便于获取错误句柄。

本标准定义了用来替换 wfs:AbstractionTransactionAction 元素的 wfs:Insert(插入)、wfs:Update(更新)、wfs:Replace(替换)和 wfs:Delete(删除)四个元素。这四个元素可以用来在 wfs:Transaction 元素中对创建、修改、替换和删除要素的操作进行编码。

WFS 应按照 wfs:Insert，wfs:Update，wfs:Replace 和 wfs:Delete 元素在 transaction 请求中的顺序来处理它们。在 transaction 请求中后续的 wfs:Replace 和 wfs:Delete 操作可以处理之前同一个 transaction 请求中插入操作所创建的要素。

wfs:Transaction 元素可以包含 0 个或者多个 wfs:Native 元素或者用于替换 wfs:Native 元素的其他元素，用来描述供应商指定的操作(见 8.4)。

一个空的 wfs:Transaction 元素是一个有效请求，能够用来解除任何被锁定要素的锁，解锁方式是通过将要解除的锁标识符赋予 lockId 属性(见 12.2.4.2)，并且将 releaseAction(释放动作)属性值设置为 ALL。

15.2.3 参数讨论

15.2.3.1 Locking(锁定)参数

15.2.3.1.1 声明支持锁定

如果服务器支持通过 AutomaticDataLocking(自动数据锁定)约束(见表 14)实现自动数据锁定，那么服务器应在能力描述文档(见 8.3)中声明这一事实。

支持自动数据锁定的服务器不会在使用 Transaction 操作对要素进行处理之前要求客户端对要素锁定。这种服务器可以选择性地实现 LockFeature 操作(见第 12 章)或者 GetFeatureWithLock 操作(见第 13 章)，并不要求在 transaction 操作中指定 lockId 属性，除非之前已经使用 LockFeature 操作或 GetFeatureWithLock 操作将要素锁定。

不支持自动数据锁定的服务器要实现一个或多个 LockFeature 操作和 GetFeatureWithLock 操作，并要求客户端在对要素进行操作之前锁定要素。若发生错误，那么服务器将抛出一个 FeaturesNotLocked 异常(见 7.5)。

15.2.3.1.2 lockId 参数

lockId 参数编码为 wfs:Transaction 元素中的属性 lockId。

lockId 参数的内容是从 LockFeature 操作(见第 12 章)或 GetFeatureWithLock(见第 13 章)操作获得的锁标识符。

如果 WFS 接收了一个更新、替换或删除已锁定的要素的 Transaction 请求并且此 Transaction 请求不包括 wfs:LockId 属性,那么服务器将抛出一个 MissingParameterValue 异常(见 7.5)。

如果 WFS 接收了一个更新、替换或删除已锁定的要素的 Transaction 请求并且< wfs:LockId >属性值与目标要素的锁标识符不匹配,那么服务器将抛出一个 InvalidParameterValue 异常(见 7.5)。

Transaction 请求中的插入操作不受 lockId 属性的影响。如果一个 Transaction 操作只包含插入操作,那么不指定 lockId 属性。

一个 Transaction 请求不必使用某一锁标识符作用于所有被锁定的要素。

15.2.3.2 releaseAction(释放动作)属性

releaseAction 参数编码为 wfs:Transaction 元素的 releaseAction 属性,它控制当事务请求完成后被锁的要素的处理方式。

参数值 ALL 表明使用指定的 wfs:LockId 锁定的所有要素应在事务完成时被解锁,而不管锁定的要素集中的某一要素是否真正被操作过。如果 releaseAction 属性没有指定值,其默认值是 ALL。

参数值 SOME 表明只有事务修改过的要素要被解锁。其他没有修改过的锁定的要素实例应使用相同的锁标识符保持锁定状态,以便后继的事务可以操作这些要素实例。当 releaseAction 属性设为值 SOME,每个事务结束后期限计数器应置为 0,除非所有锁定的要素实例都被操作过。例如,如果一个客户应用锁住 20 个要素实例,接着提交了一个只处理其中 10 个要素实例的事务请求,那么值为 SOME 的 releaseAction 意味着当事务终止后,另外 10 个没改动的要素实例将继续保持锁定状态。由客户应用提交的后继事务操作可以使用相同的锁标识符修改剩余的这 10 个要素实例。

15.2.3.3 srsName 属性

srsName 参数编码为 wfs:Transaction 元素的 srsName 属性。

见 7.6.5.5 中对 srsName 参数的描述。

15.2.4 Insert 操作

15.2.4.1 XML 编码

以下 XML 模式片段定义了 wfs:Insert 元素:

```
<xsd:element name = "Insert" type = "wfs:InsertType" substitutionGroup = "wfs:Abstract-
TransactionAction"/>
<xsd:complexType name = "InsertType">
  <xsd:complexContent >
    <xsd:extension base = "wfs:AbstractTransactionActionType">
      <xsd:sequence >
        <xsd: any namespace = "##other" maxOccurs = "unbounded"/>
      </xsd:sequence >
      <xsd:attributeGroup ref = "wfs:StandardInputParameters"/>
    </xsd:extension >
  </xsd:complexContent >
</xsd:complexType >
```

wfs:Insert 元素用来在 WFS 的数据存储中创建新的要素实例。默认情况下,创建的要素的初始状态用 GML(见 GB/T 23708—2009)表达,并且应通过 DescribeFeatureType 操作(见第 9 章)生成的

GML 应用模式的有效性验证。在一个单一的事务请求中可以包含多个 wfs:Insert 元素，并且多个要素实例可以由一个单一的 wfs:Insert 元素创建。

15.2.4.2 标准输入参数

见 7.6.5 中对标准输入参数的描述。

15.2.5 Update 操作

15.2.5.1 XML 模式

以下 XML 模式片段定义了 wfs:Update 元素：

```
<xsd:element name = "Update" type = "wfs:UpdateType"
    substitutionGroup = "wfs:AbstractTransactionAction"/>
  <xsd:complexType name = "UpdateType">
    <xsd:complexContent >
      <xsd:extension base = "wfs:AbstractTransactionActionType">
        <xsd:sequence >
          <xsd:element ref = "wfs:Property" maxOccurs = "unbounded"/>
          <xsd:element ref = "fes:Filter" minOccurs = "0"/>
        </xsd:sequence >
        <xsd:attribute name = "typeName" type = "xsd:QName" use = "required"/>
        <xsd:attributeGroup ref = "wfs:StandardInputParameters"/>
      </xsd:extension >
    </xsd:complexContent >
  </xsd:complexType >
  <xsd:element name = "Property" type = "wfs:PropertyType"/>
  <xsd:complexType name = "PropertyType">
    <xsd:sequence >
      <xsd:element name = "ValueReference">
        <xsd:complexType >
          <xsd:simpleContent >
            <xsd:extension base = "xsd:string">
              <xsd:attribute name = "action" type = "wfs:UpdateActionType"
             default = "replace"/>
            </xsd:extension >
          </xsd:simpleContent >
        </xsd:complexType >
      </xsd:element >
      <xsd:element name = "Value" minOccurs = "0"/>
    </xsd:sequence >
  </xsd:complexType >
  <xsd:simpleType name = "UpdateActionType">
    <xsd:restriction base = "xsd:string">
      <xsd:enumeration value = "insertBefore"/>
      <xsd:enumeration value = "insertAfter"/>
```

```
        <xsd:enumeration value = "remove"/>
    <xsd:enumeration value = "replace"/>
      </xsd:restriction>
    </xsd:simpleType>
```

wfs:Update 元素包含一个或多个 wfs:Property 元素，wfs:Property 元素描述明确指定的要素类型的某一特性的值变化。一个 Transaction 请求中可以包含多个 wfs:Update 元素，以便于对同一或不同种类的要素实现多种更新。

15.2.5.2 参数讨论

15.2.5.2.1 Property 元素

wfs:Property 元素包括两个子元素：wfs:ValueReference(值引用)元素和可选的 wfs:Value(值)元素。

wfs:Value 元素要么包含由 wfs:ValueReference 元素指定的节点的替换值，要么没有内容，表示引用的节点赋予了值 NULL。如果根据要素类型的模式，引用的节点不允许赋予 NULL 值，那么服务器将抛出 InvalideValue 异常(见 7.5)。

wfs:ValueReference 元素包含一个指向要修改的要素特性或要素特性的一个子结点的路径表达式。路径表达式的第一步应是一个要素类型的有效特性名称，该要素类型由 wfs:Update 元素的 typeName(类型名称)属性指定，后续步骤应遵循 ISO 19143:2010 中 7.4.4 描述的 XPath 编码规则。

wfs:ValueReference 元素的 action(动作)属性控制引用节点的值是如何被更新的。

如果属性 action 的值为“insertBefore”(在…之前插入)，那么 wfs:Value 元素的内容则插入在 wfs:ValueReference 元素内容指定的节点之前新建的一个节点中。

如果属性 action 的值为“insertAfter”(在…之后插入)，那么 wfs:Value 元素的内容则插入在 wfs:ValueReference 元素内容指定的节点之后新建的一个节点中。

如果属性 action 的值为“remove”(删除)，那么 wfs:ValueReference 元素的内容所指定的节点的内容将被移除。在这种情况下，wfs:Value 元素可以省略。但是如果 action 属性值设置为“remove”，同时又指定了 wfs:Value 元素，那么 wfs:Value 元素将被忽略。

如果属性 action 的值为“replace”(替换)，以 wfs:Value 要素的内容代替 wfs:ValueReference 元素的内容所指定的节点的内容。“replace”值是属性 action 的默认值。

在任何情况下，如果指定的 action 结果依据 DescribeFeatureType 操作(见第 9 章)获取的要素类型模式使要素实例无效，那么服务器将抛出一个 InvalidValue 异常(见 7.5)。

15.2.5.2.2 Filter 元素

一个用 fes:Filter 元素(见 ISO 19143:2010，第 7 章)编码的过滤表达式可以将一个 update 操作的范围限定在一组枚举的要素或者一组由空间和非空间约束条件指定的要素内。

如果 fes:Filter 元素没有标识任何被操作的要素，那么 update 操作没有任何效果。这不是异常情况，所以 WFS 不会抛出异常。

fes:Filter 元素在 ISO 19143:2010 第 7 章中有完整定义。

15.2.5.2.3 typeName 属性

typeName 属性值用来指定将被更新的要素类型。它的值应是在 WFS 的能力描述文档(见 8.3)中要素类型列表中所列出的要素类型中的一个。

15.2.5.2.4 标准输入参数

7.6.5 中描述了标准输入参数。

15.2.6 Replace 操作

15.2.6.1 XML 编码

以下 XML 模式片段定义了 wfs:Replace 元素：

```
<xsd:element name = "Replace" type = "wfs:ReplaceType"
substitutionGroup = "wfs:AbstractTransactionAction"/>
  <xsd:complexType name = "ReplaceType">
    <xsd:complexContent>
      <xsd:extension base = "wfs:AbstractTransactionActionType">
        <xsd:sequence>
          <xsd: any namespace = "##other"/>
          <xsd:element ref = "fes:Filter"/>
        </xsd:sequence>
        <xsd:attributeGroup ref = "wfs:StandardInputParameters"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

与修改单个要素特性的更新操作不同的是，替换操作是用指定的要素替换现存的要素。如果服务器支持版本化，那么这个操作可以创建最新的版本。

15.2.6.2 参数讨论

15.2.6.2.1 Filter 表达式

必选的过滤器表达式(见 ISO 19143:2010，第 7 章)用来指定要素集中哪一个或哪些要素被 wfs:Repalce 元素指定的要素实例替换。

15.2.6.2.2 标准输入参数

见 7.6.5 对标准输入参数的描述。

15.2.7 Delete 操作

15.2.7.1 XML 编码

以下 XML 模式片段声明了 wfs:Delete 元素：

```
  <xsd:element name = "Delete" type = "wfs:DeleteType"
substitutionGroup = "wfs:AbstractTransactionAction"/>
  <xsd:complexType name = "DeleteType">
    <xsd:complexContent>
      <xsd:extension base = "wfs:AbstractTransactionActionType">
        <xsd:sequence>
          <xsd:element ref = "fes:Filter"/>
```

```
        </xsd:sequence>
        <xsd:attribute name="typeName" type="xsd:QName" use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

wfs:Delete 元素用于对一个删除请求进行编码，删除一个或多个指定要素类型的要素实例，使客户端不能再通过 GetFeature 操作（见 11 章）、GetFeatureWithLock 操作（见 13 章）和 GetPropertyValue 操作（见第 10 章）查询这些要素。这意味着从数据存储中删除要素，或者如果数据存储支持版本化，那么这些要素将成为老的数据版本。

15.2.7.2 参数讨论

15.2.7.2.1 typeName 属性

typeName 属性用来明确指定将要删除的要素实例的类型。typeName 参数值应是服务能力描述文档（见 8.3）中列出的其中一种要素类型。

15.2.7.2.2 Filter 表达式

删除操作的作用范围由 ISO 19143:2010 第 7 章中描述的 fes:Filter 元素限定。如果 fes:Filter 元素没有标志任何要删除的要素实例，那么删除操作不起作用。这不算一个异常情况。

15.2.8 Native 操作

显然，一个开放的接口只能支持一定的公共能力集。wfs:Native 元素的目的是允许访问提供商提供的任何特殊 WFS 或数据存储的特定能力。

以下 XML 模式片段定义了 wfs:Native 元素：

```
<xsd:element name="Native" type="wfs:NativeType"
        substitutionGroup="wfs:AbstractTransactionAction"/>
<xsd:complexType name="NativeType" mixed="true">
  <xsd:complexContent>
    <xsd:extension base="wfs:AbstractTransactionActionType">
      <xsd:sequence>
        <xsd:any processContents="lax" namespace="##other" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="vendorId" type="xsd:string" use="required"/>
      <xsd:attribute name="safeToIgnore" type="xsd:boolean" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

wfs:Native 元素仅包含提供商提供的特殊命令或操作。

vendorId（提供商标识符）属性用来识别提供商，其提供了包含在 wfs:Native 元素中的命令和操作。通过这个属性，允许 WFS 决定它是否可以处理这个命令。有效的 vendorId 值应在服务的能力描述文档（见 8.3）中通过参数约束“vendorId”声明。

当 wfs:Native 中的命令或操作不被识别时，safeToIgnore（安全忽略）属性用来指导 WFS 采取行动。safeToIgnore 属性有两个可能值：True 或者 False。它们含义如下：

safeToIgnore=False

值 *False* 表明 wfs:Native 元素不能忽略，并且如果 WFS 不能处理操作，wfs:Native 元素中关联的操作将会失败。

safeToIgnore=True

True 值表示 wfs:Native 元素可以安全地忽略。

示例：这个示例举例说明了使用 wfs:Native 元素使一个基于 SQL 关系数据库中的特殊要素成为可能。在这个实例中的元素表明这是一个 Oracle 命令，并且该命令可以安全地忽略。

```
<Native vendorId = "Oracle" safeToIgnore = "True">
ALTER SESSION ENABLE PARALLEL DML
</Native>
```

15.3 响应

15.3.1 响应语义

在响应 Transaction 请求时，WFS 应生成指明事务最终状态的 XML 文档。另外，如果一个 Transaction 请求包括 wfs:Insert 元素，那么 WFS 响应应报告所有新创建的要素标识符。

图 31 描述了 Transaction 操作的响应。

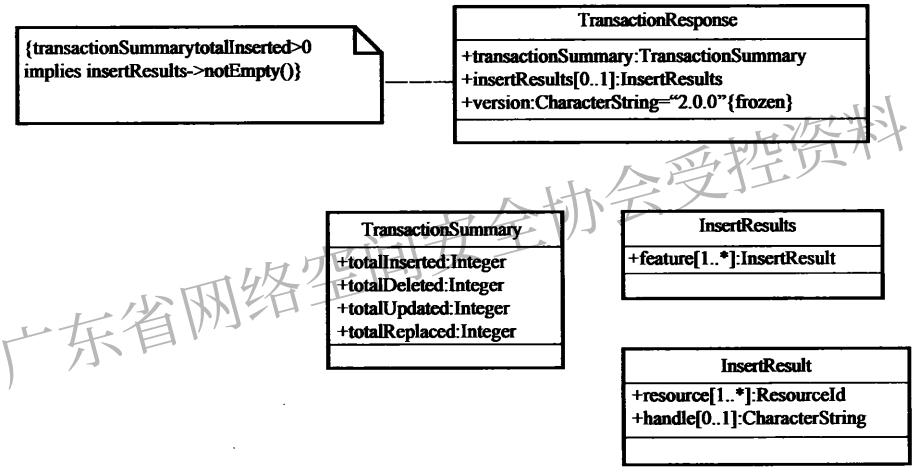

图 31 Transaction 响应

15.3.2 TransactionResponse(事务响应)元素

事务请求响应的根元素称为 wfs:TransactionResponse。

以下 XML 模式片段声明了 wfs:TransactionResponse 元素：

```
<xsd:element name = "TransactionResponse"
        type = "wfs:TransactionResponseType"/>
<xsd:complexType name = "TransactionResponseType">
  <xsd:sequence>
    <xsd:element name = "TransactionSummary"
            type = "wfs:TransactionSummaryType"/>
    <xsd:element name = "InsertResults"
            type = "wfs:ActionResultsType" minOccurs = "0"/>
    <xsd:element name = "UpdateResults"
            type = "wfs:ActionResultsType" minOccurs = "0"/>
```

```
    <xsd:element name = "ReplaceResults"
                 type = "wfs:ActionResultsType" minOccurs = "0"/>
  </xsd:sequence>
  <xsd:attribute name = "version" type = "xsd:string" use = "required" fixed = "2.0.0"/>
</xsd:complexType>
```

wfs:TransactionResponse 元素包括事务所执行操作的一个总结，并且可以选择性地包括事务所创建的任意新要素或者新要素版本的一系列标识符。

15.3.3 TransactionSummary(事务总结)元素

以下 XML 模式片段定义了 wfs:TransactionSummary 元素：

```
<xsd:complexType name = "TransactionSummaryType">
  <xsd:sequence>
    <xsd:element name = "totalInserted" type = "xsd:nonNegativeInteger" minOccurs = "
0"/>
    <xsd:element name = "totalUpdated" type = "xsd:nonNegativeInteger" minOccurs = "
0"/>
    <xsd:element name = "totalReplaced" type = "xsd:nonNegativeInteger" minOccurs = "
0"/>
    <xsd:element name = "totalDeleted" type = "xsd:nonNegativeInteger" minOccurs = "
0"/>
  </xsd:sequence>
</xsd:complexType>
```

wfs:TransactionSummary 元素包括事务请求所创建的[例如 wfs:totalInserted(所有被插入的)]、修改的[例如 wfs:totalUpdated(所有被更新的)]、替换的(例如 wfs:totalReplaced)(所有被替换的)或者删除的[例如 wfs:totalDeleted(所有被删除的)]一系列要素的总数。

如果事务不包含 Insert 操作(见 15.2.4)，那么省略 wfs:totalInserted 元素。

如果事务不包含 Update 操作(见 15.2.5)，那么省略 wfs:totalUpdate 元素。

如果事务不包含 Replace 操作(见 15.2.6)，那么省略 wfs:totalReplaced 元素。

如果事务不包含 Delete 操作(见 15.2.7)，那么省略 wfs:totalDeleted 元素。

当生成此总结时，只有列在 WFS 的能力描述文档中的要素类型被计数。

15.3.4 InsertResults(插入结果)元素

以下 XML 模式片段声明了 wfs:ActionResultsType 类型，此类型为 wfs:InsertResults(插入结果)、wfs:UpdateResults(更新结果)和 wfs:RelaceResults(替换结果)元素的类型。

```
<xsd:complexType name = "ActionResultsType">
  <xsd:sequence>
    <xsd:element name = "Feature"
                 type = "wfs:CreatedOrModifiedFeatureType"
                 maxOccurs = "unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name = "CreatedOrModifiedFeatureType">
  <xsd:sequence maxOccurs = "unbounded">
    <xsd:element ref = "fes:ResourceId"/>
```

```
    </xsd:sequence>
    <xsd:attribute name="handle" type="xsd:string"/>
  </xsd:complexType>
```

如果一个事务请求中包含插入、更新和(或)替换操作,那么在事务响应中应指定 wfs:InsertResults、wfs:UpdateResults 和(或)wfs:ReplaceResults 元素。

wfs:InsertResults 元素包含一个或多个 wfs:Feature 元素,此元素指明新创建的要素实例的要素标识符,每个新创建的要素实例对应一个 wfs:Feature 元素。

另外,wfs:Feature 元素应以 wfs:Transaction 元素中插入操作的顺序列出来。

而且,wfs:Feature 元素可以与使用 handle 属性的插入操作相关联。如果一个值指定为创建要素的 wfs:Insert 元素中的 handle 属性值,那么同一个 handle 属性值可以指定为 wfs:Feature 元素中的 handle 属性值,用于表明具体的插入操作产生的具体要素实例。

15.3.5 UpdateResults(更新结果)元素

如果一个事务请求包含了更新操作,并服务器支持版本化,那么在事务响应中指定 wfs:UpdateResults 元素,以报告更新了哪些要素及其相应的新标识符。

wfs:UpdateResults 元素包含一个或者多个 wfs:Feature 元素,它们指明了被更新要素实例的新标识符。一个 wfs:Feature 元素应对应一个更新的要素实例,每个 wfs:Feature 元素应包含一个带有属性 rid 和属性 oldRid 的 fes:ResourcedId 元素(见 ISO 19143:2010,7.11),其中,属性 rid 的值为新版本要素的标识符,而属性 oldRid 的值为老版本要素的标识符。

另外,wfs:Feature 元素应以 wfs:Transaction 元素中更新操作的顺序列出来。

而且,wfs:Feature 元素可以与使用 handle 属性的更新操作相关联。如果一个值指定为更新要素的 wfs:Update 元素中的 handle 属性值,那么同样的 handle 值可以指定为 wfs:Feature 元素中的 handle 属性值,用于表明具体的更新操作生成的具体要素实例。

15.3.6 ReplaceResults(替换结果)元素

如果一个事务请求包含了替换操作,并服务器支持版本化,那么在事务响应中指定 wfs:ReplaceResults 元素,以报告替换了哪些要素及其相应的新标识符。

wfs:ReplaceResults 元素包含一个或者多个 wfs:Feature 元素,它们指明了被替换的要素实例的新标识符。一个 wfs:Feature 元素应对应一个更新的要素实例,每个 wfs:Feature 元素应包含一个带有属性 rid 和属性 oldRid 的 fes:ResourcedId 元素(见 ISO 19143:2010,7.11),其中,属性 rid 的值为新版本要素的标识符,而属性 oldRid 的值为老版本要素的标识符。

另外,wfs:Feature 元素应以 wfs:Transaction 元素中替换操作的顺序列出来。

而且,wfs:Feature 元素可以与使用 handle 属性的替换操作相关联。如果一个值指定为替换要素的 wfs:Replace 元素中 handle 属性值,那么同样的 handle 值可以指定为 wfs:Feature 元素中的 handle 属性值,用于表明具体的替换操作生成的具体要素实例。

15.4 异常

当 WFS 在解析一个 Transaction 请求时遇到错误,将抛出一个如 7.5 所描述的 OperationParsingFailed 异常。

当 WFS 在处理某一个包含在 Transaction 请求中的操作时遇到错误,将抛出一个如 7.5 所描述的 OperationProcessingFailed 异常。

附 录 A
（规范性附录）
一致性测试

A.1 一致性类

A.1.1 简单 WFS

a) 测试目的：验证服务器实现了简单 WFS 一致性类。
b) 测试方法：向服务器提交请求，然后验证下面几点：服务器生成的能力描述文档包括 GetCapabilities、DescribeFeatureType、ListStoredQueries、DescribeStoredQueries 操作和带有存储的查询 GetFeatureById 的 GetFeature 操作；验证 DescribeFeatureType 请求的结果总是一个完整的应用模式；验证 A.2.22.4；验证服务器至少支持一个服务绑定（见附件 D）；验证下面一系列一致性测试：A.2.2、A.2.3、A.2.4、A.2.5、A.2.6.1 和（或）A.2.6.2、A.2.7.1 和（或）A.2.7.2、A.2.8.1、A.2.9、A.2.14、A.2.15、A.2.16、A.2.17、A.2.19.1、A.2.19.2.1、A.2.21、A.2.22.4。验证一致性类：ISO 19143:2010、A.1.1。验证下面一致性测试：GB/T 23708—2009、A.1.1、A.1.4、A.1.5、A.1.7、B.3、B.5 and B.2.3。
c) 引用：7.9.3.6、第 8 章、第 9 章、第 10 章、第 11 章（只引用 wfs:StoredQuery 操作）、14.3、14.4。
d) 测试类型：能力。

A.1.2 基础 WFS

a) 测试目的：验证服务器实现了基础 WFS 一致性类。
b) 测试方法：验证服务器实现了简单 WFS 一致性类。验证服务器生成的能力描述文档包括 GetPropertyValue 和 gGetFeature 操作。验证包含存储的查询的 GetPropertyValue 操作和 GetFeature 操作。验证服务器至少实现了 ISO 19143 的最小空间过滤器一致性类。验证下面一系列一致性测试：A.2.2、A.2.7、A.2.8.1、A.2.11.2、A.2.12、A.2.13、A.2.19、A.2.20.1、A.2.20.2、A.2.22。对于测试 A.2.23，验证 ImplementsBasicWFS 约束值设置为“TRUE”。验证一致性测试：ISO 19143:2010、A.1.2、A.1、4、A.1.5、A.1.6、A.1.7、A.1.12、A.1.14。验证一致性测试：GB/T 23708—2009，B.4。
c) 引用：第 10 章、第 11 章（wfs:Query 和 wfs:StoredQuery 操作）。
d) 测试类型：能力。

A.1.3 事务 WFS

a) 测试目的：验证服务器实现了事务 WFS 一致性类。
b) 测试方法：验证服务器实现了基础 WFS 一致性类。验证服务器生成的能力描述文档包括事务操作。验证事务操作。验证下面一系列一致性测试：A.2.2、A.2.8.2、A.2.10、A.2.11.1、A.2.18。对于测试 A.2.23，验证 ImplementsTransactionalWFS 约束值设置为“TRUE”。
c) 引用：第 15 章。
d) 测试类型：能力。

A.1.4 锁定 WFS

a) 测试目的：验证服务器实现了锁定 WFS 一致性类。

b) 测试方法:验证服务器实现了基础 WFS 一致性类。验证服务器生成的能力描述文档包括 LockFeature 或 GetFeatureWithLock 操作,或者同时包含这两个操作。验证列举的操作集中的操作。验证下面一系列一致性测试:A.1.3。对于测试 A.2.23,验证 ImplementsLockingWFS 约束值设置为“TRUE”。

c) 引用:第 12 章、第 13 章。

d) 测试类型:能力。

A.1.5 HTTP GET

a) 测试目的:验证服务器实现了 HTTP GET 一致性类。

b) 测试方法:验证服务器实现了所有操作的 KVP 编码,这些操作为在能力描述文档中 OperationsMetadata 部分列举了的并且在本标准中定义了 KVP 编码的所有操作。验证下面一系列一致性测试:A.2.6.1、A.2.7.1、A.2.13.2。对于测试 A.2.23,验证 KVPEncoding 约束值设置为“TRUE”。

c) 引用:附录 D。

d) 测试类型:能力。

A.1.6 HTTP POST

a) 测试目的:验证服务器实现了 HTTP POST 一致性类。

b) 测试方法:验证服务器实现了所有操作的 XML 编码,这些操作为在能力描述文档中 OperationsMetadata 部分列举了的所有操作。验证下面一系列一致性测试:A.2.6.2、A.2.7.2、A.2.13.1。对于测试 A.2.23,验证 XMLEncoding 约束值设置为“TRUE”。

c) 引用:附录 D。

d) 测试类型:能力。

A.1.7 SOAP

a) 测试目的:验证服务器实现了 SOAP 一致性类。

b) 测试方法:验证服务器实现了所有操作的 XML 编码,这些操作为在能力描述文档中 OperationsMetadata 部分列举了的所有操作。验证服务器支持 SOAP 服务绑定。验证下面一系列一致性测试:A.1.2。对于测试 A.2.23,验证 SOAPEncoding 约束值设置为“TRUE”。

c) 引用:附录 D。

d) 测试类型:能力。

A.1.8 继承

a) 测试目的:验证服务器实现了 Inheritance 一致性类。

b) 测试方法:验证服务器实现了 schema-element() XPath 功能。对于测试 A.2.23,验证 ImplementsInhertance 约束值设置为“TRUE”。验证一致性测试:ISO 19143:2010,A.15。

c) 引用:7.9.2.4.2、A.2.22.1.2。

d) 测试类型:能力。

A.1.9 远程解析

a) 测试目的:验证服务器能够解析远程资源引用。

b) 测试方法:见 A.2.17.2.3。对于测试 A.2.23,验证 ImplementsRemoteResolve 约束值设置为“TRUE”。验证一致性测试:GB/T 23708—2009,B.2.1。

c) 引用:7.6.4。

d) 测试类型:能力。

A.1.10 分页响应

a) 测试目的:验证服务器实现了分页响应。

b) 测试方法:见 A.2.20。对于测试 A.2.23,验证 ImplementsResultPaging 约束值设置为“TRUE”。验证一致性测试:GB/T 23708—2009,B.3。

c) 引用:7.7.4.4。

d) 测试类型:能力。

A.1.11 标准连接

a) 测试目的:验证服务器实现了标准连接。

b) 测试方法:见 A.2.22.2.1。对于测试 A.2.23,验证 ImplementsStandardJoins 约束值设置为“TRUE”。验证一致性测试:ISO 19143:2010,A.8 和 A.9。

c) 引用:7.9.2.5.3.1。

d) 测试类型:能力。

A.1.12 空间连接

a) 测试目的:验证服务器实现了空间连接。

b) 测试方法:见 A.2.22.2.2。对于测试 A.2.23,验证 ImplementsSpatialJoins 约束值设置为“TRUE”。验证一致性测试:ISO 19143:2010,A.1.7、A.1.8。

c) 引用:7.9.2.5.3.1。

d) 测试类型:能力。

A.1.13 时间连接

a) 测试目的:验证服务器实现了时间连接。

b) 测试方法:见 A.2.22.2.3。对于测试 A.2.23,验证 ImplementsTemporalJoins 约束值设置为“TRUE”。验证一致性测试:ISO 19143:2010,A.1.9、A.1.10。

c) 引用:7.9.2.5.3.1。

d) 测试类型:能力。

A.1.14 要素版本

a) 测试目的:验证服务器实现了要素版本一致性类。

b) 测试方法:见 A.2.11。对于测试 A.2.23,验证 ImplementsFeatureVersioning 约束值设置为“TRUE”。验证一致性测试:ISO 19143:2010,A.11。

c) 引用:ISO 19143 :2010, 7.11。

d) 测试类型:能力。

A.1.15 管理存储的查询

a) 测试目的:验证服务器实现了管理存储的查询一致性类。

b) 测试方法:验证服务器实现了基础 WFS 一致性类。验证在服务器生成的能力描述文档中包含 CreateStoredQuery 和 DropStoredQuery 操作。向服务器提交请求,以验证 CreateStoredQuery 和 DropStoredQuery 操作。对于测试 A.2.23,验证 ManageStoredQueries 约束

值设置为“TRUE”。验证一致性测试:ISO 19143:2010,A.1。

c) 引用:14.4、14.5。

d) 测试类型:能力。

A.2 基本测试

A.2.1 版本协商

a) 测试目的:验证服务器能正确处理版本协商并且支持指定的版本“2.0.0”。

b) 测试方法:发送一个 version 参数设置为“2.0.0”的 GetCapabilities 请求,验证返回的响应为一个有效的如本标准所描述的能力描述文档(见 8.3)。

c) 引用:6.2.1, OGC 06-121r3:2009 的一致性测试 A.4.2.3。

d) 测试类型:基本类型。

A.2.2 列举版本号“2.0.0”作为支持的请求版本号

a) 测试目的:验证服务器在能力描述文档中列举版本号“2.0.0”作为支持的请求版本号。

b) 测试方法:执行一个 GetCapabilities 请求,验证版本号“2.0.0”作为 ows:ServiceTypeVersion(服务类型版本号)元素内容的其中一项。

c) 引用:6.2.2。

d) 测试类型:基本类型。

A.2.3 无效版本号

a) 测试目的:验证一个非 GetCapabilities 请求,如果其版本号不在服务器的能力描述文档中声明的版本号之列时,该请求失败。

b) 测试方法:查看 GetCapabilities 请求的响应文档,确定服务器声明支持的请求版本。执行一个或者多个 WFS 请求,其版本号不在支持的版本号之列,然后验证服务器会生成一个 InvalidParameterValue 异常。

c) 引用:6.2.2。

d) 测试类型:基本类型。

A.2.4 GetCapabilities 请求的版本协商

a) 测试目的:验证服务器针对 GetCapabilities 操作能够正确处理版本协商。

b) 测试方法:验证服务器遵循在 OGC 06-121r3:2009 中描述的测试。

c) 引用:6.2.3,OGC 06-121r3:2009 中的 A.4.2.3。

d) 测试类型:基本类型。

A.2.5 响应 XML 编码和 KVP 编码的请求

a) 测试目的:验证不管是哪一种编码方式,服务器响应一个请求的结果是一样的。

b) 测试方法:验证服务器同时支持 XML 编码和 KVP 编码的请求。选取一组请求,对请求进行 XML 编码和 KVP 编码,然后验证对应的 XML 编码的请求和 KVP 编码的请求的响应是一样的。

c) 引用:6.2.4。

d) 测试类型:基本类型。

A.2.6 参数排序和大小写

A.2.6.1 KVP 编码的请求

a) 测试目的:验证 KVP 编码的请求中参数的顺序和大小写不影响响应结果。
b) 测试方法:选取一组 KVP 编码的请求,并进行发送。每次使用不同顺序和混合大小写的方式指定请求中的参数。验证各种情况下服务器的响应不受影响。
c) 引用:6.2.5.2。
d) 测试类型:基本类型。

A.2.6.2 XML 编码的请求

参数排序和大小写测试不适合根据固定模式进行 XML 编码的请求。

A.2.7 无法识别的参数

A.2.7.1 KVP 编码的请求

a) 测试目的:验证服务器忽略 KVP 编码的请求中无法识别的参数。
b) 测试方法:生成一组有效的 KVP 编码的请求,并在请求中添加一个或者多个本标准未定义的参数。确保这些添加的参数不是服务能力描述文档中声明的提供商指定参数。验证服务器能够响应这些请求,并忽略添加的无法识别的参数。
c) 引用:6.2.5.2。
d) 测试类型:基本类型。

A.2.7.2 XML 编码的请求

无法识别的参数的测试不适合根据固定模式进行 XML 编码的请求。

A.2.8 服务器针对 GML 要素的操作

A.2.8.1 服务器生成 GML 要素

a) 测试目的:验证服务器能生成有效的 GML 要素。
b) 测试方法:选取一组其响应包含要素的请求(GetFeature,GetFeatureWithLock 或 GetPropertyValue)。设置 outputFormat 参数为“application/gml+xml; version=3.2”。验证这些请求的响应对于服务器声明支持的应用模式是有效的,并证明生成的要素是有效的 GML 要素。
c) 引用:7.1。
d) 测试类型:基本类型。

A.2.8.2 服务器获取 GML 要素

a) 测试目的:验证服务器能获取 GML 编码的要素。
b) 测试方法:生成一个创建新要素的事务,此事务的 inputFormat 参数设置为“application/gml+xml; version=3.2”。验证该操作能够成功执行。
c) 引用:7.1。
d) 测试类型:基本类型。

A.2.9 要素标识符

a) 测试目的:验证服务器分配唯一的永久的标识符。

b) 测试方法:创建一个新的要素实例,然后验证在事务响应中给该要素分配了一个标识符。使用该标识符在服务器中检索这个要素,验证响应中只返回了一个与查询的标识符相对应的要素。验证在能力描述文档中使用 gml:id 属性对这个标识符进行编码。

c) 引用:7.2.1、7.2.2 。

d) 测试类型:基本类型。

A.2.10 不会变化的标识符

a) 测试目的:验证 WFS 操作中要素标识符不会发生变化。

b) 测试方法:更新一些要素,验证要素标识符没有因为更新而发生变化。删除一些要素,然后又创建一个或多个要素,验证删除的要素的标识符没有重新用在新创建的要素上。

c) 引用:7.2.1。

d) 测试类型:基本类型。

A.2.11 版本化

A.2.11.1 版本创建

a) 测试目的:验证服务器在创建或修改要素的时候创建新的要素版本。

b) 测试方法:创建一个新的要素。多修改该要素以强制将其创建为多个版本。通过 A.2.11.2 测试验证新要素版本已被创建。

c) 引用:7.2.3。

d) 测试类型:基本类型。

A.2.11.2 版本搜寻

a) 测试目的:验证服务器能够维护要素的版本信息,并当执行一个查询时能够使用版本信息搜寻要素版本。

b) 测试方法:使用谓词选择一个或者多个要素版本时,这些谓词使用版本搜寻控制(见 ISO 19143 :2010, 7.11)。

c) 引用:7.2.3。

d) 测试类型:基本类型。

A.2.12 XPath 子集

a) 测试目的:验证服务器支持请求的 XPath 子集。

b) 测试方法:见 ISO 19143 :2010 的测试 A.14。

c) 引用:7.3.1、7.3.2。

d) 测试类型:基本类型。

A.2.13 谓词编码

A.2.13.1 XML 编码的请求

a) 测试目的:验证服务器能执行 XML 编码的操作,该操作带有用 fes:Filter 编码的谓词。

b) 测试方法:选取一个使用谓词的操作。使用 fes:Filter 元素编码操作中的谓词,然后验证操作能成功执行。

c) 引用:7.4。

d) 测试类型:基本类型。

A.2.13.2 KVP 编码的请求

a) 测试目的:验证服务器能执行 KVP 编码的操作,该操作带有用 KVP 参数编码的谓词。
b) 测试方法:选取一个使用谓词的操作,使用 KVP 参数编码几个操作中的谓词,并执行操作。
c) 引用:7.4。
d) 测试类型:基本类型。

A.2.14 异常报告

A.2.14.1 异常报告的有效性

a) 测试目的:依据在 OGC 06-1212r3 的第 8 章中定义的模式,验证服务器生成的异常报告是有效的。
b) 测试方法:设计并执行一个会产生错误的请求,验证服务器生成的异常报告是有效的。
c) 引用:7.5。
d) 测试类型:基本类型。

A.2.14.2 异常报告的适宜性

a) 测试目的: 通过给代码的值和定位参数设置合适的值,验证服务器能够生成一个恰当的异常报告。
b) 测试方法:设计一系列的能够产生 OGC 06121r3 中表 25 和表 3 中每一个错误编码对应的错误的请求,通过检验代码和定位参数都设置了正确的值来验证服务器能够针对每种情况产生合适的异常报告 。
c) 引用:7.5。
d) 测试类型:基本类型。

A.2.14.3 异常报告的版本

a) 测试目的:验证服务器生成的异常报告的 version 参数设置为“2.0.0”。
b) 测试方法:设计并执行一个会生成异常的请求,然后验证服务器生成的异常报告的 version 参数设置为“2.0.0”。
c) 引用:7.5。
d) 测试类型:基本类型。

A.2.15 公共请求参数

A.2.15.1 service 和 version 参数

a) 测试目的:验证服务器能够正确处理 service 和 version 两个参数。
b) 测试方法:设计一组缺少 service 和 version 参数的请求。验证服务器响应的结果中包含一个 MissingParameterValue 异常,并且定位参数命名为 missing 参数。
c) 引用:7.6.2.4、7.6.2.5。
d) 测试类型:基本类型。

A.2.15.2 handle 参数

a) 测试目的:验证服务器能够在异常中正确报告 handle 参数值。

b) 测试方法:设计一组包含多个操作(例如 GetFeature 或 Transaction)的请求,每个操作中包含 handle 参数。设计其中一个操作失败,然后验证在异常报告中使用 handle 属性作为定位参数的值,以指明哪一个参数失败了。

c) 引用:7.6.2.6。

d) 测试类型:基本类型。

A.2.16 Standard presentation 参数

A.2.16.1 startIndex 参数

a) 测试目的:验证服务器能够正确处理 startIndex 参数。

b) 测试方法:设计并执行一个不含 startIndex 参数的 GetFeature 请求,然后记录响应文档。执行同样的 GetFeature 请求,但这一次包含 startIndex 参数。验证响应文档中的起始要素的索引为 startIndex 参数值。

c) 引用:7.6.3.4。

d) 测试类型:基本类型。

A.2.16.2 count 参数

A.2.16.2.1 处理

a) 测试目的:验证服务器能够正确处理 count 参数。

b) 测试方法:设计并执行一个查询请求,然后记录响应文档中的要素个数。执行同样的请求,但这一次请求包含的 count 参数值小于响应中要素的数量。验证响应只包含数量为 count 值的记录。

c) 引用:7.6.3.5。

d) 测试类型:基本类型。

A.2.16.2.2 默认配置

a) 测试目的:验证服务器能够正确地处理为 count 参数配置好的默认值。

b) 测试方法:生成一个能力描述文档,并确定 count 参数的默认值。执行一个会生成数量多于该默认值要素的查询,然后验证响应文档只包含数量为 count 参数配置的默认值的要素。

c) 引用:7.6.3.5。

d) 测试类型:基本类型。

A.2.16.3 resultType 参数

a) 测试目的:验证服务器能够正确处理 resultType 参数。

b) 测试方法:设计并执行一个请求,此请求中 resultType 参数设置为“results”。验证服务器生成一个要素集,其内容为要素。执行同样的一个请求,但是 resultType 参数设置为“hits”。验证服务器生成一个空要素集,其响应中 numberReturned 参数设置为 0。在 resultType 参数设置为“results”的情况下,相应文档中的 numberMatched 参数值设置为查询请求返回的要素个数。

c) 引用:7.6.3.6。

d) 测试类型:基本类型。

A.2.16.4 outputFormat 参数

a) 测试目的:验证服务器实现必选的 outputFormat 值。
b) 测试方法:执行一个 outputFormat 参数值设置为“application/gml+xml; version=3.2”的请求,然后验证输出结果是有效的 GML3.2 应用模式还是无效的 GML3.2 应用模式。
c) 引用:7.6.3.7。
d) 测试类型:基本类型。

A.2.17 Standard resolve 参数

A.2.17.1 声明支持远程资源解析

a) 测试目的:验证服务器声明了具备远程解析的能力。
b) 测试方法:生成一个能力描述文档,验证由一个 ows:Parameter 元素声明服务器支持 resolve 参数值域的哪一个值(即 none,local,remote,all)。
c) 引用:7.6.4.4。
d) 测试类型:基本类型。

A.2.17.2 resolve 参数处理

A.2.17.2.1 无资源解析

a) 测试目的:验证如果 resolve 参数值设置为“none”,则在响应中没有资源解析。
b) 测试方法:设计一个包含本地资源引用的数据集。执行一个请求,此请求中 resolve 参数值设置为“none”,然后验证在响应中没有解析资源引用。
c) 引用:7.6.4.4。
d) 测试类型:基本类型。

A.2.17.2.2 本地资源解析

a) 测试目的:验证如果 resolve 参数值设置为“local”,则在响应中解析本地资源。
b) 测试方法:设计一个包含本地资源引用的数据集。执行一个请求,此请求中 resolve 参数值设置为“local”,然后验证在响应中解析了本地资源引用。
c) 引用:7.6.4.4。
d) 测试类型:基本类型。

A.2.17.2.3 远程资源解析

A.2.17.2.3.1 声明解析远程引用的能力

a) 测试目的:验证服务器声明了解析远程资源的能力。
b) 测试方法:生成一个能力描述文档,然后验证 ImplementsRemoteResolve 服务约束设置为“true”。
c) 引用:7.6.4.4。
d) 测试类型:基本类型。

A.2.17.2.3.2 远程资源解析

a) 测试目的:验证如果 resolve 参数值设置为“remote”,则在响应中解析的为远程资源。

b) 测试方法:设计一个包含远程资源引用的数据集。执行一个请求,此请求中 resolve 参数值设置为“remote”,然后验证在响应中是否解析了远程资源引用。

c) 引用:7.6.4.4。

d) 测试类型:基本类型。

A.2.17.2.3.3 所有资源引用解析

a) 测试目的:验证如果 resolve 参数值设置为“all”,则响应中解析的为所有资源引用。

b) 测试方法:设计一个包含本地和远程资源引用的数据集。执行一个请求,此请求中 resolve 参数值设置为“all”,然后验证在响应中是否解析了所有资源引用。

c) 引用:7.6.4.4。

d) 测试类型:基本类型。

A.2.17.3 解析深度处理

A.2.17.3.1 本地资源

a) 测试目的:验证当执行资源解析时,对本地资源引用的解析深度为 resolveDepth 参数指定的深度值。

b) 测试方法:设计一个数据集,此数据集中包含引用本地资源的要素,这些本地资源依次引用其他本地资源至不同级别的递归程度,同时此数据集也包含带有循环引用链的要素集。执行一组 resolveDepth 参数设置为不同值的查询,然后验证响应中资源解析是否执行到指定的深度。确保 resolveDepth 参数值“0”,“1”和“*”都测试过,以验证在响应中,只有暂时引用的本地资源和所有引用的本地资源被解析。同时验证循环引用链被正确解析了。

c) 引用:7.6.4.5。

d) 测试类型:基本类型。

A.2.17.3.2 远程资源

a) 测试目的:验证当执行资源解析时,对远程资源引用的解析深度为 resolveDepth 参数指定的深度值。

b) 测试方法:设计一个数据集,此数据集中包含引用远程资源的要素,这些远程资源依次引用其他远程资源至不同级别的递归程度,同时此数据集也包含带有循环引用链的要素集。执行一组 resolveDepth 参数设置为不同值的查询,然后验证响应中资源解析是否执行到指定的深度。确保 resolveDepth 参数值“0”,“1”和“*”都测试过,以验证在响应中,只有暂时引用的远程资源和所有引用的远程资源被解析。同时验证循环引用链被正确解析了。

c) 引用:7.6.4.5。

d) 测试类型:基本类型。

A.2.17.3.3 所有资源

a) 测试目的:验证当实现资源解析时,对所有资源引用的解析深度为 resolveDepth 参数指定的深度值。

b) 测试方法:设计一个数据集,此数据集中同时包含引用本地和远程资源的要素,这些资源依次引用其他本地或远程资源至不同级别的递归程度,同时此数据集也包含带有循环引用链的要素集。实现一组 resolveDepth 参数设置为不同值的查询,然后验证响应中资源解析是否执行到指定的深度。确保 resolveDepth 参数值“0”,“1”和“*”都测试过,以验证在响应中,只有暂

时引用的资源和所有引用的资源被解析。同时验证循环引用链正确被解析了。

c) 引用:7.6.4.5。

d) 测试类型:基本类型。

A.2.17.3.4 默认配置

a) 测试目的:验证如果对于本地和远程的资源解析,resolveDepth 参数值配置了默认值,服务器能够正确处理。

b) 测试方法:生成一个能力描述文档,并且获取 ResolveLocalScope 和 ResolveRemoteScope 操作的约束值。如果配置了 ResolveLocalScope 约束值,那么重新测试 A.2.17.3.1,其中没有设置 resolveDepth 参数,就可以使用默认值。如果配置了 ResolveRemoteScope 约束值,那么重新测试 A.2.17.3.2,其中没有设置 resolveDepth 参数,就可以使用默认值。如果同时配置了两个约束值,那么重新测试 A.2.17.3.1,A.2.17.3.2 和 A.2.17.3.3,其中没有设置 resolveDepth 参数,就可以使用默认值。

c) 引用:7.6.4.5。

d) 测试类型:基本类型。

A.2.17.4 解析暂停处理

A.2.17.4.1 处理

a) 测试目的:验证当解析暂停阶段过去之后,服务器停止试图对资源的解析。

b) 测试方法:设计只包含一个要素的数据集,此要素引用一个不存在的资源。实现一个设置了 resolve 参数和 resolveTimeout 参数的查询请求,然后验证服务器在解析暂停阶段过去之后,服务器能响应。

c) 引用:7.6.4.6。

d) 测试类型:基本类型。

A.2.17.4.2 默认配置

a) 测试目的:验证包含配置过的 resolveTimeout 参数时,服务器有何响应。

b) 测试方法:生成一个能力描述文档,通过查看 ResolveTimeoutDefault 约束来确定是否配置默认的 timeout 值。如果配置了默认的 timeout 值,那么重新测试 A.2.17.4.1,其中没有设置 ResolveTimeoutDefault 参数,就可以使用配置的值。

c) 引用:7.6.4.6。

d) 测试类型:基本类型。

A.2.17.5 不能解析资源引用

a) 测试目的:验证如果服务器不能解析资源,那么它在响应中报告不能解析的原始 URI。

b) 测试方法:设计一个只包含一个要素的数据集,此要素引用一个不存在的资源。实现一个包含 resolve 参数的查询请求,然后验证在响应中,引用为不能解析的原始 URI。

c) 引用:7.6.4.7。

d) 测试类型:基本类型。

A.2.18 标准输入参数

A.2.18.1 inputFormat 参数

a) 测试目的:验证当 inputFormat 参数值是“application/gml+xml; version=3.2”时,服务器能

够正确处理 inputFormat 参数。

b) 测试方法:创建一个带有插入操作的事务,inputFormat 设置为“application/gml+xml;version=3.2”,并且操作的内容为一个无效的要素。这个服务器能够检测到这个无效的要素编码并抛出一个异常。

c) 引用:7.6.5.4。

d) 测试类型:基本类型。

A.2.18.2 srsName 参数

a) 测试目的:验证在输入时服务器能够正确处理 srsName 参数。

b) 测试方法:插入一个带有几何形状的要素,其几何形状的 srsName 值是服务器在能力描述文档中声明支持的一个 CRS 值。验证插入操作能成功完成。插入另一个带有几何形状的要素,其几何形状的 srsName 参数值不是服务器在能力描述文档中声明支持的一个 CRS 值。验证服务器返回一个异常。最后插入一个带有几何形状的要素,其几何形状的 srsName 参数值是服务器在能力描述文档中声明支持的一个 CRS 值,但是插入的几何形状编码使用的 CRS 与作为 srsName 参数值的 CRS 不匹配。验证服务器返回一个异常。

c) 引用:7.6.5.5。

d) 测试类型:基本类型。

A.2.19 标准响应参数

A.2.19.1 timeStamp 参数

a) 测试目的:验证服务器实现了 timeStamp 参数。

b) 测试方法:服务器生成一个要素集响应,然后验证实现了 timeStamp 参数,并赋予一个有效值 xsd:dateTime。

c) 引用:7.7.4.1。

d) 测试类型:基本类型。

A.2.19.2 numberMatched 参数

A.2.19.2.1 标准处理

a) 测试目的:验证服务器实现了 numberMatched 参数。

b) 测试方法:生成一个要素集响应,然后验证实现了 numberMatched 参数,并且赋予一个正整数值或者值“unavailable”。

c) 引用:7.7.4.2。

d) 测试类型:基本类型。

A.2.19.2.2 处理 resultType 参数

A.2.19.2.2.1 没有分页的响应

a) 测试目的:验证当将不支持响应分页的服务器的 resultType 参数设置为“hits”时,numberMatched 参数能够正确地实现。

b) 测试方法:生成一个请求的要素集响应,请求中将 resultType 参数设置为“hits”。验证响应是一个空的要素集,numberMatched 参数包含许多在响应中期望的要素。

c) 引用:7.7.4.2。

d) 测试类型:基本类型。

A.2.19.2.2.2 分页响应

a) 测试目的:验证当将支持响应分页的服务器的 resultType 参数设置为“hits”时,numberMatched 参数能够正确地实现。

b) 测试方法:生成一个请求的要素集响应,请求中将 resultType 参数设置为“hits”。验证响应是一个空的要素集,其 numberMatched 参数包含许多在响应中期望的要素。验证响应中下一个出现的参数和它的值是一个获取第一个结果集的 URI。

c) 引用:7.7.4.2。

d) 测试类型:基本类型。

A.2.20 响应分页

A.2.20.1 声明支持响应分页

a) 测试目的:验证服务器声明支持响应分页的能力。

b) 测试方法:生成一个能力描述文档,然后验证设置了 ImplementsResponsePaging 服务约束。

c) 引用:7.7.4.4.1。

d) 测试类型:基本类型。

A.2.20.2 处理

a) 测试目的:验证服务器在分页响应时能正确地运行。

b) 测试方法:实现一个选择一组要素的查询请求,查询中设置 count 参数为一个小的数值,以保证整个响应将以至少三个响应文档来显示。验证第一个响应文档包含一个 next 参数,其 next 参数包含了一个 URI,解析这个 URI 以检索响应的下一页。验证第二页包含 next 和 previous 参数,解析作为 next 参数值的 URI。验证响应的最后一页仅包含 previous 参数,解析这个作为 previous 参数值的 URI,然后验证响应为第二页。

c) 引用:7.7.4.4.1。

d) 测试类型:基本类型。

A.2.20.3 事务一致性

A.2.20.3.1 声明事务一致性

a) 测试目的:验证服务器声明了其响应分页是否支持事务一致性。

b) 测试方法:生成一个能力描述文档,然后验证包含了 PagingIsTransactionSafe 操作约束。

c) 引用:7.7.4.4.2.1。

d) 测试类型:基本类型。

A.2.20.3.2 分页响应是事务性安全的

a) 测试目的:验证分页响应是事务性安全的。

b) 测试方法:执行 A.2.20.2 描述的测试,除此之外,在获取下一页或前一页时,在另外一个环境下执行一个从结果集中添加或移除记录的事务操作。验证响应页中没有添加或移除记录。

c) 引用:7.7.4.4.2.2。

d) 测试类型:基本类型。

A.2.20.3.3 分页响应不是事务性安全的

a) 测试目的:验证分页响应不是事务性安全的。
b) 测试方法:执行 A.2.20.2 描述的测试,除此之外,在获取下一页或前一页时,在另外一个环境下执行一个从结果集添加或移除记录的操作。验证响应页中已添加或移除了记录。
c) 引用:7.7.4.4.2.3。
d) 测试类型:基本类型。

A.2.21 schemaLocation 参数

a) 测试目的:验证在要素集响应中使用 schemaLocation 属性来帮助验证响应的有效性。
b) 测试方法:生成一个要素集响应,然后验证实现了 schemaLocation 属性,且属性值包含一个模式引用,此模式引用可用来验证要素集中要素的有效性。
c) 引用:7.8。
d) 测试类型:基本类型。

A.2.22 查询表达式

A.2.22.1 即席查询表达式

A.2.22.1.1 typeNames 参数

a) 测试目的:验证 typeNames 参数的值域。
b) 测试方法:从服务能力描述文档中获取服务器提供的一系列要素类型。编码一个请求,此请求引用服务器没有提供的要素类型,然后验证响应为一个 InvalidParameterValue 异常报告。
c) 引用:7.9.2.4.1。
d) 测试类型:基本类型。

A.2.22.1.2 schema-element()函数

a) 测试目的:验证服务器在能力描述文档中声明支持 schema-element()函数。
b) 测试方法:生成一个能力描述文档,然后验证实现了 ImplementsInheritance 服务约束,并其值为“TRUE”。
c) 引用:7.9.2.4.2。
d) 测试类型:基本类型。

A.2.22.1.3 aliases 参数

a) 测试目的:验证服务器正确处理 aliases 参数。
b) 测试方法:编码一个包含 aliases 参数的查询请求,其别名的数量与 typeNames 参数中列举出的要素类型数量不相等。验证服务器将抛出一个 InvalidParameterValue 异常。
c) 引用:7.9.2.4.3。
d) 测试类型:基本类型。

A.2.22.1.4 srsName 参数

a) 测试目的:验证服务器正确处理 srsName 参数。
b) 测试方法:生成一个能力描述文档,并记下服务器支持的默认 CRS 和任何其他的 CRS。实现一个不包含 srsName 参数的查询,然后验证响应文档中的几何形状是使用默认 CRS 编码的。

选择一个或者多个支持的 CRS,然后实现查询,验证几何形状是使用正确的 CRS 编码的。选择一个 CRS,此 CRS 既不是默认的 CRS,也不是其他服务器声明支持的 CRS,然后实现查询,验证服务器生成一个 InvalidParameterValue 异常。

c) 引用:7.9.2.4.4。

d) 测试类型:基本类型。

A.2.22.1.5 映射子句

A.2.22.1.5.1 选择可选的特性

a) 测试目的:验证要素集响应中只包含映射子句指定的可选要素特性。

b) 测试方法:实现一个包含映射子句的查询,然后验证响应只包含映射子句选择的可选特性。

c) 引用:7.9.2.4.5.1。

d) 测试类型:基本类型。

A.2.22.1.5.2 无效特性名称

a) 测试目的:验证如果映射子句中指定了无效的特性名称,那么服务器生成一个异常。

b) 测试方法:实现一个请求,此请求中映射子句包含不存在的特性名称。然后验证服务器生成一个 InvalidParameterValue 异常。

c) 引用:7.9.2.4.6.1。

d) 测试类型:基本类型。

A.2.22.1.5.3 resolvePath 参数

a) 测试目的:验证 resolvePath 参数的行为。

b) 测试方法:实现一个解析资源引用的查询,并指定 resolvePath 参数一个合适的值。验证解析路径上的资源在响应中是否被解析了。验证本地资源的行为,如果服务器声明支持远程资源解析,那么验证远程资源的行为。

c) 引用:7.9.2.4.7。

d) 测试类型:基本类型。

A.2.22.2 选择子句

A.2.22.2.1 标准连接

A.2.22.2.1.1 声明支持标准连接

a) 测试目的:验证服务器正确声明和支持标准连接。

b) 测试方法:生成一个能力描述文档,然后验证 ImplementsStandardJoin 服务约束值设置为“TRUE”。

c) 引用:7.9.2.5.3.1。

d) 测试类型:基本类型。

A.2.22.2.1.2 处理

a) 测试目的:验证服务器正确处理标准连接。

b) 测试方法:实现一个请求,此请求从至少两类要素类型中获取数据,并且包含一个不牵涉空间或时间操作的连接谓词。验证通过使用 wfs:Tuple 元素包含每个要素元组,服务器生成一个

有效的响应。

c) 引用:7.9.2.5.3.1。

d) 测试类型:基本类型。

A.2.22.2.2 空间连接

A.2.22.2.2.1 声明支持空间连接

a) 测试目的:验证服务器正确声明和支持空间连接。

b) 测试方法:生成一个能力描述文档,然后验证 ImplementsSpatialJoin 服务约束值设置为“TRUE”。

c) 引用:7.9.2.5.3.1。

d) 测试类型:基本类型。

A.2.22.2.2.2 处理

a) 测试目的:验证服务器正确处理空间连接。

b) 测试方法:实现一个请求,此请求从至少两类要素类型中获取数据,并且包含一个使用空间操作的连接谓词。验证通过使用 wfs:Tuple 元素包含每个要素元组,服务器生成一个有效的响应。

c) 引用:7.9.2.5.3.1。

d) 测试类型:基本类型。

A.2.22.2.3 时间连接

A.2.22.2.3.1 声明支持空间连接

a) 测试目的:验证服务器正确声明和支持时间连接。

b) 测试方法:生成一个能力描述文档,然后验证 ImplementsTemporalJoin 服务约束值设置为“TRUE”。

c) 引用:7.9.2.5.3.1。

d) 测试类型:基本类型。

A.2.22.2.3.2 处理

a) 测试目的:验证服务器正确处理时间连接。

b) 测试方法:实现一个请求,此请求从至少两个要素类型中获取数据,并且包含一个使用时间操作的连接谓词。验证通过使用 wfs:Tuple 元素包含每个要素元组,服务器生成一个有效的响应。

c) 引用:7.9.2.5.3.1。

d) 测试类型:基本类型。

A.2.22.3 排序子句

A.2.22.3.1 基本排序

a) 测试目的:验证正确的排序行为。

b) 测试方法:实现一个包含排序子句的请求,然后验证响应中要素是以指定的特性或特性集按照指定的排序方式来进行排序的。

c) 引用:7.9.2.5.4.4。

d) 测试类型:基本类型。

A.2.22.3.2 默认排序

a) 测试目的:验证服务器按照默认排序方式返回要素。

b) 测试方法:对相同的一组要素重复地实现相同的查询,然后验证每一次要素都是以相同的顺序呈现。指定的排序与此不相关;相关的是,通过对同样的要素集进行同样的查询的所有调用,要素呈现的任何顺序都能被保持。

c) 引用:7.9.2.5.4.4。

d) 测试类型:基本类型。

A.2.22.3.3 排序级别

a) 测试目的:验证服务器遵守它在能力描述文档中声明的任何 SortLevelLimit。

b) 测试方法:实现一个带有排序子句的请求,排序子句中包含一个或者多个服务器在能力描述文档中使用 SortLevelLimit 约束声明支持的特性。验证服务器生成一个异常。

c) 引用:表 15。

d) 测试类型:基本类型。

A.2.22.4 存储的查询

A.2.22.4.1 存储的查询的操作

a) 测试目的:验证服务器实现了必选的存储的查询操作。

b) 测试方法:生成一个能力描述文档,然后验证服务器实现了 ListStoredQueries 和 DescribeStoredQueries 操作。使用 ListStoredQueries 操作验证响应中列举了一个带有标识符 urn:ogc:def:query:OGC-WFS:GetFeatureById 的可存储的查询。

c) 引用:7.9.3.6、8.2、14.3、14.4。

d) 测试类型:基本类型。

A.2.22.4.2 返回要素类型

a) 测试目的:验证存储的查询返回的所有要素类型为列举在要素类型清单中的要素类型。

b) 测试方法:生成一个能力描述文档,并获取服务器提供的要素类型清单。生成一组服务器提供的存储的查询,然后验证这些查询返回的要素类型都列举在要素类型清单中。

c) 引用:7.9.3、8.1、14.3。

d) 测试类型:基本类型。

A.2.23 声明一致性

a) 测试目的:验证服务器在能力描述文档中声明了它实现的能力。

b) 测试方法:生成一个能力描述文档,然后验证在表 13 列举的所有服务约束都在能力描述文档中呈现,并都包含值“TRUE”或“FALSE”,以指明服务器是否遵循一致性类。

c) 引用:表 13。

d) 测试类型:基本类型。

附 录 B
（资料性附录）
示 例

B.1 异常报告示例

下面是一个异常报告的示例。在这个示例中，一个 CreateStoredQuery 操作由于查询标识符重复而失败。

```
<? xml version = "1.0" ? >
< ExceptionReport
  version = "2.0.0"
  xmlns = "http://www.opengis.net/ows/1.1"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/ows/1.1
              http://schemas.opengis.net/ows/1.1.0/owsAll.xsd">
    < Exception exceptionCode = " DuplicateStoredQueryIdValue " locator = "
FeatureInPolygon">
      < ExceptionText > The identifier urn:SomeServer:StoredQueries:FeaturesInPolygon has
already been assigned to a stored query.</ExceptionText >
    </Exception >
</ExceptionReport >
```


B.2 DescribeFeatureType 示例

B.2.1 示例 1

假设在 SQL 数据库中定义了 TreesA_1M、RoadL_1M 和 LakesA_1M 地理要素类型，这些要素类型在数据库中有如下的描述：

```
SQL > describe TREESA_1M (描述 TREESA_1M)

Name(名称)          Null? (空)        Type (类型)
------------------  ------            ---------
EXTENT(范围)        NOT NULL(非空)    POLYGON (多边形)
ID                                    NUMBER(10) (数值型)
TREE_TYPE(树木类型)                   VARCHAR2(80) (字符型)
SQL > describe ROADL_1M (描述 ROADL_1M)
Name(名称)          Null? (空)        Type (类型)
------------------  ------            ---------
CENTERLINE(中心线)  NOT NULL          LINE (线型)
DESIGNATION (名称)                    VARCHAR2(30)
SURFACE_TYPE(表面物质类型)            VARCHAR2(30)
NLANES(车道数)                        NUMBER(2)
SQL > describe LAKESA_1M (描述 LAKESA_1M)
```

```
Name                  Null?      Type
--------------------  ---------  ---------
EXTENT                NOT NULL   POLYGON
NAME                             VARCHAR2(30)
AVG_DEPTH (平均深度)              NUMBER
MAX_DEPTH (最大深度)              NUMBER
```

DescribeFeatureType 请求的响应：

```
<? xml version = "1.0" ? >
< DescribeFeatureType
  service = "WFS"
  version = "2.0.0"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:myns = "http://www.myserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance" ISO/DIS 19142
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  < TypeName > myns:TreesA_1M </TypeName >
  < TypeName > myns:RoadL_1M </TypeName >
</DescribeFeatureType >
```

WFS 可能生成下面 XML 模式文档：

```
<? xml version = "1.0" ? >
< schema
  targetNamespace = "http://www.someserver.com/myns"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns = "http://www.w3.org/2001/XMLSchema"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  elementFormDefault = "qualified" version = "2.0.0">
  < import namespace = "http://www.opengis.net/gml/3.2"
      schemaLocation = "http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  <! -- = = = = = = = = = = = = = = = = = = = = = = = =
    定义全局元素
    = = = = = = = = = = = = = = = = = = = = = = = = = -->
  < element name = "TreesA_1M"
      type = "myns:TreesA_1M_Type"
      substitutionGroup = "gml:AbstractFeature"/>
  < element name = "RoadL_1M"
      type = "myns:RoadL_1M_Type"
      substitutionGroup = "gml:AbstractFeature"/>
  < element name = "LakesA_1M"
      type = "myns:LakesA_1M_Type"
      substitutionGroup = "gml:AbstractFeature"/>
  <! -- = = = = = = = = = = = = = = = = = = = = = = = =
    定义复杂类型(类)
    = = = = = = = = = = = = = = = = = = = = = = = = = -->
  < complexType name = "TreesA_1M_Type">
```

```
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="extent"
                type="gml:SurfacePropertyType" nillable="false"/>
          <element name="id" nillable="true" minOccurs="0">
            <simpleType>
              <restriction base="integer">
                <totalDigits value="10"/>
              </restriction>
            </simpleType>
          </element>
          <element name="treeType" nillable="true" minOccurs="0">
            <simpleType>
              <restriction base="string">
                <maxLength value="80"/>
              </restriction>
            </simpleType>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="RoadL_1M_Type">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="centerLine"
                type="gml:CurvePropertyType"
                nillable="false"/>
          <element name="designation" nillable="true" minOccurs="0">
            <simpleType>
              <restriction base="string">
                <maxLength value="30"/>
              </restriction>
            </simpleType>
          </element>
          <element name="surfaceType" nillable="true" minOccurs="0">
            <simpleType>
              <restriction base="string">
                <maxLength value="30"/>
              </restriction>
            </simpleType>
          </element>
```

```
            <element name="nLanes" nillable="true" minOccurs="0">
              <simpleType>
                <restriction base="integer">
                  <totalDigits value="2"/>
                </restriction>
              </simpleType>
            </element>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
    <complexType name="LakesA_1M_Type">
      <complexContent>
        <extension base="gml:AbstractFeatureType">
          <sequence>
            <element name="extent" type="gml:SurfacePropertyType"/>
            <element name="name" type="xsd:string"/>
            <element name="waterType" type="xsd:string" minOccurs="0"/>
            <element name="avgDepth"
                  type="gml:MeasureType" minOccurs="0"/>
            <element name="maxDepth"
                  type="gml:MeasureType" minOccurs="0"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </schema>
```

注意响应文档中不但包含被请求的要素 TreesA_1M 和 RoadL_1M，而且包括应用模式中的所有其他要素(本示例中为 LakesA_1M)。也就是说，服务器生成了完整的应用模式。

使用这个模式描述，在下面的示例中，客户端可以表示 TreesA_1M 和(或)RoadL_1M 要素实例的状态，示例如下：

```
<? xml version="1.0" ? >
<wfs:FeatureCollection
  timeStamp="2008-09-07T19:00:00"
  numberReturned="2"
  numberMatched="unknown"
  xmlns="http://www.someserver.com/myns"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.someserver.com/myns
```

```
                    .DescribeFeatureType_Example01_Response.xsd">
    <wfs:member>
      <TreesA_1M gml:id="TRESSA_1M.1013">
        <extent>
          <gml:Polygon gml:id="GID_10" srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:exterior>
              <gml:LinearRing>
                  <gml:posList srsDimension="2">65.588264 -120.000000 65.590782
-120.003571 65.590965 -120.011292 65.595215 -120.022491 65.592880 -120.031212 65.586121
-120.019363 65.585365 -120.030350 65.581848 -120.045082 65.584938 -120.059540 65.590500
-120.067284 65.595436 -120.067284 65.613441 -120.067337 65.613777 -120.067337 65.606346
-120.060997 65.605545 -120.045517 65.599777 -120.022675 65.601036 -120.003975 65.602081
-120.000000 65.602081 -120.000000 65.588264 -120.000000</gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </extent>
        <id>0000000002</id>
        <treeType>Maple</treeType>
      </TreesA_1M>
    </wfs:member>
    <wfs:member>
      <RoadL_1M gml:id="ROADL_1M.1914">
        <centerLine>
          <gml:LineString gml:id="GID_5" srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:posList>-59.478340 -52.226578 -59.484871 -52.223564 -59.488991 -52.198524
-59.485958 -52.169559 -59.480400 -52.152615 -59.465576 -52.141491 -59.462002 -52.136417 -59.447968
-52.127190 -59.422928 -52.120701 -59.411915 -52.117844 -59.397972 -52.116440 -59.371311 -52.121300
</gml:posList>
          </gml:LineString>
        </centerLine>
        <designation>HYW 401</designation>
        <surfaceType>ASPHALT</surfaceType>
        <nLanes>12</nLanes>
      </RoadL_1M>
    </wfs:member>
  </wfs:FeatureCollection>
```

B.2.2 示例 2

DescribeFeatureType 请求的响应为：

```
<? xml version="1.0" ? >
<DescribeFeatureType
  service="WFS"
  version="2.0.0"
  outputFormat=" application/gml + xml; version=3.2"
```

```
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
            http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
  < TypeName > myns:Person </TypeName >
</DescribeFeatureType >
```

WFS 可能生成如下 XML 模式文档：

```
<? xml version = "1.0"? >
< xsd:schema
  targetNamespace = "http://www.someserver.com/myns"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:gml = "http://www. opengis.net/gml/3.2"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns:gml = "http://www. w3.org/2001/XMLSchema"
  elementFormDefault = "qualified"
  version = "1.0">
  < import namespace = "http://www.opengis.net/gml/3.2"
          schemaLocation = "http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  < element name = "Person" type = "myns:PersonType"
        substitutionGroup = "gml:AbstractFeature"/>
  < complexType name = "PersonType">
    < complexContent >
      < extension base = "gml:AbstractFeatureType">
        < sequence >
          < element name = "lastName" nillable = "true">
            < simpleType >
              < restriction base = "string">
                < maxLength value = "30"/>
              </restriction >
            </simpleType >
          </element >
          < element name = "firstName" nillable = "true">
            < simpleType >
              < restriction base = "string">
                < maxLength value = "10"/>
              </restriction >
            </simpleType >
          </element >
          < element name = "age" type = "integer" nillable = "true"/>
          < element name = "sex" type = "string"/>
          < element name = "spouse"
                type = "myns:PersonPropertyType" minOccurs = "0"/>
          < element name = "location"
                type = "gml:PointPropertyType"
```

```
                nillable = "true"/>
            < element name = "mailAddress"
                type = "myns:AddressPropertyType" nillable = "true"/>
            < element name = "phone" type = "xsd:string"
                minOccurs = "0" maxOccurs = "unbounded"/>
            < element name = "livesIn" type = "myns:HousePropertyType"
                minOccurs = "0"/>
            < element name = "isDriving" type = "myns:CarPropertyType"
                minOccurs = "0"/>
          </sequence >
        </extension >
      </complexContent >
    </complexType >
    < complexType name = "PersonPropertyType">
      < sequence >
        < element ref = "myns:Person" minOccurs = "0"/>
      </sequence >
      < attributeGroup ref = "gml:AssociationAttributeGroup"/>
    </complexType >
    < complexType name = "AddressPropertyType">
      < sequence >
        < element name = "Address"
            type = "myns:AddressType" minOccurs = "0" />
      </sequence >
      < attributeGroup ref = "gml:AssociationAttributeGroup"/>
    </complexType >
    < complexType name = "AddressType">
      < sequence >
        < element name = "streetName" nillable = "true">
          < simpleType >
            < restriction base = "string">
              < maxLength value = "30"/>
            </restriction >
          </simpleType >
        </element >
        < element name = "streetNumber" nillable = "true">
          < simpleType >
            < restriction base = "string">
              < maxLength value = "10"/>
            </restriction >
          </simpleType >
        </element >
        < element name = "city" nillable = "true">
          < simpleType >
```

```
          <restriction base = "string">
            <maxLength value = "30"/>
          </restriction>
        </simpleType>
      </element>
      <element name = "province" nillable = "true">
        <simpleType>
          <restriction base = "string">
            <maxLength value = "30"/>
          </restriction>
        </simpleType>
      </element>
      <element name = "postalCode" nillable = "true">
        <simpleType>
          <restriction base = "string">
            <maxLength value = "15"/>
          </restriction>
        </simpleType>
      </element>
      <element name = "country" nillable = "true">
        <simpleType>
          <restriction base = "string">
            <maxLength value = "30"/>
          </restriction>
        </simpleType>
      </element>
    </sequence>
    <attribute ref = "gml:id" use = "required"/>
  </complexType>
  <element name = "Car" type = "myns:CarType"
        substitutionGroup = "gml:AbstractFeature"/>
  <complexType name = "CarType">
    <complexContent>
      <extension base = "gml:AbstractFeatureType">
        <sequence>
          <element name = "model" type = "xsd:string"/>
          <element name = "age" type = "xsd:nonNegativeInteger"/>
          <element name = "colour">
            <simpleType>
              <restriction base = "string">
                <enumeration value = "red"/>
                <enumeration value = "green"/>
                <enumeration value = "blue"/>
                <enumeration value = "yellow"/>
```

```
                <enumeration value="black"/>
                <enumeration value="white"/>
              </restriction>
            </simpleType>
          </element>
          <element name="location" type="gml:PointPropertyType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="CarPropertyType">
    <sequence>
      <element ref="myns:Car" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <element name="House" type="myns:HouseType"
      substitutionGroup="gml:AbstractFeature"/>
  <complexType name="HouseType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="numFloors" type="xsd:nonNegativeInteger"/>
          <element name="area" type="gml:MeasureType"/>
          <element name="location" type="gml:PointPropertyType"/>
          <element name="frontsOn" type="gml:CurvePropertyType"/>
          <element name="address" type="myns:AddressPropertyType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="HousePropertyType">
    <sequence>
      <element ref="myns:House" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
</schema>
```

如 B.2.1 中示例，服务器生成一个完整的应用模式，此模式中包含请求的要素类型（即 myns:Person），也包括应用模式中其他所有的要素类型。

通过这个模式验证有效的一个文档样例如下：

```
<? xml version="1.0" ? >
<wfs:FeatureCollection
  timeStamp="2008-09-07T19:00:00"
```

```
numberReturned = "2"
numberMatched = "unknown"
xmlns = "http://www.someserver.com/myns"
xmlns:myns = "http://www.someserver.com/myns"
xmlns:rds = "http://www.someserver.com/rds"
xmlns:wfs = "http://www.opengis.net/wfs/2.0"
xmlns:gml = "http://www.opengis.net/gml/3.2"
xmlns:xlink = "http://www.w3.org/1999/xlink"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.someserver.com/myns
              ./DescribeFeatureType_Example02_Response.xsd
              http://www.someserver.com/rds
              ./DescribeFeatureType_Example02_Road.xsd">
<wfs:member>
  <myns:Person gml:id = "p4456">
    <gml:identifier
    codeSpace = "http://www.canadaSIN.com"> 424679374 </gml:identifier>
    <myns:lastName> Smith </myns:lastName>
    <myns:firstName> Fred </myns:firstName>
    <myns:age> 35 </myns:age>
    <myns:sex> male </myns:sex>
    <myns:spouse xlink:href = "#p4467"/>
    <myns:location xlink:href = "#pt102" />
    <myns:mailAddress xlink:href = "#a201"/>
    <myns:phone> 416-123-4567 </myns:phone>
    <myns:phone> 416-890-1234 </myns:phone>
    <myns:livesIn xlink:href = "#h32"/>
  </myns:Person>
</wfs:member>
<wfs:member>
  <myns:Person gml:id = "p4467">
    <gml:identifier
    codeSpace = "http://www.canadaSIN.com"> 424679360 </gml:identifier>
    <myns:lastName> Smith </myns:lastName>
    <myns:firstName> Mary </myns:firstName>
    <myns:age> 18 </myns:age>
    <myns:sex> female </myns:sex>
    <myns:spouse xlink:href = "#p4456"/>
    <myns:location xlink:href = "#pt101" />
    <myns:mailAddress xlink:href = "#a201"/>
    <myns:phone> 416-123-4567 </myns:phone>
    <myns:phone> 416-890-4532 </myns:phone>
    <myns:livesIn xlink:href = "#h32"/>
```

```
      <myns:isDriving xlink:href="#r1432"/>
    </myns:Person>
  </wfs:member>
  <wfs:member>
    <myns:Car gml:id="r1432">
      <gml:identifier
      codeSpace="http://www.carserial.org">51465243</gml:identifier>
      <myns:model>Ford Pinto</myns:model>
      <myns:age>4</myns:age>
      <myns:colour>red</myns:colour>
      <myns:location>
        <gml:Point gml:id="pt102">
          <gml:pos>-59.603958 -52.106559</gml:pos>
        </gml:Point>
      </myns:location>
    </myns:Car>
  </wfs:member>
  <wfs:member>
    <myns:House gml:id="h32">
      <gml:identifier
      codeSpace="http://www.toronto.ca/reg.xml">654365143</gml:identifier>
      <myns:numFloors>2</myns:numFloors>
      <myns:area>200</myns:area>
      <myns:location>
        <gml:Point gml:id="pt101">
          <gml:pos>16 18</gml:pos>
        </gml:Point>
      </myns:location>
      <myns:frontsOn xlink:href="#rs11"/>
      <myns:address>
        <myns:Address gml:id="a201">
          <myns:streetName>Main St.</myns:streetName>
          <myns:streetNumber>5</myns:streetNumber>
          <myns:city>SomeCity</myns:city>
          <myns:province>Someprovince</myns:province>
          <myns:postalCode>X1X 1X1</myns:postalCode>
          <myns:country>Canada</myns:country>
        </myns:Address>
      </myns:address>
    </myns:House>
  </wfs:member>
  <wfs:member>
    <rds:Road gml:id="rs11">
      <rds:numLanes>3</rds:numLanes>
```

```
        <rds:centerline>
          <gml:LineString gml:id="GID_5" srsName="urn:ogc::def:crs:EPSG::4326">
            <gml:posList>-59.478340 -52.226578 -59.484871 -52.223564 -59.488991 -52.198524 -
59.485958 -52.169559 -59.480400 -52.152615 -59.465576 -52.141491 -59.462002 -52.136417 -59.447968
-52.127190 -59.422928 -52.120701 -59.411915 -52.117844 -59.397972 -52.116440 -59.371311 -52.121300
</gml:posList>
          </gml:LineString>
        </rds:centerline>
      </rds:Road>
    </wfs:member>
  </wfs:FeatureCollection>
```

B.3 GetFeature 示例

B.3.1 概述

这部分包含大量 GetFeature 请求的示例。一些示例包含输出实例。

B.3.2 示例 1

这个示例获取要素类型*InWaterA_1M* 的具体实例，其标识符为"*InWaterA_1M.1234*"。

```
<? xml version="1.0" ? >
<wfs:GetFeature
  service="WFS"
  version="2.0.0"
  outputFormat=" application/gml + xml; version = 3.2"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  <wfs:Query typeName="myns:InWaterA_1M">
    <fes:Filter>
      <fes:ResourceId rid="InWaterA_1M.1234"/>
    </fes:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

B.3.3 示例 2

这个示例获取要素类型 InWaterA_1M 的特性子集。此请求获取的具体实例标识符为"InWaterA_1M.1013"。

```
<? xml version="1.0" ? >
<wfs:GetFeature
  service="WFS"
  version="2.0.0"
```

```
      xmlns:wfs = "http://www.opengis.net/wfs/2.0"
      xmlns:fes = "http://www.opengis.net/fes/2.0"
      xmlns:myns = "http://www.someserver.com/myns"
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
      <wfs:Query typeNames = "myns:InWaterA_1M">
        <wfs:PropertyName>myns:wkbGeom</wfs:PropertyName>
        <wfs:PropertyName>myns:tileId</wfs:PropertyName>
        <wfs:PropertyName>myns:facId</wfs:PropertyName>
        <fes:Filter>
          <fes:ResourceId rid = "InWaterA_1M.1013"/>
        </fes:Filter>
      </wfs:Query>
    </wfs:GetFeature>
```

B.3.4 示例 3

对于一组枚举的要素实例，本例请求要素类型 InWaterA_1M 所有的特性。fes:ResourceId 元素用来标识请求的每个要素。

```
    <? xml version = "1.0" ? >
    <GetFeature
      version = "2.0.0"
      service = "WFS"
      xmlns = "http://www.opengis.net/wfs/2.0"
      xmlns:fes = "http://www.opengis.net/fes/2.0"
      xmlns:myns = "http://www.someserver.com/myns"
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
      <Query typeNames = "myns:InWaterA_1M">
       <fes:Filter>
         <fes:ResourceId rid = "InWaterA_1M.1013"/>
         <fes:ResourceId rid = "InWaterA_1M.1014"/>
         <fes:ResourceId rid = "InWaterA_1M.1015"/>
        </fes:Filter>
      </Query>
    </GetFeature>
```

B.3.5 示例 4

本例与上一个示例相似，但是本例只请求了枚举要素集的部分特性。wfs:PropertyName 元素用来列举检索的特性。

```
    <? xml version = "1.0" ? >
    <GetFeature
      version = "2.0.0"
      service = "WFS"
```

```
    xmlns = "http://www.opengis.net/wfs/2.0"
    xmlns:fes = "http://www.opengis.net/fes/2.0"
    xmlns:myns = "http://www.someserver.com/myns"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = " http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
    <Query typeNames = "myns:InWaterA_1M">
      <wfs:PropertyName>myns:wkbGeom</wfs:PropertyName>
      <wfs:PropertyName>myns:tileId</wfs:PropertyName>
      <fes:Filter>
        <fes:ResourceId rid = "InWaterA_1M.1013"/>
        <fes:ResourceId rid = "InWaterA_1M.1014"/>
        <fes:ResourceId rid = "InWaterA_1M.1015"/>
      </fes:Filter>
    </Query>
  </GetFeature>
```

对该请求响应的样例片段如下：

```
  <? xml version = "1.0"? >
  <wfs:FeatureCollection
    timeStamp = "2010-02-01T22:56:09"
    numberMatched = "3"
    numberReturned = "3"
    xmlns = "http://www.someserver.com/myns"
    xmlns:myns = "http://www.someserver.com/myns"
    xmlns:wfs = "http://www.opengis.net/wfs/2.0"
    xmlns:gml = "http://www.opengis.net/gml/3.2"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.someserver.com/myns ./GetFeature_05.xsd
                  http://www.opengis.net/wfs/2.0
                  http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                  http://www.opengis.net/gml/3.2
                  http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    <wfs:boundedBy>
      <gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
        <gml:lowerCorner>-31.27141761779785 115.001335144043</gml:lowerCorner>
        <gml:upperCorner>-30.10202789306641 117.9325866699219</gml:upperCorner>
      </gml:Envelope>
    </wfs:boundedBy>
  <wfs:member>
    <InWaterA_1M gml:id = "InWaterA_1M.1013">
      <wkbGeom>
        <gml:Polygon srsName = "urn:ogc:def:crs:EPSG::4326" gml:id = "P1">
          <gml:exterior>
            <gml:LinearRing>
              <gml:posList>-30.93597221374512 117.6290588378906 -30.94830513000489
```

117.6447219848633 -30.95219421386719 117.6465530395508 -30.95219421386719 117.6431121826172 -30.94802856445312 117.6386108398438 -30.94799995422363 117.6314163208008 -30.946138381958 117.62850189209 -30.94430541992188 117.6295852661133 -30.93280601501464 117.6240539550781 -30.92869377136231 117.624641418457 -30.92386054992676 117.6201400756836 -30.92111206054688 117.6206970214844 -30.92458343505859 117.6275863647461 -30.93597221374512 117.6290588378906</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</wkbGeom>
<id>28022</id>
<tileId>177</tileId>
</InWaterA_1M>
</wfs:member>
<wfs:member>
<InWaterA_1M gml:id="InWaterA_1M.1014">
<wkbGeom>
<gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326" gml:id="P2">
<gml:exterior>
<gml:LinearRing>
<gml:posList>-30.92013931274414 117.6552810668945 -30.92383384704589 117.661361694336 -30.93005561828613 117.6666412353516 -30.93280601501464 117.6663589477539 -30.93186187744141 117.6594467163086 -30.93780517578125 117.6541137695312 -30.94397163391114 117.6519470214844 -30.94255638122559 117.6455535888672 -30.93402862548828 117.6336364746094 -30.92874908447266 117.6355285644531 -30.92138862609864 117.6326370239258 -30.92236137390137 117.6395568847656 -30.91708374023438 117.6433029174805 -30.91711044311523 117.6454467773437 -30.92061042785645 117.6484985351563 -30.92061042785645 117.6504135131836 -30.91638946533203 117.6504440307617 -30.92013931274414 117.6552810668945</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</wkbGeom>
<id>28021</id>
<tileId>177</tileId>
</InWaterA_1M>
</wfs:member>
<wfs:member>
<InWaterA_1M gml:id="InWaterA_1M.1015">
<wkbGeom>
<gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326" gml:id="P3">
<gml:exterior>
<gml:LinearRing>
<gml:posList>-31.27141761779785 117.9237747192383 -31.2668342590332 117.9219131469727 -31.26474952697754 117.9165802001953 -31.26177787780761 117.9173889160156 -31.25813865661621 117.9240798950195 -31.25927734375 117.928337097168 -31.26297187805175

```
117.9320526123047-31.26572227478028 117.9325866699219 -31.27097129821777 117.9283065795898
-31.27141761779785  117.9237747192383 </gml:posList >
                </gml:LinearRing >
              </gml:exterior >
            </gml:Polygon >
          </wkbGeom >
          < id > 28074 </id >
          < tileId > 177 </tileId >
        </InWaterA_1M >
      </wfs:member >
    </wfs:FeatureCollection >
```

可以看到,该要素实例包括请求的特性 wkbGeom,tileId 和 id,其中包含 id 是 WFS 为了产生有效地响应文档(见 7.9.2.4.5.1)。

B.3.6 示例 5

选择要素类型*InWaterA_1M* 的所有实例,最多为 10 000 个要素。

```
<? xml version = "1.0" ? >
< GetFeature
  version = "2.0.0"
  service = "WFS"
  count = "10000"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = " http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  < Query typeNames = "myns:InWaterA_1M"/>
</GetFeature >
```

B.3.7 示例 6

下面未约束的请求要求获取一个枚举的要素类型集的所有实例。注意要素类型不全都是在一个命名空间之下。

```
<? xml version = "1.0" ? >
< GetFeature
  version = "2.0.0"
  service = "WFS"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:yourns = "http://demo.someserver.com/yourns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = " http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  < Query typeNames = "myns:InWaterA_1M"/>
  < Query typeNames = "myns:BuiltUpA_1M"/>
  < Query typeNames = "yourns:RoadL_1M"/>
```

```
</GetFeature>
```

B.3.8 示例 7

下面的示例选择在 Grand Banks 范围内的要素 Hydrograghy 的几何形状和深度。Grand Banks 的范围为[46.2023,-57.9118, 51.8145, -46.6873]。

```
<? xml version = "1.0" ? >
< GetFeature
  version = "2.0.0"
  service = "WFS"
  handle = "Query01"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = " http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
    http://www.opengis.net/gml/3.2
            http://schemas.opengis.net/gml/3.2.1/gml.xsd
            http://www.someserver.com/myns ./GetFeature_07.xsd">
  < Query typeNames = "myns:Hydrography">
    < PropertyName > myns:geoTemp </ PropertyName >
    < PropertyName > myns:depth </ PropertyName >
  < PropertyName > myns:temperature </PropertyName >
    < fes:Filter >
      < fes:Not >
        < fes:Disjoint >
          < fes:ValueReference > myns:geoTemp </fes:ValueReference >
          < gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
            < gml:lowerCorner > 46.2023 -57.9118 </gml:lowerCorner >
            < gml:upperCorner > 51.8145 -46.6873 </gml:upperCorner >
          </gml:Envelope >
        </fes:Disjoint >
      </fes:Not >
    </fes:Filter >
  < fes:SortBy >
      < fes:SortProperty >
        < fes:ValueReference > myns:depth </fes:ValueReference >
      </fes:SortProperty >
      < fes:SortProperty >
        < fes:ValueReference > myns:temperature </fes:ValueReference >
        < fes:SortOrder > DESC </fes:SortOrder >
      </fes:SortProperty >
    </fes:SortBy >
  </Query >
```

```
</GetFeature>
```

该请求的输出可能是：

```
<? xml version = "1.0" encoding = "UTF-8"? >
< wfs:FeatureCollection
timeStamp = "2009-11-04T11:05:00"
numberMatched = "3"
numberReturned = "3"
xmlns = "http://www.someserver.com/myns"
xmlns:wfs = "http://www.opengis.net/wfs/2.0"
xmlns:gml = "http://www.opengis.net/gml/3.2"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation = "http://www.someserver.com/myns ./GetFeature_07.xsd
                http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.opengis.net/gml/3.2
                http://schemas.opengis.net/gml/3.2.1/gml.xsd">
< wfs:member >
  < Hydrography gml:id = "HydrographyHydrography.450">
    < geoTemp >
     < gml:Point gml:id = "GID_3" srsName = "urn:ogc:def:crs:EPSG::4326">
       < gml:pos > 47.2002 -56.314 </gml:pos >
     </gml:Point >
    </geoTemp >
    < depth uom = "mm"> 565 </depth >
    < temperature uom = "C"> 10 </temperature >
  </Hydrography >
</wfs:member >
< wfs:member >
  < Hydrography gml:id = "HydrographyHydrography.451">
    < geoTemp >
     < gml:Point gml:id = "GID_4" srsName = "urn:ogc:def:crs:EPSG::4326">
       < gml:pos > 47.2003 -56.313 </gml:pos >
     </gml:Point >
    </geoTemp >
    < depth uom = "mm"> 1015 </depth >
    < temperature uom = "C"> 7 </temperature >
  </Hydrography >
</wfs:member >
< wfs:member >
  < Hydrography gml:id = "HydrographyHydrography.452">
    < geoTemp >
     < gml:Point gml:id = "GID_5" srsName = "urn:ogc:def:crs:EPSG::4326">
       < gml:pos > 47.2003 -56.313 </gml:pos >
     </gml:Point >
    </geoTemp >
```

```
        <depth uom = "mm"> 2456 </depth>
        <temperature uom = "C"> 4 </temperature>
      </Hydrography>
    </wfs:member>
    </wfs:FeatureCollection>
```

B.3.9 示例 8

这个示例描述了在单个感兴趣区域内获取 *ROADS* 和 *RAILS* 实例的查询。

```
    <? xml version = "1.0" ? >
    <GetFeature
      version = "2.0.0"
      service = "WFS"
      handle = "Example Query"
      xmlns = "http://www.opengis.net/wfs/2.0"
      xmlns:fes = "http://www.opengis.net/fes/2.0"
      xmlns:gml = "http://www.opengis.net/gml/3.2"
      xmlns:myns = "http://www.someserver.com/myns"
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation = " http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
                        http://www.opengis.net/gml/3.2
                    http://schemas.opengis.net/gml/3.2.1/gml.xsd">
      <Query typeNames = "myns:Roads" handle = "Q01">
        <PropertyName > myns:path </PropertyName >
        <PropertyName > myns:nLanes </PropertyName >
        <PropertyName > myns:surfaceType </PropertyName >
        <fes:Filter >
          <fes:Within >
            <fes:ValueReference > myns:path </fes:ValueReference >
            <fes:Literal >
              <gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
                <gml:lowerCorner > 50 40 </gml:lowerCorner >
                <gml:upperCorner > 100 60 </gml:upperCorner >
              </gml:Envelope >
            </fes:Literal >
          </fes:Within >
        </fes:Filter >
      </Query >
      <Query typeNames = "myns:Rails" handle = "Q02">
        <PropertyName > myns:track </PropertyName >
        <PropertyName > myns:gauge </PropertyName >
        <fes:Filter >
          <fes:Within >
            <fes:ValueReference > myns:track </fes:ValueReference >
            <fes:Literal >
```

```
            <gml:Envelope srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:lowerCorner>-33 103</gml:lowerCorner>
              <gml:upperCorner>1 135</gml:upperCorner>
            </gml:Envelope>
          </fes:Literal>
        </fes:Within>
      </fes:Filter>
    </Query>
  </GetFeature>
```

这两个请求的结果依据 11.3.3.5 合在一起组成了结果要素集。

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection
  timeStamp="2008-08-01T22:47:02"
  numberMatched="unknown"
  numberReturned="200"
  xmlns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.someserver.com/myns ./RoadRail.xsd
                      http://www.opengis.net/wfs/2.0
                      http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                      http://www.opengis.net/gml/3.2
                      http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  <wfs:boundedBy>
    <gml:Envelope srsName="urn:ogc:def:crs:EPSG::4326">
      <gml:lowerCorner>-32.7638053894043 104.1429748535156</gml:lowerCorner>
      <gml:upperCorner>0 133.3553924560547</gml:upperCorner>
    </gml:Envelope>
  </wfs:boundedBy>
  <!-- Road collection -->
  <wfs:member>
    <wfs:FeatureCollection
      timeStamp="2008-08-01T22:47:02"
      numberMatched="unknown"
      numberReturned="167">
      <wfs:member>
        <Road gml:id="Road.100">
          <centerLine>
            <gml:LineString gml:id="LS1" srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:posList>-24.90258407592773 113.6248092651367 -24.90200042724609
113.6278305053711 -24.89602851867676 113.6336364746094 -24.89147186279297 113.6345825195312 -
24.8884449005127 113.6405029296875 -24.87958335876465 113.6336669921875 -24.8785285949707
113.6328582763672 -24.87274932861328 113.6313323974609 -24.87163925170898 113.6195526123047 -
24.87466621398926 113.614387512207 -24.87716674804688 113.6137771606445 -24.88877868652344
113.6236114501953 -24.89338874816895 113.6241683959961 -24.89838981628418 113.622444152832 -
```

```
24.90258407592773 113.6248092651367 </gml:posList >
                </gml:LineString >
              </centerLine >
              < surfaceType > ASPHALT </surfaceType >
              < nLanes > 4 </nLanes >
            </Road >
          </wfs:member >
          < wfs:member >
            < Road gml:id = "Road.105">
              < centerLine >
                < gml:LineString gml:id = "LS2" srsName = "urn:ogc:def:crs:EPSG::4326">
                  < gml:posList >-0.3605277836322785 104.2599411010742 -0.3567777872085571
104.2643585205078 -0.3565555512905121 104.2669982910156 -0.3578888773918152
104.2689743041992 -0.3623055517673492 104.2698364257812 -0.3702777922153473
104.2678298950195 -0.3751111030578613 104.2592468261719 -0.3728888928890228
104.2574996948242 -0.3664722144603729 104.2566375732422 -0.3605277836322785
104.2599411010742 </gml:posList >
                </gml:LineString >
              </centerLine >
              < surfaceType > GRAVEL </surfaceType >
              < nLanes > 2 </nLanes >
            </Road >
          </wfs:member >
          <! — ... more Road instances ... —>
        </wfs:FeatureCollection >
      </wfs:member >
      <! — Rail collection —>
      < wfs:member >
        < wfs:FeatureCollection
          timeStamp = "2008-08-01T22:47:02"
          numberMatched = "unknown"
          numberReturned = "33">
          < wfs:member >
            < Rail gml:id = "Rail.119">
              < track >
                < gml:LineString gml:id = "LS3" srsName = "urn:ogc:def:crs:EPSG::4326">
                  < gml:posList >-15.34877777099609 124.3914184570313 -15.34572219848633
124.3990859985352 -15.34027767181396 124.39111328125 -15.34066677093506 124.3884963989258 -
15.34805583953857 124.3888320922852 -15.34877777099609  124.3914184570313 </gml:posList >
                </gml:LineString >
              </track >
              < gauge > 24 </gauge >
            </Rail >
          </wfs:member >
          <! — ... more Rail instances ... —>
```

```
    </wfs:FeatureCollection>
  </wfs:member>
</wfs:FeatureCollection>
```

B.3.10 示例 9

这个示例描述如何使用 XPath 表达式引用要素的复合特性。要素类型 *Person* 定义为：

```
<? xml version = "1.0" ? >
< schema
  targetNamespace = "http://www.someserver.com/myns"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "qualified"
  version = "1.0">
  < import namespace = "http://www.opengis.net/gml/3.2"
       schemaLocation = "http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
  < element name = "Person" type = "myns:PersonType"
        substitutionGroup = "gml:AbstractFeature"/>
  < complexType name = "PersonType">
    < complexContent >
      < extension base = "gml:AbstractFeatureType">
        < sequence >
          < element name = "lastName" nillable = "true">
            < simpleType >
              < restriction base = "string">
                < maxLength value = "30"/>
              </restriction >
            </simpleType >
          </element >
          < element name = "firstName" nillable = "true">
            < simpleType >
              < restriction base = "string">
                < maxLength value = "10"/>
              </restriction >
            </simpleType >
          </element >
          < element name = "age" type = "integer" nillable = "true"/>
          < element name = "sex" type = "string"/>
          < element name = "spouse">
            < complexType >
              < attribute name = "SIN" type = "xsd:anyURI" use = "required" />
            </complexType >
          </element >
          < element name = "location"
```

```
                type = "gml:PointPropertyType"
                nillable = "true"/>
            <element name = "mailAddress"
                type = "myns:AddressPropertyType" nillable = "true"/>
            <element name = "salary" type = "positiveInteger" nillable = "true"/>
          </sequence>
          <attribute name = "SIN" type = "xsd:anyURI" use = "required"/>
        </extension>
      </complexContent>
    </complexType>
    <complexType name = "AddressPropertyType">
      <sequence>
        <element name = "Address"
            type = "myns:AddressType" minOccurs = "0" />
      </sequence>
    </complexType>
    <complexType name = "AddressType">
      <sequence>
        <element name = "streetName" nillable = "true">
          <simpleType>
            <restriction base = "string">
              <maxLength value = "30"/>
            </restriction>
          </simpleType>
        </element>
        <element name = "streetNumber" nillable = "true">
          <simpleType>
            <restriction base = "string">
              <maxLength value = "10"/>
            </restriction>
          </simpleType>
        </element>
        <element name = "city" nillable = "true">
          <simpleType>
            <restriction base = "string">
              <maxLength value = "30"/>
            </restriction>
          </simpleType>
        </element>
        <element name = "province" nillable = "true">
          <simpleType>
            <restriction base = "string">
              <maxLength value = "30"/>
            </restriction>
```

```
            </simpleType>
          </element>
          <element name="postalCode" nillable="true">
            <simpleType>
              <restriction base="string">
                <maxLength value="15"/>
              </restriction>
            </simpleType>
          </element>
          <element name="country" nillable="true">
            <simpleType>
              <restriction base="string">
                <maxLength value="30"/>
              </restriction>
            </simpleType>
          </element>
        </sequence>
      </complexType>
    </schema>
```

mailAddress 是一个复合特性。

下面示例获取一群女性的姓，这群女性是住在“SomeTown”镇的第 10 000 街区“Main St.”上并且年收入在 $35 000 之上的所有女性。注意在谓词中使用 XPath 表达式来引用复合特性。

```
<? xml version="1.0" ? >
<GetFeature
  version="2.0.0"
  service="WFS"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.someserver.com/myns ./GetFeature_09.xsd">
  <Query typeNames="Person">
    <PropertyName>myns:lastName</ PropertyName>
    <fes:Filter>
      <fes:And>
        <fes:And>
          <fes:PropertyIsGreaterThanOrEqualTo>
<fes:ValueReference>myns:Person/myns:mailAddress/myns:Address/myns:streetNumber</
fes:ValueReference>
            <fes:Literal>10000</fes:Literal>
          </fes:PropertyIsGreaterThanOrEqualTo>
          <fes:PropertyIsLessThanOrEqualTo>
<fes:ValueReference>myns:Person/myns:mailAddress/myns:Address/myns:streetNumber</
```

```
fes:ValueReference>
                <fes:Literal>10999</fes:Literal>
              </fes:PropertyIsLessThanOrEqualTo>
            </fes:And>
            <fes:And>
              <fes:PropertyIsEqualTo>
    <fes:ValueReference>myns:Person/myns:mailAddress/myns:Address/myns:streetName</fes:
ValueReference>
                <fes:Literal>Main St.</fes:Literal>
              </fes:PropertyIsEqualTo>
              <fes:PropertyIsEqualTo>
    <fes:ValueReference>myns:Person/myns:mailAddress/myns:Address/myns:city</fes:
ValueReference>
                <fes:Literal>SomeTown</fes:Literal>
              </fes:PropertyIsEqualTo>
              <fes:PropertyIsEqualTo>
                <fes:ValueReference>myns:Person/myns:sex</fes:ValueReference>
                <fes:Literal>Female</fes:Literal>
              </fes:PropertyIsEqualTo>
              <fes:PropertyIsGreaterThan>
                <fes:ValueReference>myns:Person/myns:salary</fes:ValueReference>
                <fes:Literal>35000</fes:Literal>
              </fes:PropertyIsGreaterThan>
            </fes:And>
          </fes:And>
        </fes:Filter>
      </Query>
    </GetFeature>
```

B.3.11 示例 10

这个示例使用 wfs:PropertyName 元素描述 GetFeature 请求，用来与例 11 和例 12 作比较。请求如下：

```
<? xml version = "1.0"? >
<wfs:GetFeature
  service = "WFS"
  version = "2.0.0"
  outputFormat = "application/gml + xml; version = 3.2"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.opengis.net/gml/3.2
                http://schemas.opengis.net/gml/3.2.1/gml.xsd">
```

```
  <wfs:Query typeNames = "Town">
    <wfs:PropertyName > gml:name </wfs:PropertyName >
    <wfs:PropertyName resolve = "none"> gml:directedNode </wfs:PropertyName >
    <fes:Filter >
      <fes:ResourceId rid = "t1"/>
    </fes:Filter >
  </wfs:Query >
</wfs:GetFeature >
```

响应中包含一个 xlink:href,但是因为属性 resolve 设置为 none,所以该响应返回的是 as-is。

```
<? xml version = "1.0" encoding = "UTF-8"? >
<wfs:FeatureCollection
  timeStamp = "2009-02-14T01:27:44"
  numberMatched = "1"
  numberReturned = "1"
  xmlns = "http://www.someserver.com/myns"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.opengis.net/gml/3.2
                http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  <wfs:boundedBy >
    <gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
      <gml:lowerCorner > 10 10 </gml:lowerCorner >
      <gml:upperCorner > 20 20 </gml:upperCorner >
    </gml:Envelope >
  </wfs:boundedBy >
  <wfs:member >
    <Town gml:id = "t1">
      <gml:name > Bedford </gml:name >
      <gml:directedNode orientation = " + " xlink:href = " # n1"/>
    </Town >
  </wfs:member >
</wfs:FeatureCollection >
```

B.3.12 示例 11

这个示例描述例 10 的 GetFeature 请求中 resovleDepth 和 resolveTimeout 属性的使用。请求如下:

```
<? xml version = "1.0"? >
<wfs:GetFeature
  service = "WFS"
  version = "2.0.0"
  resolveDepth = "1"
```

```
    resolveTimeout = "1"
    outputFormat = "application/gml + xml; version = 3.2"
    xmlns:wfs = "http://www.opengis.net/wfs/2.0"
    xmlns:fes = "http://www.opengis.net/fes/2.0"
    xmlns:gml = "http://www.opengis.net/gml/3.2"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                  http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                  http://www.opengis.net/gml/3.2
                  http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    < wfs:Query typeNames = "Town">
      < wfs:PropertyName > gml:name </wfs:PropertyName >
      < wfs:PropertyName resolve = "all"> gml:directedNode </wfs:PropertyName >
      < fes:Filter >
        < fes:ResourceId rid = "t1"/>
      </fes:Filter >
    </wfs:Query >
  </wfs:GetFeature >
```

响应中，第一层的 xlink:href 被遍历，返回的内容如下：

```
<? xml version = "1.0"? >
< wfs:FeatureCollection
    timeStamp = "2009-02-14T01:27:44"
    numberMatched = "1"
    numberReturned = "1"
    xmlns = "http://www.someserver.com/myns"
    xmlns:wfs = "http://www.opengis.net/wfs/2.0"
    xmlns:gml = "http://www.opengis.net/gml/3.2"
    xmlns:xlink = "http://www.w3.org/1999/xlink"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                  http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                  http://www.opengis.net/gml/3.2
                  http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    < wfs:boundedBy >
      < gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
        < gml:lowerCorner > 10 10 </gml:lowerCorner >
        < gml:upperCorner > 20 20 </gml:upperCorner >
      </gml:Envelope >
    </wfs:boundedBy >
    < wfs:member >
      < Town gml:id = "t1">
        < gml:name > Bedford </gml:name >
        < gml:directedNode orientation = " + " xlink:href = " # n1"/>
        <! -- xlink:href = " # n1"/> -->
        < gml:directedNode orientation = " + ">
          < gml:Node gml:id = "n1">
```

```
            <gml:pointProperty
          xlink:href = "http://www.bedford.town.uk/civilworks/gps.gml#townHall"/>
          </gml:Node>
        </gml:directedNode>
      </Town>
    </wfs:member>
  </wfs:FeatureCollection>
```

B.3.13 示例 12

这个示例描述带有 resolveDepth 和 resolveTimeout 属性的 wfs:XlinkPropertyName 元素的使用。这两个属性重写了例 11 中 GetFeature 请求的两个属性。请求如下：

```
<? xml version = "1.0"? >
<wfs:GetFeature
  service = "WFS"
  version = "2.0.0"
  resolveDepth = "1"
  resolveTimeout = "1"
  outputFormat = "application/gml + xml; version = 3.2"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.opengis.net/gml/3.2
              http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  <wfs:Query typeNames = "Town">
    <wfs:PropertyName>gml:name</wfs:PropertyName>
    <wfs:PropertyName resolve = "all"
                resolveDepth = "2">gml:directedNode</wfs:PropertyName>
    <fes:Filter>
      <fes:ResourceId rid = "t1"/>
    </fes:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

在这个响应中，gml:directedNode 属性中第一层和第二层的 xlink:href 被遍历，如 7.6.4 所述，它们的返回内容如下：

```
<? xml version = "1.0"? >
<wfs:FeatureCollection
  timeStamp = "2009-02-14T01:27:44"
  numberMatched = "1"
  numberReturned = "1"
  xmlns = "http://www.someserver.com/myns"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
```

```
    xmlns:xlink = "http://www.w3.org/1999/xlink"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                  http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                  http://www.opengis.net/gml/3.2
                  http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    <wfs:boundedBy>
      <gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
        <gml:lowerCorner>10 10</gml:lowerCorner>
        <gml:upperCorner>20 20</gml:upperCorner>
      </gml:Envelope>
    </wfs:boundedBy>
    <wfs:member>
      <Town gml:id = "t1">
        <gml:name>Bedford</gml:name>
        <gml:directedNode orientation = " + " xlink:href = " # n1"/>
        <! — xlink:href = " # n1"/> —>
        <gml:directedNode orientation = " + ">
          <gml:Node gml:id = "n1">
            <gml:pointProperty>
              <gml:Point gml:id = "townHall" srsName = "urn:ogc:def:crs:EPSG::4326">
                <gml:pos>-47 34</gml:pos>
              </gml:Point>
            </gml:pointProperty>
          </gml:Node>
        </gml:directedNode>
      </Town>
    </wfs:member>
  </wfs:FeatureCollection>
```

B.3.14 示例 13

下面示例实现了一个连接操作,目的是为找到包含在 Algonquin 公园中的一个湖泊。

```
<? xml version = "1.0"? >
<GetFeature
  service = "WFS"
  version = "2.0.0"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
  <Query typeNames = "myns:Parks myns:Lakes">
    <fes:Filter>
      <fes:And>
```

```
        <fes:PropertyIsEqualTo>
          <fes:ValueReference>/myns:Parks</fes:ValueReference>
          <fes:Literal>Algonquin Park</fes:Literal>
        </fes:PropertyIsEqualTo>
        <fes:Contains>
          <fes:ValueReference>/myns:Parks/geometry</fes:ValueReference>
          <fes:ValueReference>/myns:Lakes/geometry</fes:ValueReference>
        </fes:Contains>
      </fes:And>
    </fes:Filter>
  </Query>
</GetFeature>
```

WFS 对该请求产生的响应如下：

```
<? xml version = "1.0" encoding = "UTF-8"? >
<wfs:FeatureCollection timeStamp = "2008-08-15T11:36:00" numberMatched = "12"
  numberReturned = "12"xmlns = "http://www.someserver.com/myns"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.someserver.com/myns ./SampleSchema.xsd
              http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.opengis.net/gml/3.2
              http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  <wfs:member>
    <wfs:Tuple>
      <wfs:member>
        <Parks gml:id = "Parks.287796">
          <Name>Algonquin Park</Name>
          <Boundary>
            <gml:Polygon gml:id = "GID_13"
              srsName = "urn:ogc:def:crs:EPSG::4326">
              <gml:exterior>
                <gml:LinearRing>
                        <gml:posList>-78.62456512451172 44.95297622680663
-78.62757110595703 44.93315887451171 -78.62944030761717 44.93072891235352 -78.62944030761717
44.93072891235352 -78.64232635498047 44.93759918212891 -78.66877746582031 44.92557144165039
-78.66857147216798 44.92765045166015 -78.65638732910156 44.93304443359375 -78.63591003417969
44.94966506958008 -78.62456512451172 44.95297622680663</gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </Boundary>
        </Parks>
```

```
          </wfs:member>
          <wfs:member>
            <Lakes gml:id="Lakes.287797">
              <Name>Canisbay Lake</Name>
              <Boundary>
                <gml:Polygon gml:id="GID_14"
                  srsName="urn:ogc:def:crs:EPSG::4326">
                  <gml:exterior>
                    <gml:LinearRing>
                      <gml:posList>-78.70862579345703 44.96030044555664
-78.71179962158205 44.96092224121094 -78.71607208251953 44.96665191650391 -78.71385955810547
44.97250747680664 -78.71767425537109 44.97959136962891 -78.71420288085938 44.98193359375
-78.71136474609375 44.98753356933594 -78.70468902587891 44.98627471923828 -78.69895172119141
44.99493026733398 -78.69511413574219 44.9945068359375 -78.69327545166016 44.99028015136719
-78.70228576660156 44.97490310668945 -78.69764709472655 44.96639633178711 -78.6983642578125
44.96253204345703 -78.70416259765625 44.96282958984375 -78.70862579345703 44.96030044555664</gml:
posList>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:Polygon>
              </Boundary>
            </Lakes>
          </wfs:member>
        </wfs:Tuple>
      </wfs:member>
      <wfs:member>
        <wfs:Tuple>
          <wfs:member xlink:href="#Parks.287796"/>
          <wfs:member>
            <Lakes gml:id="Lakes.287798">
              <Name>Kearney Lake</Name>
              <Boundary>
                <gml:Polygon gml:id="GID_15"
                  srsName="urn:ogc:def:crs:EPSG::4326">
                  <gml:exterior>
                    <gml:LinearRing>
                      <gml:posList>-78.60539245605469 45 -78.59363555908202 45
-78.59363555908202 45 -78.59047698974609 44.99884414672852 -78.58409118652344 45
-78.58409118652344 45 -78.58390808105469 45 -78.58390808105469 45 -78.57343292236328 45
-78.57343292236328 45 -78.58406829833984 44.99701690673828 -78.59286499023438
44.98714447021484 -78.5972900390625 44.97566604614257 -78.61244964599609 44.96060943603516
-78.62032318115234 44.9564323425293 -78.62032318115234 44.9564323425293 -78.61922454833984
44.96395111083984 -78.60328674316406 44.9897575378418 -78.60232543945312 44.99590301513671
-78.60539245605469 45</gml:posList>
                    </gml:LinearRing>
```

```
                </gml:exterior>
              </gml:Polygon>
            </Boundary>
          </Lakes>
        </wfs:member>
      </wfs:Tuple>
    </wfs:member>
    <wfs:member>
      <wfs:Tuple>
        <wfs:member xlink:href="#Parks.287796"/>
        <wfs:member>
          <Lakes gml:id="Lakes.287799">
            <Name>Lake Of Two Rivers</Name>
            <Boundary>
              <gml:Polygon gml:id="GML_16"
                srsName="urn:ogc:def:crs:EPSG::4326">
                <gml:exterior>
                  <gml:LinearRing>
                    <gml:posList>-78.50907135009766 44.964599609375 -78.51042175292969
44.96719360351563 -78.50795745849609 44.97211456298828 -78.49131011962891 44.97031402587891 -
78.47936248779297 44.97610855102539 -78.46945190429688 44.97489166259765 -78.46963500976562
44.9700813293457 -78.46161651611328 44.96322250366211 -78.45389556884766 44.96281051635742 -
78.45217895507812 44.96065139770508 -78.46006774902344 44.95647811889649 -78.46690368652344
44.94995498657227 -78.4732666015625 44.95718383789062 -78.47340393066406 44.9620132446289 -
78.47900390625 44.96414947509766 -78.48902893066406 44.96123886108398 -78.49602508544922
44.96551132202148 -78.50370788574219 44.96638107299804 -78.50907135009766 44.964599609375
</gml:posList>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </Boundary>
          </Lakes>
        </wfs:member>
      </wfs:Tuple>
    </wfs:member>
    <wfs:member>
      <wfs:Tuple>
        <wfs:member xlink:href="#Parks.287796"/>
        <wfs:member>
          <Lakes gml:id="Lakes.287806">
            <Name>Mew Lake</Name>
            <Boundary>
              <gml:Polygon gml:id="GID_17"
                srsName="urn:ogc:def:crs:EPSG::4326">
                <gml:exterior>
```

```
                    <gml:LinearRing>
                        <gml:posList>-78 44.96173095703125 -78.01906585693359
44.95022583007812 -78.02398681640625 44.95257568359375 -78.03146362304686 44.94656753540039 -
78.03414916992188 44.95430755615234 -78.03836059570312 44.95431900024414 -78.04071044921875
44.95148086547851 -78.04399108886719 44.94751739501953 -78.05403900146484 44.94165802001952 -
78.06121063232422 44.93264770507813 -78.06439971923828 44.93305587768555 -78.06134033203125
44.94873428344727 -78.05697631835938 44.95009231567382 -78.05594635009766 44.9534797668457 -
78.05167388916016 44.95691299438477 -78.06182861328125 44.96736907958984 -78.06884765625
44.98270034790039 -78.0685577392578 44.98819732666016 -78.06369781494141 44.98815155029297 -
78.05403900146484 44.97611618041992 -78.04621124267578 44.97658920288086 -78.04701232910156
44.96951293945312 -78.03591156005859 44.97025680541992 -78.02657318115234 44.9812126159668 -
78.02198791503906 44.98163986206055 -78.02023315429688 44.97993469238281 -78.02134704589844
44.97585678100586 -78.03009033203125 44.97015380859375 -78.04158782958984 44.9630012512207 -
78.03964996337891 44.96013259887695 -78.03326416015626 44.95930862426758 -78.02040863037109
44.96408843994141 -78.01665496826172 44.96295166015625 -78.00716400146486 44.96378707885742 -
78.00435638427734 44.96569442749023 -78.00571441650391 44.97082901000977 -78 44.97296905517578 -78
44.97296905517578 -78 44.96173095703125</gml:posList>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:Polygon>
              </Boundary>
            </Lakes>
          </wfs:member>
        </wfs:Tuple>
      </wfs:member>
      <wfs:member>
        <wfs:Tuple>
          <wfs:member xlink:href="#Parks.287796"/>
          <wfs:member>
            <Lakes gml:id="Lakes.287817">
              <Name>Pog Lake</Name>
              <Boundary>
                <gml:Polygon gml:id="GID_18"
                  srsName="urn:ogc:def:crs:EPSG::4326">
                  <gml:exterior>
                    <gml:LinearRing>
                      <gml:posList>-77.97476959228516 44.97800827026367 -77.97828674316406
44.97269821166992 -77.99055480957031 44.9674301147461 -78 44.96173095703125 -78 44.96173095703125 -
78 44.97296905517578 -78 44.97296905517578 -77.99826812744141 44.97361755371094 -77.98856353759766
44.97351837158203 -77.98063659667969 44.97765731811523 -77.97476959228516 44.97800827026367
</gml:posList>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:Polygon>
              </Boundary>
            </Lakes>
```

```
        </wfs:member>
      </wfs:Tuple>
    </wfs:member>
    <wfs:member>
      <wfs:Tuple>
        <wfs:member xlink:href="#Parks.287796"/>
        <wfs:member>
          <Lakes gml:id="Lakes.291062">
            <Name>Rock Lake</Name>
            <Boundary>
              <gml:Polygon gml:id="GID_19"
                srsName="urn:ogc:def:crs:EPSG::4326">
                <gml:exterior>
                  <gml:LinearRing>
                    <gml:posList>-79.13716125488281 45.33119964599609 -79.14396667480469
45.32807922363281 -79.15678405761719 45.32684707641602 -79.16410827636719 45.31824493408203
-79.16771697998047 45.31795501708984 -79.17977142333984 45.3109474182129 -79.18424224853516
45.31184387207031 -79.18424224853516 45.31184387207031 -79.18721008300781 45.31496810913086 -
79.19295501708984 45.31615447998048 -79.19607543945312 45.32111740112305 -79.19607543945312
45.32111740112305 -79.19659423828125 45.32251739501953 -79.18892669677734 45.33133697509766 -
79.1656494140625 45.33644104003906 -79.15970611572266 45.33386993408203 -79.14437866210938
45.33406448364258 -79.13716125488281 45.33119964599609</gml:posList>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </Boundary>
          </Lakes>
        </wfs:member>
      </wfs:Tuple>
    </wfs:member>
  </wfs:FeatureCollection>
```

B.3.15 示例 14

在这个示例中，使用标准连接将 myns:RoadSegments 要素和 myns:Bridges 要素连接在一起，以找到在指定道路上的桥。这个示例假设这两个要素类型都包含 RoadCode 特性。在 myns:RoadSegments 要素中，属性 RodeCode 用来指明名称相同道路的所有部分。在 myns:Bridge 要素中，属性 RoadCode 指明包括桥在内的命名道路。

```
<? xml version="1.0"? >
<GetFeature
  service="WFS"
  version="2.0.0"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                  http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
    < Query typeNames = "myns:RoadSegments myns:Bridges">
      < fes:Filter >
        < fes:And >
          < fes:PropertyIsEqualTo >
            < fes:ValueReference >/myns:RoadSegments/RoadName </fes:ValueReference >
            < fes:Literal > Main St.</fes:Literal >
          </fes:PropertyIsEqualTo >
          < fes:PropertyIsEqualTo >
            < fes:ValueReference >/myns:RoadSegments/RoadCode </fes:ValueReference >
            < fes:ValueReference >/myns:Bridges/RoadCode </fes:ValueReference >
          </fes:PropertyIsEqualTo >
        </fes:And >
      </fes:Filter >
    </Query >
  </GetFeature >
```

B.3.16 示例 15

下面的示例是使用带有别名的自关联的 GetFeature 请求寻找与指定的某个指定兴趣区相交的所有路段。

```
  <? xml version = "1.0"? >
  < GetFeature
    service = "WFS"
    version = "2.0.0"
    xmlns = "http://www.opengis.net/wfs/2.0"
    xmlns:myns = "http://www.someserver.com/myns"
    xmlns:fes = "http://www.opengis.net/fes/2.0"
    xmlns:gml = "http://www.opengis.net/gml/3.2"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                  http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
                  http://www.opengis.net/gml/3.2
                  http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    < Query typeNames = "myns:RoadSegments myns:RoadSegments"
        aliases = "RS1 RS2">
      < fes:Filter >
        < fes:And >
          < fes:BBOX >
            < fes:ValueReference >/RS1/geometry </fes:ValueReference >
            < gml:Envelope srsName = "urn:ogc:def:crs:EPSG::1234">
              < gml:lowerCorner > 10 10 </gml:lowerCorner >
              < gml:upperCorner > 20 20 </gml:upperCorner >
            </gml:Envelope >
          </fes:BBOX >
```

```
        <fes:BBOX>
          <fes:ValueReference>/RS2/geometry</fes:ValueReference>
          <gml:Envelope srsName="urn:ogc:def:crs:EPSG::1234">
            <gml:lowerCorner>10 10</gml:lowerCorner>
            <gml:upperCorner>20 20</gml:upperCorner>
          </gml:Envelope>
        </fes:BBOX>
        <fes:Crosses>
          <fes:ValueReference>/RS1/geometry</fes:ValueReference>
          <fes:ValueReference>/RS2/geometry</fes:ValueReference>
        </fes:Crosses>
      </fes:And>
    </fes:Filter>
  </Query>
</GetFeature>
```

B.3.17 示例 16

下面的 GetFeature 示例用于查找与指定多边形相交的所有道路。

```
<? xml version="1.0" ? >
<GetFeature
  version="2.0.0"
  service="WFS"
  handle="Example Query"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.opengis.net/gml/3.2
              http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  <Query typeNames="myns:Roads" handle="Q01">
    <fes:Filter>
      <fes:Intersects>
        <fes:ValueReference>myns:path</fes:ValueReference>
        <gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326" gml:id="P1">
          <gml:exterior>
            <gml:LinearRing>
              <gml:posList>-19.06099128723145 -169.9416961669922 -19.0565 3190612793 -
169.9346008300781 -19.0523681640625 -169.9278564453125 -19.047290802 00195 -169.9230346679688
-19.03918266296387 -169.9215698242188 -19.0405883789062 5 -169.9138641357422 -19.04656600952148 -
169.9136047363281 -19.05992698669434 -1 69.9196014404297 -19.06432342529297 -169.9275665283203
-19.06826400756836 -169.9 364929199219 -19.06099128723145 -169.9416961669922</gml:posList>
            </gml:LinearRing>
```

```
            </gml:exterior>
          </gml:Polygon>
        </fes:Intersects>
      </fes:Filter>
    </Query>
  </GetFeature>
```

B.3.18 示例 17

本标准定义了 wfs:Query 和 wfs:StoredQuery（见 7.9）两个要素，它们用于 XML 编码的查询表达式中。

这个示例说明如何定义一个其他的要素用于编码可查询表达式，这可能是扩展的一个示例或本标准的一个概述。在该实例中，sql:Query 要素定义允许 SQL 查询在服务器中表现出来。

本示例假设如下：

1) 该服务器用于了解 sql:Query 要素中所包含的各种查询。
2) 新定义的查询表达式要素通过 QueryExpression 限制（见表 14）在服务器的能力描述文档中列举出来。
3) 在查询中使用的表通过 WFS 接口具体化为要素类型。

下面的 XML 片段定义了新的查询要素 sql:Query 的模式：

```
<? xml version = "1.0" encoding = "UTF-8"? >
< xsd:schema
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  targetNamespace = "http://www.someserver.com/sql/1.0"
  xmlns:sql = "http://www.someserver.com/sql/1.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xml = "http://www.w3.org/XML/1998/namespace"
  elementFormDefault = "qualified" version = "2.0.0">
  < xsd:import namespace = "http://www.opengis.net/fes/2.0"
          schemaLocation = "http://schemas.opengis.net/filter/2.0.0/filterAll.xsd"/>
  < xsd:element name = "Query" type = "sql:QueryType"
          substitutionGroup = "fes:AbstractQueryExpression"/>
  < xsd:complexType name = "QueryType">
    < xsd:complexContent >
      < xsd:extension base = "fes:AbstractQueryExpressionType">
        < xsd:sequence >
          < xsd:element name = "Text" type = "xsd:string"/>
        </xsd:sequence >
        < xsd:attribute name = "targetDb" type = "xsd:string" use = "optional"/>
        < xsd:attribute name = "targetDbVer" type = "xsd:string" use = "optional"/>
      </xsd:extension >
    </xsd:complexContent >
  </xsd:complexType >
</xsd:schema >
```

下面的 XML 编码的 GetFeature 请求通过 sql:Query 要素来标识位于洪水范围内的建筑物：

```
<? xml version = "1.0" ? >
< wfs:GetFeature
  service = "WFS"
  version = "2.0.0"
  outputFormat = "application/gml + xml; version = 3.2"
  xmlns:sql = "http://www.someserver.com/sql/1.0"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.someserver.com/sql/1.0 ./SqlQuery.xsd">
  < sql:Query targetDb = "SQLMM">
    < sql:Text >
      SELECT * FROM buildings AS b, rivers AS r
      WHERE b.ground_plot.ST_Within(r.flood_zones) = 1
    </sql:Text >
  </sql:Query >
</wfs:GetFeature >
```

B.3.19 示例 18

下面的示例通过设置 resultType 参数值为"hits"获得了 InWaterA_1M 的要素实例：

```
<? xml version = "1.0" ? >
< GetFeature
  version = "2.0.0"
  service = "WFS"
  resultType = "hits"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
  < Query typeNames = "myns:InWaterA_1M"/>
</GetFeature >
```

该响应表明一共有 339963 个要素。numberReturned 参数值为 0 表明在该响应中没有返回任何要素。

```
<? xml version = "1.0"? >
< wfs:FeatureCollection
  timeStamp = "2010-02-01T22:56:09"
  numberMatched = "339963"
  numberReturned = "0"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd"/>
```

B.3.20 示例 19

虽然 GML 3.2（见 GB/T 23708-2009）是本标准所支持的公认版本，但是其他 GML 也可以使用。下面的 GetFeature 示例包括一个空间谓词，该谓词的几何结构是使用 GML 2.1.2 来编码的。

```
<? xml version = "1.0" ? >
< GetFeature
  version = "2.0.0"
  service = "WFS"
  handle = "Example Query"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:gml = "http://www.opengis.net/gml"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.opengis.net/gml
                http://schemas.opengis.net/gml/2.1.2/geometry.xsd">
  < Query typeNames = "myns:Roads" handle = "Q01">
    < fes:Filter >
      < fes:Intersects >
        < fes:ValueReference > myns:path </fes:ValueReference >
        < gml:Polygon srsName = "urn:ogc:def:crs:EPSG::4326">
          < gml:outerBoundaryIs >
            < gml:LinearRing >
              < gml:coordinates >-19.06099128723145,-169.9416961669922 -19.05653190612793,-
169.9346008300781 -19.0523681640625,-169.9278564453125 -19.04729080200195,-169.9230346679688
-19.03918266296387,-169.9215698242188 -19.04058837890625,-169.9138641357422 -19.04656600952148,-
169.9136047363281 -19.05992698669434,-169.9196014404297 -19.06432342529297,-169.9275665283203
-19.06826400756836,-169.9364929199219 -19.06099128723145,-169.9416961669922 </gml:coordinates >
            </gml:LinearRing >
          </gml:outerBoundaryIs >
        </gml:Polygon >
      </fes:Intersects >
    </fes:Filter >
  </Query >
</GetFeature >
```

B.4 GetPropertyValue 示例

B.4.1 概述

本例假设了 WFS 提供如下虚拟的要素实例集。

```
< myns:Person gml:id = "p4456">
    < gml: identifier codeSpace = " http://www.canadaSIN.com " > 424679374 </gml:
identifier >
```

```
        <myns:lastName>Smith</myns:lastName>
        <myns:firstName>Fred</myns:firstName>
        <myns:age>35</myns:age>
        <myns:sex>Male</myns:sex>
        <myns:spouse xlink:href="#p4467"/>
        <myns:location xlink:href="#p102"/>
        <myns:mailAddress>
          <myns:Address gml:id="a201">
            <myns:streetName>Main St.</myns:streetName>
            <myns:streetNumber>5</myns:streetNumber>
            <myns:city>SomeCity</myns:city>
            <myns:province>Someprovince</myns:province>
            <myns:postalCode>X1X 1X1</myns:postalCode>
            <myns:country>Canada</myns:country>
          </myns:Address>
        </myns:mailAddress>
        <myns:phone>416-123-4567</myns:phone>
        <myns:phone>416-890-1234</myns:phone>
      </myns:Person>
      <myns:Car gml:id="r1432">
        <gml:identifier codeSpace="http://www.carserial.org">51465243</gml:identifier>
        <myns:model>Ford Pinto</myns:model>
        <myns:age>4</myns:age>
        <myns:colour>red</myns:colour>
        <myns:location>
          <gml:Point gml:id="p102">
            <gml:pos>15 15</gml:pos>
          </gml:Point>
        </myns:location>
      </myns:Car>
      <myns:House gml:id="h32">
        <gml:identifier codeSpace="http://www.google.org/houses.xml">654365143</gml:identifier>
        <myns:numFloors>2</myns:numFloors>
        <myns:area uom="sqm">200</myns:area>
        <myns:location>
          <gml:Point gml:id="p101">
            <gml:pos>16 18</gml:pos>
          </gml:Point>
        </myns:location>
        <myns:frontsOn xlink:href="#rs11"/>
        <myns:address xlink:href="#a201"/>
      </myns:House>
```

```
<abc:Road gml:id="rs11">
  <abc:numLanes>3</abc:numLanes>
  <abc:centerline>
    <gml:LineString gml:id="l123">? </gml:LineString>
  </abc:centerline>
</abc:Road>

<abc:RoadNetwork gml:id="rn202">
  <abc:operator>RTA</abc:operator>
  <abc:members xlink:href="#rs11"/>
  <abc:topology>
    <gml:TopoComplex>
      <gml:Edge gml:id="e1">
        <gml:pointProperty xlink:href="#l123"/>
      </gml:Edge>
    </gml:TopoComplex>
  </abc:topology>
</abc:RoadNetwork>
<myns:Person gml:id="p4467">
    <gml:identifier codeSpace="http://www.canadaSIN.com">424679360</gml:identifier>
  <myns:lastName>Smith</myns:lastName>
  <myns:firstName>Mary</myns:firstName>
  <myns:age>31</myns:age>
  <myns:sex>Female</myns:sex>
  <myns:spouse xlink:href="#p4456"/>
  <myns:location xlink:href="#p101"/>
  <myns:mailAddress xlink:href="#a201"/>
  <myns:phone>416-123-4567</myns:phone>
  <myns:phone>416-890-1234</myns:phone>
  <myns:livesIn xlink:href="#h32"/>
  <myns:isDriving xlink:href="r1432"/>
</myns:Person>
```

B.4.2 示例 1

下面的示例通过 GetFeature 请求查找 Fred Smith 的地址。Resolve 和 resolveDepth 参数的使用引导 WFS 解决任何本地引用问题。

```
<? xml version="1.0" ? >
<GetFeature
  service="WFS"
  version="2.0.0"
  resolveDepth="*"
  outputFormat="application/xml; subtype=gml/3.2"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
```

```
      xmlns:myns = "http://www.myserver.com/myns"
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                  http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                  http://www.someserver.com/myns
                  http://www.someserver.com/schemas/Common/SampleSchema.xsd">
      < Query typeNames = "myns:Person">
        < PropertyName resolve = "local"> myns:location </PropertyName >
        < fes:Filter >
          < fes:And >
            < fes:PropertyIsEqualTo >
              < fes:ValueReference > myns:firstName </fes:ValueReference >
              < fes:Literal > Fred </fes:Literal >
            </fes:PropertyIsEqualTo >
            < fes:PropertyIsEqualTo >
              < fes:ValueReference > myns:lastName </fes:ValueReference >
              < fes:Literal > Smith </fes:Literal >
            </fes:PropertyIsEqualTo >
          </fes:And >
        </fes:Filter >
      </Query >
    </GetFeature >
```

响应文档是：

```
    <? xml version = "1.0" encoding = "UTF-8"? >
    < wfs:ValueCollection timeStamp = "2008-09-07T19:00:00" numberReturned = "2"
      numberMatched = "unknown" xmlns = "http://www.someserver.com/myns"
      xmlns:myns = "http://www.someserver.com/myns"
      xmlns:abc = "http://www.someserver.com/abc"
      xmlns:wfs = "http://www.opengis.net/wfs/2.0"
      xmlns:gml = "http://www.opengis.net/gml/3.2"
      xmlns:xlink = "http://www.w3.org/1999/xlink"
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                  http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                  http://www.someserver.com/myns ./myns.xsd
                  http://www.someserver.com/abc ./abc.xsd">
      < wfs:member >
        < myns:Person gml:id = "p4456">
          < gml: identifier codeSpace = " http://www.canadaSIN.com " > 424679374 </gml:
identifier >
          < myns:lastName > Smith </myns:lastName >
          < myns:firstName > Fred </myns:firstName >
          < myns:age > 35 </myns:age >
          < myns:sex > male </myns:sex >
          < myns:spouse xlink:href = " #p4467"/>
```

```
        <myns:location xlink:href = "#pt102"/>
        <myns:mailAddress>
          <myns:Address gml:id = "a201">
            <myns:streetName>Main St.</myns:streetName>
            <myns:streetNumber>5</myns:streetNumber>
            <myns:city>SomeCity</myns:city>
            <myns:province>Someprovince</myns:province>
            <myns:postalCode>X1X 1X1</myns:postalCode>
            <myns:country>Canada</myns:country>
          </myns:Address>
        </myns:mailAddress>
        <myns:phone>416-123-4567</myns:phone>
        <myns:phone>416-890-1234</myns:phone>
      </myns:Person>
    </wfs:member>
    <wfs:additionalValues>
      <wfs:ValueCollection
        timeStamp = "2008-09-07T19:00:00"
        numberReturned = "2"
        numberMatched = "2">
        <wfs:member>
          <myns:Person gml:id = "p4467">
            <gml:identifier codeSpace = "http://www.canadaSIN.com">424679360</gml:iden-
tifier>
            <myns:lastName>Smith</myns:lastName>
            <myns:firstName>Mary</myns:firstName>
            <myns:age>18</myns:age>
            <myns:sex>Female</myns:sex>
            <myns:spouse xlink:href = "#p4456"/>
            <myns:location xlink:href = "#p101"/>
            <myns:mailAddress xlink:href = "#a201"/>
            <myns:phone>416-123-4567</myns:phone>
            <myns:phone>416-890-1234</myns:phone>
            <myns:livesIn xlink:href = "#h32"/>
            <myns:isDriving xlink:href = "r1432"/>
          </myns:Person>
        </wfs:member>
        <wfs:member>
          <myns:Car gml:id = "r1432">
            <gml:identifier codeSpace = "http://www.carserial.org">51465243</gml:iden-
tifier>
            <myns:model>Ford Pinto</myns:model>
            <myns:age>4</myns:age>
            <myns:colour>red</myns:colour>
```

```
            <myns:location>
              <gml:Point gml:id="pt102">
                <gml:pos>-59.603958 -52.106559</gml:pos>
              </gml:Point>
            </myns:location>
          </myns:Car>
        </wfs:member>
      </wfs:ValueCollection>
    </wfs:additionalValues>
  </wfs:ValueCollection>
```

响应文档包含 Fred Smith's myns:Person 要素，wfs:additionalObjects 部分包含解析的本地引用和配偶引用。解析的本地引用标识着 Fred Smith 的地址。

B.4.3 示例 2

如下操作执行与例 1（见 B.4.2）一样的查询，但是使用 GetPropertyValue 操作代替 GetFeature 操作。

```
<? xml version="1.0"? >
<GetPropertyValue
  service="WFS"
  version="2.0.0"
  valueReference="myns:location"
  resolve="local"
  resolveDepth="*"
  outputFormat="application/xml; subtype=gml/3.2"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:myns="http://www.myserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.someserver.com/myns ./myns.xsd">
  <Query typeNames="myns:Person">
    <fes:Filter>
      <fes:And>
        <fes:PropertyIsEqualTo>
          <fes:ValueReference>myns:firstName</fes:ValueReference>
          <fes:Literal>Fred</fes:Literal>
        </fes:PropertyIsEqualTo>
        <fes:PropertyIsEqualTo>
          <fes:ValueReference>myns:lastName</fes:ValueReference>
          <fes:Literal>Smith</fes:Literal>
        </fes:PropertyIsEqualTo>
      </fes:And>
    </fes:Filter>
  </Query>
```

```
</GetPropertyValue>
```

响应文档包含 Fred Smith 的地址：

```
<? xml version = "1.0" ? >
< wfs:ValueCollection
  timeStamp = "2008-09-07T19:00:00"
  numberReturned = "1"
  numberMatched = "unknown"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
  < wfs:member >
    < gml:Point gml:id = "pt102">
      < gml:pos >-59.603958 -52.106559 </gml:pos >
    </gml:Point >
  </wfs:member >
</wfs:ValueCollection >
```

B.4.4 示例 3

下面的 GetFeature 请求通过一个三种要素类型的连接来查找 Mary Smith 所居房屋前面的道路。

```
<? xml version = "1.0" ? >
< GetFeature
  service = "WFS"
  version = "2.0.0"
  outputFormat = "application/xml; subtype = gml/3.2"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:abc = "http://www.myserver.com/abc"
  xmlns:myns = "http://www.myserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.someserver.com/myns ./myns.xsd
                http://www.someserver.com/abc ./abc.xsd">
  < Query typeNames = "myns:Person myns:House abc:Road">
   < PropertyName resolve = "local"> abc:numberOfLanes </PropertyName >
   < fes:Filter >
     < fes:And >
       < fes:PropertyIsEqualTo >
         < fes:ValueReference > myns:firstName </fes:ValueReference >
         < fes:Literal > Mary </fes:Literal >
       </fes:PropertyIsEqualTo >
       < fes:PropertyIsEqualTo >
         < fes:ValueReference > myns:lastName </fes:ValueReference >
```

```
            <fes:Literal>Smith</fes:Literal>
          </fes:PropertyIsEqualTo>
    <fes:PropertyIsEqualTo>
    <fes:ValueReference>myns:Person/valueOf(myns:livesIn)/@gml:id</fes:ValueReference>
    <fes:ValueReference>myns:House/@gml:id</fes:ValueReference>
    </fes:PropertyIsEqualTo>
    <fes:PropertyIsEqualTo>
    <fes:ValueReference>myns:House/valueOf(abc:frontsOn)/@gml:id</fes:ValueReference>
        <fes:ValueReference>abc:Road/@gml:id</fes:ValueReference>
          </fes:PropertyIsEqualTo>
        </fes:And>
       </fes:Filter>
     </Query>
    </GetFeature>
```

该请求的响应是满足连接谓词条件的元组：

```
    <? xml version = "1.0" ? >
    <wfs:FeatureCollection
      timeStamp = "2008-09-07T19:00:00"
      numberReturned = "1"
      numberMatched = "unknown"
      xmlns = "http://www.someserver.com/myns"
      xmlns:myns = "http://www.someserver.com/myns"
      xmlns:abc = "http://www.someserver.com/abc"
      xmlns:wfs = "http://www.opengis.net/wfs/2.0"
      xmlns:gml = "http://www.opengis.net/gml/3.2"
      xmlns:xlink = "http://www.w3.org/1999/xlink"
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                  http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                  http://www.someserver.com/myns ./myns.xsd
                  http://www.someserver.com/abc ./abc.xsd">
        <wfs:member>
      <wfs:Tuple>
        <wfs:member>
          <myns:Person gml:id = "p4467">
            <gml:identifier
            codeSpace = "http://www.canadaSIN.com">424679360</gml:identifier>
            <myns:lastName>Smith</myns:lastName>
            <myns:firstName>Mary</myns:firstName>
            <myns:age>18</myns:age>
            <myns:sex>female</myns:sex>
            <myns:spouse xlink:href = "#p4456"/>
            <myns:location xlink:href = "#pt101" />
            <myns:mailAddress>
```

```
            <myns:Address gml:id="a201">
              <myns:streetName>Main St.</myns:streetName>
              <myns:streetNumber>5</myns:streetNumber>
              <myns:city>SomeCity</myns:city>
              <myns:province>Someprovince</myns:province>
              <myns:postalCode>X1X 1X1</myns:postalCode>
              <myns:country>Canada</myns:country>
            </myns:Address>
          </myns:mailAddress>
          <myns:phone>416-123-4567</myns:phone>
          <myns:phone>416-890-4532</myns:phone>
          <myns:livesIn xlink:href="#h32"/>
          <myns:isDriving xlink:href="#r1432"/>
        </myns:Person>
      </wfs:member>
      <wfs:member>
        <myns:House gml:id="h32">
          <gml:identifier
          codeSpace="http://www.toronto.ca/reg.xml">654365143</gml:identifier>
          <myns:numFloors>2</myns:numFloors>
          <myns:area uom="sqm">200</myns:area>
          <myns:location>
            <gml:Point gml:id="pt101">
              <gml:pos>16 18</gml:pos>
            </gml:Point>
          </myns:location>
          <myns:frontsOn xlink:href="#rs11"/>
          <myns:address xlink:href="#a201"/>
        </myns:House>
      </wfs:member>
      <wfs:member>
        <abc:Road gml:id="rs11">
          <abc:numLanes>3</ab:numLanes>
          <abc:centerLineOf>
            <gml:LineString gml:id="GID_5" srsName="urn:ogc:def:crs:EPSG::4326">
                <gml:posList>-59.478340 -52.226578 -59.484871 -52.223564 -59.488991
-52.198524 -59.485958 -52.169559 -59.480400 -52.152615 -59.465576 -52.141491 -59.462002 -52.136417
-59.447968 -52.127190 -59.422928 -52.120701 -59.411915 -52.117844 -59.397972 -52.116440 -59.371311
-52.121300</gml:posList>
            </gml:LineString>
          </abc:centerLineOf>
        </abc:Road>
      </wfs:member>
    </wfs:Tuple>
```

```
</wfs:member>
</wfs:FeatureCollection>
```

响应文档表明 Mary Smith 家前面的道路有 3 条。

B.4.5 示例 4

下面的 GetPropertyValue 操作和先前的示例(见 B.4.4)回答同样问题。wfs:Query 要素标识着 Mary Smith 的家,valueReference 属性中的路径表达式检索所需要的 abc:numLanes 值。路径表达式中的 valueOf()操作引用了 myns:livesIn 和 myns:frontsOn 引用。

```
<? xml version = "1.0" ? >
< GetPropertyValue
  service = "WFS"
  version = "2.0.0"
  valueReference = "valueOf(myns:livesIn)/valueOf(myns:frontsOn)/abc:numLanes"
  outputFormat = "application/xml; subtype = gml/3.2"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:myns = "http://www.myserver.com/myns"
  xmlns:abc = "http://www.myserver.com/abc"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
               http://www.someserver.com/myns ./myns.xsd
               http://www.someserver.com/abc ./abc.xsd">
  < Query typeNames = "myns:Person">
    < fes:Filter >
      < fes:And >
        < fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:firstName </fes:ValueReference >
          < fes:Literal > Fred </fes:Literal >
        </fes:PropertyIsEqualTo >
        < fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:lastName </fes:ValueReference >
          < fes:Literal > Smith </fes:Literal >
        </fes:PropertyIsEqualTo >
      </fes:And >
    </fes:Filter >
  </Query >
</GetPropertyValue >
```

响应文档:

```
<? xml version = "1.0" ? >
< wfs:ValueCollection
  timeStamp = "2008-09-07T19:00:00"
  numberReturned = "1"
  numberMatched = "1"
```

```
    xmlns:wfs = "http://www.opengis.net/wfs/2.0"
    xmlns:gml = "http://www.opengis.net/gml/3.2"
    xmlns:xlink = "http://www.w3.org/1999/xlink"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
    <wfs:member>3</wfs:member>
  </wfs:ValueCollection>
```

这表明 Mary Smith 家的前面有 3 条道路。

B.4.6 示例 5

下面的 GetFeature 请求使用一个连接来查找每个房屋前面的道路。连接谓词使用 valueOf()操作来解析"frontsOn"引用来获得每个道路相应的 gml:id。

```
<? xml version = "1.0" ? >
<GetFeature
  service = "WFS"
  version = "2.0.0"
  outputFormat = "application/xml; subtype = gml/3.2"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:myns = "http://www.myserver.com/myns"
  xmlns:abc = "http://www.myserver.com/abc"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.someserver.com/myns ./myns.xsd
              http://www.someserver.com/abc ./abc.xsd">
  <Query typeNames = "myns:House abc:Road">
   <fes:Filter>
     <fes:PropertyIsEqualTo>
  <fes:ValueReference>myns:House/valueOf(myns:frontsOn)/@gml:id</fes:
ValueReference>
       <fes:ValueReference>abd:Road/@gml:id</fes:ValueReference>
     </fes:PropertyIsEqualTo>
   </fes:Filter>
  </Query>
</GetFeature>
```

其响应是房屋-道路元组集,它表明房屋前面的道路。

```
<? xml version = "1.0" ? >
<wfs:FeatureCollection
  timeStamp = "2008-09-07T19:00:00"
  numberReturned = "1"
  numberMatched = "unknown"
  xmlns = "http://www.someserver.com/myns"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:abc = "http://www.someserver.com/abc"
```

```
xmlns:wfs = "http://www.opengis.net/wfs/2.0"
xmlns:gml = "http://www.opengis.net/gml/3.2"
xmlns:xlink = "http://www.w3.org/1999/xlink"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.someserver.com/myns ./myns.xsd
                http://www.someserver.com/abc ./abc.xsd">
<wfs:member>
<wfs:Tuple>
  <wfs:member>
    <myns:House gml:id = "h32">
      <gml:identifier
        codeSpace = "http://www.toronto.ca/reg.xml">654365143</gml:identifier>
      <myns:numFloors>2</myns:numFloors>
      <myns:area uom = "sqm">200</myns:area>
      <myns:location>
        <gml:Point gml:id = "pt101">
          <gml:pos>16 18</gml:pos>
        </gml:Point>
      </myns:location>
      <myns:frontsOn xlink:href = "#rs11"/>
      <myns:address xlink:href = "#a201"/>
    </myns:House>
  </wfs:member>
  <wfs:member>
    <abc:Road gml:id = "rs11">
      <abc:numLanes>3</abc:numLanes>
      <abc:centerLineOf>
        <gml:LineString gml:id = "GID_5" srsName = "urn: ogc:def:crs:EPSG::4326">
            <gml:posList>-59.478340 -52.226578 -59.484871 -52.223564 -59.488991
-52.198524 -59.485958 -52.169559 -59.480400 -52.152615 -59.465576 -52.141491 -59.462002 -52.136417
-59.447968 -52.127190 -59.422928 -52.120701 -59.411915 -52.117844 -59.397972 -52.116440 -59.371311
-52.121300</gml:posList>
        </gml:LineString>
      </abc:centerLineOf>
    </abc:Road>
  </wfs:member>
</wfs:Tuple>
</wfs:member>
</wfs:FeatureCollection>
```

B.4.7 示例 6

下面的 GetPropertyValue 请求获取 Mary Smith 的邮政编码，wfs:Query 操作获取 Mary Smith 的 myns:Person 要素和 XPath(参见 W3C XML 路径语言)，其中 XPath 的值是获取邮政编码的 valueRef-

erence 参数的值。当解析 XPath 表达式时，valueof()操作子用于解析任何 myns:livesIn 和 myns:mailAddress 引用。

```
<? xml version = "1.0" ? >
< GetPropertyValue
  service = "WFS"
  version = "2.0.0"
  valueReference = "valueOf(myns:livesIn)/valueof(myns:mailAddress)/myns:postalCode"
  outputFormat = "application/xml; subtype = gml/3.2"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:myns = "http://www.myserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.someserver.com/myns ./myns.xsd">
  < Query typeNames = "myns:Person">
    < fes:Filter >
      < fes:And >
        < fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:firstName </fes:ValueReference >
          < fes:Literal > Mary </fes:Literal >
        </fes:PropertyIsEqualTo >
        < fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:lastName </fes:ValueReference >
          < fes:Literal > Smith </fes:Literal >
        </fes:PropertyIsEqualTo >
      </fes:And >
    </fes:Filter >
  </Query >
</GetPropertyValue >
```

下面的响应文档表明 Mary Smith 的邮政编码是 X1X 1X1：

```
<? xml version = "1.0" ? >
< wfs:ValueCollection
  timeStamp = "2008-09-07T19:00:00"
  numberReturned = "1"
  numberMatched = "unknown"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd ">
  < wfs:member > X1X 1X1 </wfs:member >
</wfs:ValueCollection >
```

B.4.8 示例 7

以下请求可获得 Mary Smith 的年龄：

```
<? xml version = "1.0" ? >
< GetPropertyValue
  service = "WFS"
  version = "2.0.0"
  valueReference = "myns:age"
  outputFormat = "application/xml; subtype = gml/3.2"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:myns = "http://www.myserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.someserver.com/myns ./myns.xsd">
  < Query typeNames = "myns:Person">
    < fes:Filter >
      < fes:And >
        < fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:firstName </fes:ValueReference >
          < fes:Literal > Mary </fes:Literal >
        </fes:PropertyIsEqualTo >
        < fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:lastName </fes:ValueReference >
          < fes:Literal > Smith </fes:Literal >
        </fes:PropertyIsEqualTo >
      </fes:And >
    </fes:Filter >
  </Query >
</GetPropertyValue >
```

下面的响应文档表明 Mary Smith 的年龄是 31：

```
<? xml version = "1.0" ? >
< wfs:ValueCollection
  timeStamp = "2008-09-07T19:00:00"
  numberReturned = "1"
  numberMatched = "unknown"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
            http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
< wfs:member > 31 </wfs:member >
</wfs:ValueCollection >
```

B.4.9 示例 8

以下 GetPropertyValue 请求可获取 Fred Smith 的所有电话号码。查询表达式获取 Fred Smith 的 myns:Person 要素，valueReference Xpath 表达式则从这些要素中获取电话号码。

```
<? xml version = "1.0" ? >
< GetPropertyValue
  service = "WFS"
  version = "2.0.0"
  valueReference = "myns:age"
  outputFormat = "application/xml; subtype = gml/3.2"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:myns = "http://www.myserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.someserver.com/myns ./myns.xsd">
  < Query typeNames = "myns:Person">
    < fes:Filter >
      < fes:And >
        < fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:firstName </fes:ValueReference >
          < fes:Literal > Fred </fes:Literal >
        </fes:PropertyIsEqualTo >
        < fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:lastName </fes:ValueReference >
          < fes:Literal > Smith </fes:Literal >
        </fes:PropertyIsEqualTo >
      </fes:And >
    </fes:Filter >
  </Query >
</GetPropertyValue >
```

包含所有电话号码的响应文档如下：

```
<? xml version = "1.0" ? >
< wfs:ValueCollection
  timeStamp = "2008-09-07T19:00:00"
  numberReturned = "2"
  numberMatched = "2"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  < wfs:member > 416-123-4567 </wfs:member >
  < wfs:member > 416-890-1234 </wfs:member >
</wfs:ValueCollection >
```

B.4.10 示例 9

下面的请求检索 Fred Smith 的第 2 个电话号码。wfs:Query 操作检索 Fred Smith 的 myns:

Person 要素记录，Xpath 表达式 myns:phone[2]从这些记录中获得第 2 个电话号码。

```
<? xml version = "1.0" ? >
< GetPropertyValue
  service = "WFS"
  version = "2.0.0"
  valueReference = "myns:phone[2]"
  outputFormat = "application/xml; subtype = gml/3.2"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:myns = "http://www.myserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.someserver.com/myns ./myns.xsd">
  < Query typeNames = "myns:Person">
    < fes:Filter >
      < fes:And >
        < fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:firstName </fes:ValueReference >
          < fes:Literal > Fred </fes:Literal >
        </fes:PropertyIsEqualTo >
        < fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:lastName </fes:ValueReference >
          < fes:Literal > Smith </fes:Literal >
        </fes:PropertyIsEqualTo >
      </fes:And >
    </fes:Filter >
  </Query >
</GetPropertyValue >
```

包含第 2 个电话号码的响应文档如下：

```
<? xml version = "1.0" ? >
< wfs:ValueCollection
  timeStamp = "2008-09-07T19:00:00"
  numberReturned = "1"
  numberMatched = "1"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = " http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  < wfs:member > 416-890-1234 </wfs:member >
</wfs:ValueCollection >
```

B.5 LockFeature 示例

B.5.1 示例 1

锁定一组枚举的要素。在此例中,WFS 试图锁定尽可能多的要素。

请求:

```
<? xml version = "1.0" ? >
< LockFeature
  version = "2.0.0"
  service = "WFS"
  lockAction = "SOME"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
  < Query typeNames = "myns:InWaterA_1M">
    < fes:Filter >
      < fes:ResourceId rid = "InWaterA_1M.1013"/>
      < fes:ResourceId rid = "InWaterA_1M.1014"/>
      < fes:ResourceId rid = "InWaterA_1M.1015"/>
      < fes:ResourceId rid = "InWaterA_1M.1016"/>
      < fes:ResourceId rid = "InWaterA_1M.1017"/>
    </fes:Filter >
  </Query >
</LockFeature >
```

响应实例:

```
<? xml version = "1.0" ? >
< LockFeatureResponse
  lockId = "1"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
  < FeaturesLocked >
    < fes:ResourceId rid = "InWaterA_1M.1013"/>
    < fes:ResourceId rid = "InWaterA_1M.1014"/>
    < fes:ResourceId rid = "InWaterA_1M.1016"/>
    < fes:ResourceId rid = "InWaterA_1M.1017"/>
  </FeaturesLocked >
  < FeaturesNotLocked >
```

```
    <fes:ResourceId rid = "InWaterA_1M.1015"/>
  </FeaturesNotLocked>
</LockFeatureResponse>
```

B.5.2 示例 2

锁定 InWaterA_1M 类型的所有要素实例。

请求如下：

```
<? xml version = "1.0" ? >
<LockFeature
  version = "2.0.0"
  service = "WFS"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
  <Query typeNames = "myns:InWaterA_1M"/>
</LockFeature>
```

响应实例：

```
<? xml version = "1.0" ? >
<LockFeatureResponse
  lockId = "2"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd"/>
```

B.5.3 示例 3

此例中使用空间约束的 filter 表达式是用来标识将被锁定的一组要素实例。

请求如下：

```
<? xml version = "1.0" ? >
<LockFeature
  version = "2.0.0"
  service = "WFS"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd
                http://www.opengis.net/gml/3.2
                http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  <Query handle = "Lock1" typeNames = "myns:InWaterA_1M">
    <fes:Filter>
```

```
        <fes:Within>
          <fes:ValueReference>myns:wkbGeom</fes:ValueReference>
          <gml:Polygon gml:id="GID_45"
             srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:exterior>
              <gml:LinearRing>
              <gml:posList>-95.7 38.1 -97.8 38.2 -98.9 38.2 -95.7 38.1</gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </fes:Within>
      </fes:Filter>
    </Query>
  </LockFeature>
```

响应实例：

```
<? xml version="1.0" ? >
< LockFeatureResponse
  lockId="A1014375BD"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0
            http://schemas.opengis.net/wfs/2.0.0/wfs.xsd"/>
```

B.5.4 示例 4

此例锁定 BuiltUpA_1M 和 InWaterA_1M 类型的要素，标识为 LOCK1 的锁对所有在定义的窗口中的要素进行锁定；标识为 LOCK2 的锁对 InWaterA_1M.1212、InWaterA_1M.1213 和 InWaterA_1M.10 类型的要素进行锁定。

请求如下：

```
<? xml version="1.0"? >
<LockFeature
  version="2.0.0"
  service="WFS"
  expiry="4"
  lockAction="SOME"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:myns="http://www.someserver.com/myns"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.opengis.net/gml/3.2
              http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  <Query handle="LOCK1" typeNames="myns:BuiltUpA_1M">
    <fes:Filter>
```

```
        <fes:Within>
          <fes:ValueReference>BuiltUpA_1M/wkbGeom</fes:ValueReference>
          <gml:Polygon gml:id="GID_71" srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:exterior>
              <gml:LinearRing>
                <gml:posList>-72.4343981 18.4774721 -72.4344232 18.4775742 -72.4343699
18.477586 -72.4343448 18.4774839 -72.4343981 18.4774721</gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </fes:Within>
      </fes:Filter>
    </Query>
    <Query handle="LOCK2" typeNames="myns:InWaterA_1M">
      <fes:Filter>
        <fes:ResourceId rid="InWaterA_1M.1212"/>
        <fes:ResourceId rid="InWaterA_1M.1213"/>
        <fes:ResourceId rid="InWaterA_1M.10"/>
      </fes:Filter>
    </Query>
  </LockFeature>
```

响应实例：

```
<? xml version="1.0" ? >
< LockFeatureResponse
  lockId="LOCK1A"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd"> lockId="LOCK1A">
  <FeaturesLocked>
    <fes:ResourceId rid="BuiltUpA_1M.1" />
    <fes:ResourceId rid="BuiltUpA_1M.10" />
    <fes:ResourceId rid="BuiltUpA_1M.34" />
    <fes:ResourceId rid="BuiltUpA_1M.786" />
    <fes:ResourceId rid="BuiltUpA_1M.3" />
    <fes:ResourceId rid="BuiltUpA_1M.13" />
    <fes:ResourceId rid="BuiltUpA_1M.47563" />
    <fes:ResourceId rid="InWaterA_1M.1212" />
    <fes:ResourceId rid="InWaterA_1M.1213" />
    <fes:ResourceId rid="InWaterA_1M.10" />
  </FeaturesLocked>
<LockFeatureResponse>
```

B.6 Transaction 示例

B.6.1 Insert 示例

下面的事务创建 InWaterA_1M 要素类型的两个要素实例。

```
<? xml version = "1.0"? >
< wfs:Transaction
  version = "2.0.0"
  service = "WFS"
  xmlns = "http://www.someserver.com/myns"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.someserver.com/myns ./SampleSchema.xsd
                http://www.opengis.net/wfs/2.0
                http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                http://www.opengis.net/gml/3.2
                http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  < wfs:Insert >
  < InWaterA_1M gml:id = "F1">
    < wkbGeom >
     < gml:Polygon srsName = "urn:ogc:def:crs:EPSG::4326" gml:id = "P1">
      < gml:exterior >
        < gml:LinearRing >
          < gml: posList >-30. 93597221374512  117. 6290588378906  -30. 94830513000489
117.6447219848633 -30.95219421386719 117.6465530395508 -30.95219421386719 117.6431121826172 -
30.94802856445312  117. 6386108398438  -30. 94799995422363  117. 6314163208008  -30. 946138381958
117.62850189209 -30.94430541992188 117.6295852661133 -30.93280601501464 117.6240539550781 -
30.92869377136231 117.624641418457 -30.92386054992676 117.6201400756836 -30.92111206054688
117.6206970214844 -30.92458343505859 117.6275863647461 -30.93597221374512 117.6290588378906
</gml:posList >
        </gml:LinearRing >
      </gml:exterior >
     </gml:Polygon >
    </wkbGeom >
    < id > 28022 </id >
    < fCode > BH000 </fCode >
    < hyc > 6 </hyc >
    < tileId > 177 </tileId >
    < facId > 132 </facId >
  </InWaterA_1M >
  < InWaterA_1M gml:id = "F2">
    < wkbGeom >
     < gml:Polygon srsName = "urn:ogc:def:crs:EPSG::4326" gml:id = "P2">
      < gml:exterior >
```

```
            <gml:LinearRing>
                <gml:posList>-30.92013931274414 117.6552810668945 -30.92383384704589
117.661361694336 -30.93005561828613 117.6666412353516 -30.93280601501464 117.6663589477539 -
30.93186187744141 117.6594467163086 -30.93780517578125 117.6541137695312 -30.94397163391114
117.6519470214844 -30.94255638122559 117.6455535888672 -30.93402862548828 117.6336364746094 -
30.92874908447266 117.6355285644531 -30.92138862609864 117.6326370239258 -30.92236137390137
117.6395568847656 -30.91708374023438 117.6433029174805 -30.91711044311523 117.6454467773437 -
30.92061042785645 117.6484985351563 -30.92061042785645 117.6504135131836 -30.91638946533203
117.6504440307617 -30.92013931274414 117.6552810668945</gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </wkbGeom>
      <id>28021</id>
      <fCode>BH000</fCode>
      <hyc>6</hyc>
      <tileId>177</tileId>
      <facId>131</facId>
    </InWaterA_1M>
    </wfs:Insert>
  </wfs:Transaction>
```

此例中，schemaLocation 属性引用一个声明 Inwatera_1M 元素的静态 XML 模式文档(Sample-Schema.xsd)。InWatera_1M 元素的声明可以非常简单地使用 DescribeFeatureType 操作(见第 9 章)进行动态引用。

B.6.2 Update 示例

B.6.2.1 示例 1

下例更新由要素标识符 BuiltUpA_1M.1013 标识的要素特性 population。

```
<? xml version="1.0" ? >
<wfs:Transaction
  version="2.0.0"
  service="WFS"
  xmlns="http://www.someserver.com/myns"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  <wfs:Update typeName="BuiltUpA_1M">
    <wfs:Property>
      <wfs:ValueReference>population</wfs:ValueReference>
      <wfs:Value>4070000</wfs:Value>
    </wfs:Property>
    <fes:Filter>
```

```
        <fes:ResourceId rid = "BuiltUpA_1M.10131"/>
      </fes:Filter>
    </wfs:Update>
  </wfs:Transaction>
```

B.6.2.2 示例 2

更新枚举要素集的 populationType 特性。本例中,标识该要素的是标识符:

BuiltUpA_1M.1013
BuiltUpA_1M.34
BuiltUpA_1M.24256

这些标识符将它们的 *populationType* 设置成值 "CITY"。

```
<? xml version = "1.0" ? >
<wfs:Transaction
  version = "2.0.0"
  service = "WFS"
  xmlns = "http://www.someserver.com/myns"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  <wfs:Update typeName = "BuiltUpA_1M">
    <wfs:Property>
      <wfs:ValueReference>populationType</wfs:ValueReference>
      <wfs:Value>CITY</wfs:Value>
    </wfs:Property>
    <fes:Filter>
      <fes:ResourceId rid = "BuiltUpA_1M.1013"/>
      <fes:ResourceId rid = "BuiltUpA_1M.34"/>
      <fes:ResourceId rid = "BuiltUpA_1M.24256"/>
    </fes:Filter>
  </wfs:Update>
</wfs:Transaction>
```

B.6.2.3 示例 3

更新枚举要素集的 *name* 特性并将 *tileId* 属性值限制的另一个要素集 *FAC_ID* 特性更新为大于 1 000的值。

```
<? xml version = "1.0" ? >
<wfs:Transaction
  version = "2.0.0"
  service = "WFS"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
```

```
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
    <wfs:Update typeNames = "myns:BuiltUpA_1M">
      <wfs:Property>
        <wfs:ValueReference>myns:name</wfs:ValueReference>
        <wfs:Value>somestring</wfs:Value>
      </wfs:Property>
      <fes:Filter>
        <fes:ResourceId rid = "BuiltUpA_1M.1013"/>
        <fes:ResourceId rid = "BuiltUpA_1M.34"/>
        <fes:ResourceId rid = "BuiltUpA_1M.24256"/>
      </fes:Filter>
    </wfs:Update>
    <wfs:Update typeName = "myns:BuiltUpA_1M">
      <wfs:Property>
        <wfs: ValueReference>myns:facId</wfs: ValueReference>
        <wfs:Value>100</wfs:Value>
      </wfs:Property>
      <fes:Filter>
        <fes:PropertyIsGreaterThan>
          <fes:ValueReference>BuiltUpA_1M/tileId</fes:PropertyName>
          <fes:Literal>1000</fes:Literal>
        </fes:PropertyIsGreaterThan>
      </fes:Filter>
    </wfs:Update>
  </wfs:Transaction>
```


B.6.2.4 示例 4

此例更新两个要素类 OceansA_1M 和 TreesA_1MTreesA_1M。OceansA_1M 类的所有深度大于 2 400米的要素被更新，要素 TreesA_1MTreesA_1M.1010 也同样被更新。

```
<? xml version = "1.0" ? >
<wfs:Transaction
  version = "2.0.0"
  service = "WFS"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:myns = "http://www.someserver.com/myns"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  <wfs:Update typeName = "myns:OceansA_1M">
    <wfs:Property>
      <wfs: ValueReference>myns:depth</wfs: ValueReference>
      <wfs:Value>2400</wfs:Value>
    </wfs:Property>
```

```
        <fes:Filter>
          <fes:PropertyIsGreaterThan>
            <fes:ValueReference>OceansA_1M/depth</fes:ValueReference>
            <fes:Literal>2400</fes:Literal>
          </fes:PropertyIsGreaterThan>
        </fes:Filter>
      </wfs:Update>
      <wfs:Update typeName="myns:TreesA_1M">
        <wfs:Property>
          <wfs: ValueReference>myns:treeType</wfs: ValueReference>
          <wfs:Value>CONIFEROUS</wfs:Value>
        </wfs:Property>
        <fes:Filter>
          <fes:ResourceId rid="TreesA_1M.1010"/>
        </fes:Filter>
      </wfs:Update>
    </wfs:Transaction>
```

B.6.3 Delete 示例

B.6.3.1 示例 1

删除单个要素。

```
<? xml version="1.0" ? >
<wfs:Transaction
  version="2.0.0"
  service="WFS"
  xmlns="http://www.someserver.com/myns"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  <wfs:Delete typeName="InWaterA_1M">
    <fes:Filter>
      <fes:ResourceId rid="InWaterA_1M.1013"/>
    </fes:Filter>
  </wfs:Delete>
</wfs:Transaction>
```

B.6.3.2 示例 2

此例删除一组枚举的要素实例。

```
<? xml version="1.0" ? >
<wfs:Transaction
  version="2.0.0"
  service="WFS"
```

```
    xmlns:myns = "http://www.someserver.com/myns"
    xmlns:fes = "http://www.opengis.net/fes/2.0"
    xmlns:wfs = "http://www.opengis.net/wfs/2.0"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
    <wfs:Delete typeName = "myns:InWaterA_1M">
      <fes:Filter>
        <fes:ResourceId rid = "InWaterA_1M.1013"/>
        <fes:ResourceId rid = "InWaterA_1M.10"/>
        <fes:ResourceId rid = "InWaterA_1M.13"/>
        <fes:ResourceId rid = "InWaterA_1M.140"/>
        <fes:ResourceId rid = "InWaterA_1M.5001"/>
        <fes:ResourceId rid = "InWaterA_1M.2001"/>
      </fes:Filter>
    </wfs:Delete>
  </wfs:Transaction>
```

B.6.3.3 示例 3

此例删除要素类型 InWaterA_1M 的一组要素，这组要素位于由谓词中指定的多边形定义的区域内。fes:Filter 元素用于约束操作的范围，而 GML 用于表达多边形的几何形状。

```
  <? xml version = "1.0" ? >
  <wfs:Transaction
    version = "2.0.0"
    service = "WFS"
    xmlns = "http://www.someserver.com/myns"
    xmlns:wfs = "http://www.opengis.net/wfs/2.0"
    xmlns:gml = "http://www.opengis.net/gml/3.2"
    xmlns:fes = "http://www.opengis.net/fes/2.0"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd
                  http://www.opengis.net/gml/3.2
                  http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    <wfs:Delete typeName = "InWaterA_1M">
      <fes:Filter>
        <fes:Within>
          <fes:ValueReference>wkbGeom</fes: ValueReference>
          <gml:Polygon srsName = "urn:ogc:def:crs:EPSG::4326" gml:id = "pp9">
            <gml:exterior>
              <gml:LinearRing>
                < gml: posList >-30.15600013732911 115.0352249145508 -30.17819404602051
115.0252532958984 -30.16072273254395 115.021614074707 -30.1563892364502 115.0222473144531 -
30.15555572509765 115.0259704589844 -30.1512508392334 115.0282211303711 -30.14588928222656
115.0344696044922 -30.15600013732911  115.0352249145508 </gml:posList>
```

```
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </fes:Within>
    </fes:Filter>
  </wfs:Delete>
</wfs:Transaction>
```

B.6.4 混合事务示例

此例定义了一个复杂事务，即“Transaction 01”，实现插入、更新和删除操作，还包括一些复杂特性，此外，XPath 表达式被用作过滤表达式来明确引用所需的特性，此例中包含的评论解析了各种操作。

```
<? xml version = "1.0" ? >
< wfs:Transaction
  version = "2.0.0"
  service = "WFS"
  handle = "Transaction 01"
  xmlns = "http://www.someserver.com/myns"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.someserver.com/myns ./SampleSchema.xsd
              http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.opengis.net/gml/3.2
              http://www.opengis.net/gml/3.2.1/gml.xsd">
  < wfs:Native vendorId = "BigDbCorp" safeToIgnore = "true">
    ALTER SESSION ENABLE PARALLEL DML;
    < bdbc:SomeXmlCommand xmlns:bdbc = "http://www.bigdbcorp.com">
      < bdbc:Command > allow </bdbc:Command >
    </bdbc:SomeXmlCommand >
  </wfs:Native >
  <! —创建 ELEVP_1M 要素类型的一个实例 —>
  < wfs:Insert handle = "Statement 1">
    < ElevP_1M gml:id = "E1">
      < location >
        < gml:Point srsName = "urn:ogc:def:crs:EPSG::4326" gml:id = "e33">
          < gml:pos > 24.2633 -68.5485 </gml:pos >
        </gml:Point >
      </location >
      < id > 167928 </id >
      < fCode > CA030 </fCode >
      < acc > 2 </acc >
      < ela > 1 </ela >
      < zv2 > 29999 </zv2 >
```

```
        <tileId>250</tileId>
        <endId>111</endId>
      </ElevP_1M>
    </wfs:Insert>
    <!--创建 RoadL_1M 要素类型的一个实例,它带有复杂特性:路段描述和道路类型-->
    <wfs:Insert handle="ComplexInsert">
      <RoadL_1M gml:id="R1">
        <name>Highway 401</name>
        <segment>
          <SegmentDesc>
            <designation>SEG_A41</designation>
            <geometry>
              <gml:LineString gml:id="e3"
                srsName="urn:ogc:def:crs:EPSG::4326">
                 <gml:posList>-46.32769393920898 168.9116668701172 -46.31716537475586
168.9369659423828 -46.31291580200195 168.9402465820312 -46.30877685546875 168.9530792236328 -
46.30380630493164 168.9554138183594 -46.30014038085938 168.9626159667969 -46.29613876342773
168.9665222167969 -46.28755569458008  168.9909210205078</gml:posList>
              </gml:LineString>
            </geometry>
          </SegmentDesc>
        </segment>
        <roadType>
          <RoadDesc>
            <surfaceType>Asphalt</surfaceType>
            <nLanes>12</nLanes>
            <grade>15</grade>
          </RoadDesc>
        </roadType>
      </RoadL_1M>
    </wfs:Insert>
    <!--更新若干路段的一个指定范围内的名称,这些路段被压缩为一个单一路段,该过滤器用一
个 XPath 表达式引用名称属性-->
    <wfs:Update typeName="RoadL_1M">
      <wfs:Property>
        <wfs:ValueReference>RoadL_1M/segment/designation</wfs:ValueReference>
        <wfs:Value>SEG_A60</wfs:Value>
      </wfs:Property>
      <fes:Filter>
        <fes:PropertyIsBetween>
          <fes:ValueReference>RoadL_1M/segment/designation</fes:ValueReference>
          <fes:LowerBoundary>
            <fes:Literal>SEG_A60</fes:Literal>
          </fes:LowerBoundary>
```

```
            <fes:UpperBoundary>
              <fes:Literal>SEG_A69</fes:Literal>
            </fes:UpperBoundary>
          </fes:PropertyIsBetween>
        </fes:Filter>
      </wfs:Update>
      <!—创建要素类型 BuiltUpA_1M 的两个要素实例—>?
      <wfs:Insert handle="Statement 2">
        <BuiltUpA_1M gml:id="B1">
          <placeId>4070</placeId>
          <name>Toronto</name>
          <population>4000000</population>
          <bndry>
            <gml:Polygon gml:id="g3"
              srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:exterior>
                <gml:LinearRing>
                  <gml:posList>-32.49074935913086 137.7553100585938 -32.49797058105469
137.7500762939453 -32.50219345092773 137.7501068115234 -32.49538803100586 137.7489929199219 -
32.49352645874023 137.748779296875 -32.49258422851562 137.7469940185547 -32.4914436340332
137.7448272705078 -32.49761199951172 137.7403869628906 -32.49774932861328 137.7377014160156 -
32.48825073242188 137.7359161376953 -32.47955703735352 137.7342987060547 -32.4787483215332
137.7353668212891 -32.47758483886719 137.7368927001953 -32.47419357299805 137.7501678466797 -
32.47983169555664 137.7571411132812 -32.48297119140625 137.759033203125 -32.48714065551758
137.7578887939453 -32.49074935913086 137.7553100585938</gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </bndry>
        </BuiltUpA_1M>
        <BuiltUpA_1M gml:id="B2">
          <placeId>1725</placeId>
          <name>Montreal</name>
          <population>2000000</population>
          <bndry>
            <gml:Polygon gml:id="e4"
              srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:exterior>
                <gml:LinearRing>
                  <gml:posList>-33.20774841308594 138.0020904541016 -33.20772171020508
137.997802734375 -33.20266723632812 137.9964141845703 -33.20236206054688 137.9871368408203 -
33.19541549682617 137.9859924316406 -33.19369506835938 137.9819183349609 -33.17902755737305
137.9871978759766 -33.18127822875977 137.9911346435547 -33.17658233642578 137.9950256347656 -
33.18086242675781 138.0019378662109 -33.18441772460938 138.0037536621094 -33.18713760375977
138.0029144287109 -33.19083404541016 137.9978637695312 -33.19633483886719 137.9975280761719 -
```

```
33.19985961914062 137.9996185302734 -33.19947052001953 138.0031127929688 -33.18966674804688
138.01025390625 -33.19049835205078 138.0133361816406 -33.19108200073242 138.0155792236328 -
33.19660949707031 138.0187530517578 -33.20325088500977 138.0167236328125 -33.20430374145508
138.0137481689453 -33.21133422851562 138.009033203125 -33.20774841308594 138.0041961669922
-33.20774841308594      138.0020904541016 </gml:posList >
                </gml:LinearRing >
              </gml:exterior >
            </gml:Polygon >
          </bndry >
        </BuiltUpA_1M >
        </wfs:Insert >
        <! -- 更新 BuiltUpA_1M.1210 要素实例的名称特性 -->
        < wfs:Update typeName = "BuiltUpA_1M">
          < wfs:Property >
            < wfs:ValueReference > BuiltUpA_1M/name </wfs:ValueReference >
            < wfs:Value > SMALLVILLE </wfs:Value >
          </wfs:Property >
          < fes:Filter >
            < fes:ResourceId rid = "BuiltUpA_1M.1210"/>
          </fes:Filter >
        </wfs:Update >
        <! --更新 BuiltUpA_1M.1725 要素实例的几何特征. -->
        < wfs:Update typeName = "BuiltUpA_1M">
          < wfs:Property >
            < wfs:ValueReference > BuiltUpA_1M/bndry </wfs:ValueReference >
            < wfs:Value >
              < gml:Polygon gml:id = "g5"
                srsName = "urn:ogc:def:crs:EPSG::4326">
                < gml:exterior >
                  < gml:LinearRing >
                    < gml:posList >-34.08438873291016 151.1380767822266 -34.08183288574219
151.1435852050781 -34.08427810668945 151.1475219726562 -34.08483505249023 151.1522216796875 -
34.08349990844727 151.1563110351562 -34.09330368041992 151.1546325683594 -34.08549880981445
151.1357269287109 -34.08438873291016   151.1380767822266 </gml:posList >
                </gml:LinearRing >
              </gml:exterior >
            </gml:Polygon >
          </wfs:Value >
        </wfs:Property >
        < fes:Filter >
          < fes:ResourceId rid = "BuiltUpA_1M.1725"/>
        </fes:Filter >
      </wfs:Update >
      <! -- 删除 BuiltUpA_1M.1013 要素实例 -->
      < wfs:Delete typeName = "BuiltUpA_1M">
```

```
        <fes:Filter>
          <fes:ResourceId rid="BuiltUpA_1M.1013"/>
        </fes:Filter>
      </wfs:Delete>
      <!-- 删除要素类型 BuiltUpA_1M 的所有实例,这些实例位于一个多边形窗口中,且人口数在
100 到 2 000 之间。-->
      <wfs:Delete typeName="BuiltUpA_1M">
        <fes:Filter>
          <fes:And>
            <fes:Within>
              <fes:ValueReference>BuiltUpA_1M/bndry</fes:ValueReference>
              <gml:Polygon gml:id="WINDOW" srsName="urn:ogc:def:crs:EPSG::4326">
                <gml:exterior>
                  <gml:LinearRing>
                    <gml:posList>-33.9866943359375 150.8492279052734 -33.98172378540039
150.8462829589844 -33.97666549682618 150.8461151123047 -33.97611236572266 150.8504791259766 -
33.97283172607422 150.8525543212891 -33.97174835205078 150.8591156005859 -33.97491836547852
150.8614501953125 -33.98274993896484 150.8600769042969 -33.98555374145508 150.857421875
-33.9866943359375 150.8492279052734</gml:posList>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </fes:Within>
            <fes:And>
              <fes:PropertyIsGreaterThanOrEqualTo>
                <fes:ValueReference>BuiltUpA_1M/population</fes:ValueReference>
                <fes:Literal>100</fes:Literal>
              </fes:PropertyIsGreaterThanOrEqualTo>
              <fes:PropertyIsLessThanOrEqualTo>
                <fes:ValueReference>BuiltUpA_1M/population</fes:ValueReference>
                <fes:Literal>2000</fes:Literal>
              </fes:PropertyIsLessThanOrEqualTo>
            </fes:And>
          </fes:And>
        </fes:Filter>
      </wfs:Delete>
      <wfs:Replace>
        <BuiltUpA_1M gml:id="B01a">
          <placeId>34159</placeId>
          <name>MyTown</name>
          <population>14357</population>
          <bndry>
          <gml:Polygon srsName="urn:ogc:def:crs:EPSG::4326" gml:id="P01a">
            <gml:exterior>
```

```
            <gml:LinearRing>
                <gml:posList>-31.121000289917 120.4429473876953 -31.12280464172363
120.444580078125 -31.12605476379395 120.442253112793 -31.12794494628907 120.4380264282227 -
31.12230491638183 120.4315032958984 -31.11816596984863 120.432502746582 -31.11672210693359
120.4361953735352 -31.11802864074707 120.4410247802734 -31.121000289917 120.4429473876953
</gml:posList>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
        </bndry>
      </BuiltUpA_1M>
      <fes:Filter>
        <fes:ResourceId rid="BuiltUpA_1M.45783"/>
      </fes:Filter>
    </wfs:Replace>
  </wfs:Transaction>
```

响应文档：

```
<?xml version="1.0"?>
<wfs:TransactionResponse
  version="2.0.0"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0
          http://schemas.opengis.net/wfs/2.0.0/wfs.xsd">
  <wfs:TransactionSummary>
    <wfs:totalInserted>3</wfs:totalInserted>
    <wfs:totalUpdated>3</wfs:totalUpdated>
    <wfs:totalReplaced>1</wfs:totalReplaced>
    <wfs:totalDeleted>2</wfs:totalDeleted>
  </wfs:TransactionSummary>
  <wfs:InsertResults>
    <wfs:Feature handle="Statement 1">
      <fes:ResourceId rid="ElevP_1M.1001"/>
    </wfs:Feature>
    <wfs:Feature handle="ComplexInsert">
      <fes:ResourceId rid="RoadL_1M.1553"/>
    </wfs:Feature>
    <wfs:Feature handle="Statement 2">
      <fes:ResourceId rid="BuiltUpA_1M.509876"/>
      <fes:ResourceId rid="BuiltUpA_1M.509877"/>
    </wfs:Feature>
  </wfs:InsertResults>
</wfs:TransactionResponse>
```

B.6.5 Transaction 响应示例

假设一个事务请求创建了若干要素类型，该要素实例采用标识为"STMT1"，"STMT2"和"STMT3"的三个< wfs:Insert >元素创建要素实例。此请求的典型响应可能为：

```
<? xml version = "1.0" ? >
< wfs:TransactionResponse
  version = "2.0.0"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = " http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
2.0.0/wfs.xsd">
  < wfs:TransactionSummary >
    < wfs:totalInserted > 5 </wfs:totalInserted >
  </wfs:TransactionSummary >
  < wfs:InsertResults >
    < wfs:Feature handle = "STMT1">
      < fes:ResourceId rid = "SomeFeature.4567"/>
    </wfs:Feature >
    < wfs:Feature handle = "STMT1">
      < fes:ResourceId rid = "SomeFeature.4568"/>
    </wfs:Feature >
    < wfs:Feature handle = "STMT1">
      < fes:ResourceId rid = "SomeFeature.4569"/>
    </wfs:Feature >
    < wfs:Feature handle = "STMT2">
      < fes:ResourceId rid = "Feature1.4569"/>
    </wfs:Feature >
    < wfs:Feature handle = "STMT3">
      < fes:ResourceId rid = "Feature2.389345"/>
    </wfs:Feature >
  </wfs:InsertResults >
</wfs:TransactionResponse >
```

B.7 GetCapabilities 示例

此例说明适用于基础 WFS 的能力描述文档的样式。要请求一个能力描述文档，一个客户端可能发送如下请求：

```
<? xml version = "1.0" ? >
< GetCapabilities
  service = "WFS"
  version = "2.0.0"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = " http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/wfs/
```

```
2.0.0/wfs.xsd"/>
```

实现简单 WFS 一致性类(见表 1)的服务器产生下列能力描述文档:

```
<? xml version = "1.0"? >
< WFS_Capabilities
  version = "2.0.0"
  xmlns = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:fes = "http://www.opengis.net/fes/2.0"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:ows = "http://www.opengis.net/ows/1.1"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
              http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
              http://www.opengis.net/ows/1.1
              http://schemas.opengis.net/ows/1.1.0/owsAll.xsd">
  < ows:ServiceIdentification >
    < ows:Title > OGC Member WFS </ows:Title >
    < ows:Abstract > Web Feature Service maintained by NSDI data provider, serving FGDC
framework layer XXX; contact Paul.Bunyon@BlueOx.org </ows:Abstract >
    < ows:Keywords >
      < ows:Keyword > FGDC </ows:Keyword >
      < ows:Keyword > NSDI </ows:Keyword >
      < ows:Keyword > Framework Data Layer </ows:Keyword >
      < ows:Keyword > BlueOx </ows:Keyword >
      < ows:Type > String </ows:Type >
    </ows:Keywords >
    < ows:ServiceType > WFS </ows:ServiceType >
    < ows:ServiceTypeVersion > 2.0.0 </ows:ServiceTypeVersion >
    < ows:ServiceTypeVersion > 1.1.0 </ows:ServiceTypeVersion >
    < ows:ServiceTypeVersion > 1.0.0 </ows:ServiceTypeVersion >
    < ows:Fees > NONE </ows:Fees >
    < ows:AccessConstraints > NONE </ows:AccessConstraints >
  </ows:ServiceIdentification >
  < ows:ServiceProvider >
    < ows:ProviderName > BlueOx Inc.</ows:ProviderName >
    < ows:ProviderSite xlink:href = "http://www. cubewerx.com"/>
    < ows:ServiceContact >
      < ows:IndividualName > Paul Bunyon </ows:IndividualName >
      < ows:PositionName > Mythology Manager </ows:PositionName >
      < ows:ContactInfo >
        < ows:Phone >
          < ows:Voice > 1.800.BIG.WOOD </ows:Voice >
          < ows:Facsimile > 1.800.FAX.WOOD </ows:Facsimile >
        </ows:Phone >
```

```
            <ows:Address>
              <ows:DeliveryPoint>North Country</ows:DeliveryPoint>
              <ows:City>Small Town</ows:City>
              <ows:AdministrativeArea>Rural County</ows:AdministrativeArea>
              <ows:PostalCode>12345</ows:PostalCode>
              <ows:Country>USA</ows:Country>
              <ows:ElectronicMailAddress>Paul.Bunyon@BlueOx.org</ows:ElectronicMailAd-
dress>
            </ows:Address>
            <ows:OnlineResource xlink:href="http://www.BlueOx.org/contactUs"/>
            <ows:HoursOfService>24x7</ows:HoursOfService>
            <ows:ContactInstructions>
              eMail Paul with normal requsts; Phone Paul for emergency
              requests; if you get voice mail and your request can't wait,
              contact another mythological figure listed on the contactUs
              page of our web site.
            </ows:ContactInstructions>
          </ows:ContactInfo>
          <ows:Role>PointOfContact</ows:Role>
        </ows:ServiceContact>
      </ows:ServiceProvider>
      <ows:OperationsMetadata>
        <ows:Operation name="GetCapabilities">
          <ows:DCP>
            <ows:HTTP>
              <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
              <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
            </ows:HTTP>
          </ows:DCP>
          <ows:Parameter name="AcceptVersions">
            <ows:AllowedValues>
            <ows:Value>2.0.0</ows:Value>
            </ows:AllowedValues>
          </ows:Parameter>
        </ows:Operation>
        <ows:Operation name="DescribeFeatureType">
          <ows:DCP>
            <ows:HTTP>
              <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
              <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
            </ows:HTTP>
          </ows:DCP>
        </ows:Operation>
        <ows:Operation name="ListStoredQueries">
```

```
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="DescribeStoredQueries">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="GetFeature">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Parameter name="version">
    <ows:AllowedValues>
    <ows:Value>2.0.0</ows:Value>
    </ows:AllowedValues>
  </ows:Parameter>
  <!-- ********************************************************* -->
  <!-- *   一致性申明                                            * -->
  <!-- ********************************************************* -->
  <ows:Constraint name="ImplementsBasicWFS">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="ImplementsTransactionalWFS">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="ImplementsLockingWFS">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="KVPEncoding">
```

```
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="XMLEncoding">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="SOAPEncoding">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="ImplementsInheritance">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="ImplementsRemoteResolve">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="ImplementsResultPaging">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="ImplementsStandardJoins">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="ImplementsSpatialJoins">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="ImplementsTemporalJoins">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="ImplementsFeatureVersioning">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <ows:Constraint name="ManageStoredQueries">
    <ows:NoValues/>
    <ows:DefaultValue>FALSE</ows:DefaultValue>
  </ows:Constraint>
  <!-- ************************************************** -->
```

```
        <! - *  能力限定                                                    * ->
        <! - ********************************************************** ->
        < ows:Constraint name = "CountDefault">
          < ows:NoValues/>
          < ows:DefaultValue > 1000 </ows:DefaultValue >
        </ows:Constraint >
        < ows:Constraint name = "QueryExpressions">
          < ows:AllowedValues >
            < ows:Value > wfs:StoredQuery </ows:Value >
          </ows:AllowedValues >
        </ows:Constraint >
        <! - ********************************************************** ->
      </ows:OperationsMetadata >
      < FeatureTypeList >
        < FeatureType xmlns:bo = "http://www.BlueOx.org/BlueOx">
          < Name > bo:Woods </Name >
          < Title > The Great Northern Forest </Title >
          < Abstract >
            Describes the arborial diversity of the Great
            Northern Forest.
          </Abstract >
          < ows:Keywords >
            < ows:Keyword > forest </ows:Keyword >
            < ows:Keyword > north </ows:Keyword >
            < ows:Keyword > woods </ows:Keyword >
            < ows:Keyword > arborial </ows:Keyword >
            < ows:Keyword > diversity </ows:Keyword >
          </ows:Keywords >
          < DefaultCRS > urn:ogc:def:crs:EPSG::6269 </DefaultCRS >
          < ows:WGS84BoundingBox >
            < ows:LowerCorner >-180 -90 </ows:LowerCorner >
            < ows:UpperCorner > 180 90 </ows:UpperCorner >
          </ows:WGS84BoundingBox >
          < MetadataURL
    xlink:href = "http://www.ogccatservice.com/csw.cgi? service = CSW&amp;amp;version = 2.0.0&amp;amp;
request = GetRecords&amp;amp;constraintlanguage = CQL&amp;amp;recordid = urn:uuid:4ee8b2d3-9409-4a1d-
b26b-6782e4fa3d59"/>
          < ExtendedDescription >
            < Element name = "TemporalExtent" type = "xsd:date">
              < ows:Metadata xlink:href = "http://www.someserver.com/AboutTemporalExtent.ht-
ml"/>
              < ValueList >
                < Value > 2000-01-01 </Value >
                < Value > 2006-01-31 </Value >
```

```
            </ValueList>
          </Element>
          <Element name="Classifications" type="xsd:anyURI">
            <ows:Metadata xlink:href="http://www.someserver.com/AboutClassifications.
html"/>
            <ValueList>
              <Value>urn:x-ogc:specification:csw-ebrim:ClassificationScheme:ISO-
    19115:biota</Value>
            </ValueList>
          </Element>
          <Element name="ArborialDiversityIndex" type="xsd:integer">
            <ows:Metadata xlink:href="http://www.someserver.com/ArborialDiversity.
html"/>
            <ValueList>
              <Value>14</Value>
            </ValueList>
          </Element>
        </ExtendedDescription>
      </FeatureType>
    </FeatureTypeList>
    <fes:Filter_Capabilities>
      <fes:Conformance>
        <fes:Constraint name="ImplementsQuery">
          <ows:NoValues/>
          <ows:DefaultValue>TRUE</ows:DefaultValue>
        </fes:Constraint>
        <fes:Constraint name="ImplementsAdHocQuery">
          <ows:NoValues/>
          <ows:DefaultValue>FALSE</ows:DefaultValue>
        </fes:Constraint>
        <fes:Constraint name="ImplementsFunctions">
          <ows:NoValues/>
          <ows:DefaultValue>FALSE</ows:DefaultValue>
        </fes:Constraint>
        <fes:Constraint name="ImplementsMinStandardFilter">
          <ows:NoValues/>
          <ows:DefaultValue>FALSE</ows:DefaultValue>
        </fes:Constraint>
        <fes:Constraint name="ImplementsStandardFilter">
          <ows:NoValues/>
          <ows:DefaultValue>FALSE</ows:DefaultValue>
        </fes:Constraint>
        <fes:Constraint name="ImplementsMinSpatialFilter">
          <ows:NoValues/>
```

```
        <ows:DefaultValue>FALSE</ows:DefaultValue>
      </fes:Constraint>
      <fes:Constraint name="ImplementsSpatialFilter">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
      </fes:Constraint>
      <fes:Constraint name="ImplementsMinTemporalFilter">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
      </fes:Constraint>
      <fes:Constraint name="ImplementsTemporalFilter">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
      </fes:Constraint>
      <fes:Constraint name="ImplementsVersionNav">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
      </fes:Constraint>
      <fes:Constraint name="ImplementsSorting">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
      </fes:Constraint>
      <fes:Constraint name="ImplementsExtendedOperators">
        <ows:NoValues/>
        <ows:DefaultValue>FALSE</ows:DefaultValue>
      </fes:Constraint>
    </fes:Conformance>
  </fes:Filter_Capabilities>
</WFS_Capabilities>
```

实现基础 WFS 一致性类(见表 1)的服务器产生下面的响应文档：

```
<? xml version="1.0"? >
<WFS_Capabilities
  version="2.0.0"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0
                      http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                  http://www.opengis.net/ows/1.1
                  http://schemas.opengis.net/ows/1.1.0/owsAll.xsd">
<ows:ServiceIdentification>
```

```
        <ows:Title>OGC Member WFS</ows:Title>
        <ows:Abstract>Web Feature Service maintained by NSDI data provider, serving FGDC
framework layer XXX; contact Paul.Bunyon@BlueOx.org</ows:Abstract>
        <ows:Keywords>
          <ows:Keyword>FGDC</ows:Keyword>
          <ows:Keyword>NSDI</ows:Keyword>
          <ows:Keyword>Framework Data Layer</ows:Keyword>
          <ows:Keyword>BlueOx</ows:Keyword>
          <ows:Type>String</ows:Type>
        </ows:Keywords>
        <ows:ServiceType>WFS</ows:ServiceType>
        <ows:ServiceTypeVersion>2.0.0</ows:ServiceTypeVersion>
        <ows:ServiceTypeVersion>1.1.0</ows:ServiceTypeVersion>
        <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
        <ows:Fees>NONE</ows:Fees>
        <ows:AccessConstraints>NONE</ows:AccessConstraints>
      </ows:ServiceIdentification>
      <ows:ServiceProvider>
        <ows:ProviderName>BlueOx Inc.</ows:ProviderName>
        <ows:ProviderSite xlink:href="http://www.cubewerx.com"/>
        <ows:ServiceContact>
          <ows:IndividualName>Paul Bunyon</ows:IndividualName>
          <ows:PositionName>Mythology Manager</ows:PositionName>
          <ows:ContactInfo>
            <ows:Phone>
              <ows:Voice>1.800.BIG.WOOD</ows:Voice>
              <ows:Facsimile>1.800.FAX.WOOD</ows:Facsimile>
            </ows:Phone>
            <ows:Address>
              <ows:DeliveryPoint>North Country</ows:DeliveryPoint>
              <ows:City>Small Town</ows:City>
              <ows:AdministrativeArea>Rural County</ows:AdministrativeArea>
              <ows:PostalCode>12345</ows:PostalCode>
              <ows:Country>USA</ows:Country>
      <ows:ElectronicMailAddress>Paul.Bunyon@BlueOx.org</ows:ElectronicMailAddress>
            </ows:Address>
            <ows:OnlineResource xlink:href="http://www.BlueOx.org/contactUs"/>
            <ows:HoursOfService>24x7</ows:HoursOfService>
            <ows:ContactInstructions>
              eMail Paul with normal requsts; Phone Paul for emergency
              requests; if you get voice mail and your request can't wait,
              contact another mythological figure listed on the contactUs
              page of our web site.
            </ows:ContactInstructions>
```

```
      </ows:ContactInfo>
      <ows:Role>PointOfContact</ows:Role>
    </ows:ServiceContact>
  </ows:ServiceProvider>
  <ows:OperationsMetadata>
    <ows:Operation name="GetCapabilities">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        </ows:HTTP>
      </ows:DCP>
      <ows:Parameter name="AcceptVersions">
        <ows:AllowedValues>
        <ows:Value>2.0.0</ows:Value>
        </ows:AllowedValues>
      </ows:Parameter>
    </ows:Operation>
    <ows:Operation name="DescribeFeatureType">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Operation name="ListStoredQueries">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Operation name="DescribeStoredQueries">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Operation name="GetFeature">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        </ows:HTTP>
      </ows:DCP>
```

```
    </ows:Operation>
    <ows:Parameter name="version">
      <ows:AllowedValues>
      <ows:Value>2.0.0</ows:Value>
      </ows:AllowedValues>
    </ows:Parameter>
    <!-- ************************************************** -->
    <!-- *   一致性申明                                     * -->
    <!-- ************************************************** -->
    <ows:Constraint name="ImplementsBasicWFS">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsTransactionalWFS">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsLockingWFS">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="KVPEncoding">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="XMLEncoding">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="SOAPEncoding">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsInheritance">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsRemoteResolve">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsResultPaging">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
```

```
    </ows:Constraint>
    <ows:Constraint name="ImplementsStandardJoins">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsSpatialJoins">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsTemporalJoins">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsFeatureVersioning">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ManageStoredQueries">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </ows:Constraint>
    <!-- ************************************************** -->
    <!-- *   能力限定                                       * -->
    <!-- ************************************************** -->
    <ows:Constraint name="CountDefault">
      <ows:NoValues/>
      <ows:DefaultValue>1000</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="QueryExpressions">
      <ows:AllowedValues>
        <ows:Value>wfs:StoredQuery</ows:Value>
      </ows:AllowedValues>
    </ows:Constraint>
    <!-- ************************************************** -->
  </ows:OperationsMetadata>
  <FeatureTypeList>
    <FeatureType xmlns:bo="http://www.BlueOx.org/BlueOx">
      <Name>bo:Woods</Name>
      <Title>The Great Northern Forest</Title>
      <Abstract>
        Describes the arborial diversity of the Great
        Northern Forest.
      </Abstract>
      <ows:Keywords>
```

```
            <ows:Keyword>forest</ows:Keyword>
            <ows:Keyword>north</ows:Keyword>
            <ows:Keyword>woods</ows:Keyword>
            <ows:Keyword>arborial</ows:Keyword>
            <ows:Keyword>diversity</ows:Keyword>
          </ows:Keywords>
          <DefaultCRS>urn:ogc:def:crs:EPSG::6269</DefaultCRS>
          <OtherCRS>urn:ogc:def:crs:EPSG::32615</OtherCRS>
          <OtherCRS>urn:ogc:def:crs:EPSG::32616</OtherCRS>
          <OtherCRS>urn:ogc:def:crs:EPSG::32617</OtherCRS>
          <OtherCRS>urn:ogc:def:crs:EPSG::32618</OtherCRS>
          <ows:WGS84BoundingBox>
            <ows:LowerCorner>-180 -90</ows:LowerCorner>
            <ows:UpperCorner>180 90</ows:UpperCorner>
          </ows:WGS84BoundingBox>
    <MetadataURLxlink:href = " http://www.ogccatservice.com/csw.cgi? service = CSW&amp;
version = 2.0.0&amp;request = GetRecords&amp;constraintlanguage = CQL&amp;recordid = urn:uuid:
4ee8b2d3-9409-4a1d-b26b-6782e4fa3d59"/>
          <ExtendedDescription>
            <Element name = "TemporalExtent" type = "xsd:date">
    <ows:Metadata xlink:href = "http://www.someserver.com/AboutTemporalExtent.html"/>
              <ValueList>
                <Value>2000-01-01</Value>
                <Value>2006-01-31</Value>
              </ValueList>
            </Element>
            <Element name = "Classifications" type = "xsd:anyURI">
    <ows:Metadata xlink:href = "http://www.someserver.com/AboutClassifications.html"/>
              <ValueList>
                <Value>urn:x-ogc:specification:csw-ebrim:ClassificationScheme:ISO-
    19115:biota</Value>
              </ValueList>
            </Element>
            <Element name = "ArborialDiversityIndex" type = "xsd:integer">
            <ows:Metadata xlink:href = "http://www.someserver.com/ArborialDiversity.html"/>
              <ValueList>
                <Value>14</Value>
              </ValueList>
            </Element>
          </ExtendedDescription>
        </FeatureType>
      </FeatureTypeList>
      <fes:Filter_Capabilities>
          <fes:Conformance>
```

```
<fes:Constraint name="ImplementsQuery">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsAdHocQuery">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsFunctions">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsMinStandardFilter">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsStandardFilter">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsMinSpatialFilter">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsSpatialFilter">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsMinTemporalFilter">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsTemporalFilter">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsVersionNav">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
<fes:Constraint name="ImplementsSorting">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</fes:Constraint>
```

```
        <fes:Constraint name="ImplementsExtendedOperators">
          <ows:NoValues/>
          <ows:DefaultValue>FALSE</ows:DefaultValue>
        </fes:Constraint>
      </fes:Conformance>
      <fes:Id_Capabilities>
        <fes:ResourceIdentifier name="fes:ResourceId"/>
      </fes:Id_Capabilities>
      <fes:Spatial_Capabilities>
        <fes:GeometryOperands>
          <fes:GeometryOperand name="gml:Envelope"/>
        </fes:GeometryOperands>
        <fes:SpatialOperators>
          <fes:SpatialOperator name="BBOX"/>
        </fes:SpatialOperators>
      </fes:Spatial_Capabilities>
    </fes:Filter_Capabilities>
  </WFS_Capabilities>
```

实现所有一致性类(见表 1)的复杂服务器可以产生下面的能力描述文档：

```
<WFS_Capabilities
  version="2.0.0"
  xmlns="http://www.opengis.net/wfs/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0
                      http://schemas.opengis.net/wfs/2.0.0/wfs.xsd
                      http://www.opengis.net/ows/1.1
                      http://schemas.opengis/net/ows/1.1.0/owsAll.xsd">
  <ows:ServiceIdentification>
    <ows:Title>OGC Member WFS</ows:Title>
    <ows:Abstract>Web Feature Service maintained by NSDI data provider, serving FGDC
framework layer XXX; contact Paul.Bunyon@BlueOx.org</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>FGDC</ows:Keyword>
      <ows:Keyword>NSDI</ows:Keyword>
      <ows:Keyword>Framework Data Layer</ows:Keyword>
      <ows:Keyword>BlueOx</ows:Keyword>
      <ows:Type>String</ows:Type>
    </ows:Keywords>
    <ows:ServiceType>WFS</ows:ServiceType>
    <ows:ServiceTypeVersion>2.0.0</ows:ServiceTypeVersion>
```

```
      <ows:ServiceTypeVersion>1.1.0</ows:ServiceTypeVersion>
      <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
      <ows:Fees>NONE</ows:Fees>
      <ows:AccessConstraints>NONE</ows:AccessConstraints>
    </ows:ServiceIdentification>
    <ows:ServiceProvider>
      <ows:ProviderName>BlueOx Inc.</ows:ProviderName>
      <ows:ProviderSite xlink:href="http://www.cubewerx.com"/>
      <ows:ServiceContact>
        <ows:IndividualName>Paul Bunyon</ows:IndividualName>
        <ows:PositionName>Mythology Manager</ows:PositionName>
        <ows:ContactInfo>
          <ows:Phone>
            <ows:Voice>1.800.BIG.WOOD</ows:Voice>
            <ows:Facsimile>1.800.FAX.WOOD</ows:Facsimile>
          </ows:Phone>
          <ows:Address>
            <ows:DeliveryPoint>North Country</ows:DeliveryPoint>
            <ows:City>Small Town</ows:City>
            <ows:AdministrativeArea>Rural County</ows:AdministrativeArea>
            <ows:PostalCode>12345</ows:PostalCode>
            <ows:Country>USA</ows:Country>
            <ows:ElectronicMailAddress>Paul.Bunyon@BlueOx.org</ows:ElectronicMailAd-
dress>
          </ows:Address>
          <ows:OnlineResource xlink:href="http://www.BlueOx.org/contactUs"/>
          <ows:HoursOfService>24x7</ows:HoursOfService>
          <ows:ContactInstructions>eMail Paul with normal requsts; Phone Paul for emer-
gency
            requests; if you get voice mail and your request can't wait,
            contact another mythological figure listed on the contactUs
            page of our web site.</ows:ContactInstructions>
        </ows:ContactInfo>
        <ows:Role>PointOfContact</ows:Role>
      </ows:ServiceContact>
    </ows:ServiceProvider>
    <ows:OperationsMetadata>
      <ows:Operation name="GetCapabilities">
        <ows:DCP>
          <ows:HTTP>
            <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
            <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
          </ows:HTTP>
        </ows:DCP>
```

```
    <ows:Parameter name="AcceptVersions">
      <ows:AllowedValues>
        <ows:Value>2.0.0</ows:Value>
        <ows:Value>1.1.0</ows:Value>
        <ows:Value>1.0.0</ows:Value>
      </ows:AllowedValues>
    </ows:Parameter>
  </ows:Operation>
  <ows:Operation name="DescribeFeatureType">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="ListStoredQueries">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="DescribeStoredQueries">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="GetPropertyValue">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
        <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="GetFeature">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
```

```
        <ows:Post xlink:href = "http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name = "GetFeatureWithLock">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href = "http://www.BlueOx.org/wfs/wfs.cgi?"/>
        <ows:Post xlink:href = "http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name = "Transaction">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href = "http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
    <ows:Constraint name = "AutomaticDataLocking">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name = "PreservesSiblingOrder">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
  </ows:Operation>
  <ows:Operation name = "LockFeature">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href = "http://www.BlueOx.org/wfs/wfs.cgi?"/>
        <ows:Post xlink:href = "http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name = "CreateStoredQuery">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href = "http://www.BlueOx.org/wfs/wfs.cgi?"/>
        <ows:Post xlink:href = "http://www.BlueOx.org/wfs/wfs.cgi"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name = "DropStoredQuery">
```

```
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://www.BlueOx.org/wfs/wfs.cgi?"/>
          <ows:Post xlink:href="http://www.BlueOx.org/wfs/wfs.cgi"/>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Parameter name="version">
      <ows:AllowedValues>
        <ows:Value>2.0.0</ows:Value>
        <ows:Value>1.1.0</ows:Value>
        <ows:Value>1.0.0</ows:Value>
      </ows:AllowedValues>
    </ows:Parameter>
    <!-- ********************************************************* -->
    <!-- *   一致性申明                                            * -->
    <!-- ********************************************************* -->
    <ows:Constraint name="ImplementsBasicWFS">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsTransactionalWFS">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsLockingWFS">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="KVPEncoding">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="XMLEncoding">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="SOAPEncoding">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </ows:Constraint>
    <ows:Constraint name="ImplementsInheritance">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
```

```
</ows:Constraint>
<ows:Constraint name="ImplementsRemoteResolve">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</ows:Constraint>
<ows:Constraint name="ImplementsResultPaging">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</ows:Constraint>
<ows:Constraint name="ImplementsStandardJoins">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</ows:Constraint>
<ows:Constraint name="ImplementsSpatialJoins">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</ows:Constraint>
<ows:Constraint name="ImplementsTemporalJoins">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</ows:Constraint>
<ows:Constraint name="ImplementsFeatureVersioning">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</ows:Constraint>
<ows:Constraint name="ManageStoredQueries">
  <ows:NoValues/>
  <ows:DefaultValue>TRUE</ows:DefaultValue>
</ows:Constraint>
<!-- ************************************************** -->
<!-- *  能力限定                                        * -->
<!-- ************************************************** -->
<ows:Constraint name="PagingIsTransactionSafe">
  <ows:NoValues/>
  <ows:DefaultValue>FALSE</ows:DefaultValue>
</ows:Constraint>
<ows:Constraint name="CountDefault">
  <ows:NoValues/>
  <ows:DefaultValue>1000</ows:DefaultValue>
</ows:Constraint>
<ows:Constraint name="ResolveTimeoutDefault">
  <ows:NoValues/>
  <ows:DefaultValue>300</ows:DefaultValue>
</ows:Constraint>
```

```
        <ows:Constraint name="SortLevelLimit">
          <ows:NoValues/>
          <ows:DefaultValue>1</ows:DefaultValue>
        </ows:Constraint>
        <ows:Constraint name="ResolveLocalScope">
          <ows:NoValues/>
          <ows:DefaultValue>*</ows:DefaultValue>
        </ows:Constraint>
        <ows:Constraint name="ResolveRemoteScope">
          <ows:NoValues/>
          <ows:DefaultValue>5</ows:DefaultValue>
        </ows:Constraint>
        <ows:Constraint name="ResponseCacheTimeout">
          <ows:NoValues/>
          <ows:DefaultValue>300</ows:DefaultValue>
        </ows:Constraint>
        <ows:Constraint name="QueryExpressions">
          <ows:AllowedValues>
            <ows:Value>wfs:Query</ows:Value>
            <ows:Value>wfs:StoredQuery</ows:Value>
          </ows:AllowedValues>
        </ows:Constraint>
        <!-- ************************************************** -->
      </ows:OperationsMetadata>
      <FeatureTypeList>
        <FeatureType xmlns:bo="http://www.BlueOx.org/BlueOx">
          <Name>bo:Woods</Name>
          <Title>The Great Northern Forest</Title>
          <Abstract>Describes the arborial diversity of the Great
              Northern Forest.</Abstract>
          <ows:Keywords>
            <ows:Keyword>forest</ows:Keyword>
            <ows:Keyword>north</ows:Keyword>
            <ows:Keyword>woods</ows:Keyword>
            <ows:Keyword>arborial</ows:Keyword>
            <ows:Keyword>diversity</ows:Keyword>
          </ows:Keywords>
          <DefaultCRS>urn:ogc:def:crs:EPSG::6269</DefaultCRS>
          <OtherCRS>urn:ogc:def:crs:EPSG::32615</OtherCRS>
          <OtherCRS>urn:ogc:def:crs:EPSG::32616</OtherCRS>
          <OtherCRS>urn:ogc:def:crs:EPSG::32617</OtherCRS>
          <OtherCRS>urn:ogc:def:crs:EPSG::32618</OtherCRS>
          <ows:WGS84BoundingBox>
            <ows:LowerCorner>-180 -90</ows:LowerCorner>
```

```
            < ows:UpperCorner > 180 90 </ows:UpperCorner >
          </ows:WGS84BoundingBox >
          < MetadataURL
    xlink:href = "http://www.ogccatservice.com/csw.cgi? service = CSW&amp;amp;version = 2.0.0&amp;amp;
request = Get
    Records&amp;amp;constraintlanguage = CQL&amp;amp;recordid = urn:uuid:4ee8b2d3-9409-4a1d-b26b-
    6782e4fa3d59"/>
          < ExtendedDescription >
            < Element name = "TemporalExtent" type = "xsd:date">
              < ows:Metadata
                xlink:href = "http://www.someserver.com/AboutTemporalExtent.html"/>
              < ValueList >
                < Value > 2000-01-01 </Value >
                < Value > 2006-01-31 </Value >
              </ValueList >
            </Element >
            < Element name = "Classifications" type = "xsd:anyURI">
              < ows:Metadata
                xlink:href = "http://www.someserver.com/AboutClassifications.html"/>
              < ValueList >
                < Value > urn:x-ogc:specification:csw-ebrim:ClassificationScheme:ISO-
    19115:biota </Value >
              </ValueList >
            </Element >
            < Element name = "ArborialDiversityIndex" type = "xsd:integer">
              < ows:Metadata
                xlink:href = "http://www.someserver.com/ArborialDiversity.html"/>
              < ValueList >
                < Value > 14 </Value >
              </ValueList >
            </Element >
          </ExtendedDescription >
        </FeatureType >
        < FeatureType xmlns:bo = "http://www.BlueOx.org/BlueOx">
          < Name > bo:Lakes </Name >
          < Title > The Great Northern Lakes </Title >
          < Abstract > Lake boundaries for all lakes in the
              Great Northern Forest.</Abstract >
          < ows:Keywords >
            < ows:Keyword > lakes </ows:Keyword >
            < ows:Keyword > boundaries </ows:Keyword >
            < ows:Keyword > water </ows:Keyword >
            < ows:Keyword > hydro </ows:Keyword >
          </ows:Keywords >
```

```
            <DefaultCRS>urn:ogc:def:crs:EPSG::6269</DefaultCRS>
            <OtherCRS>urn:ogc:def:crs:EPSG::32615</OtherCRS>
            <OtherCRS>urn:ogc:def:crs:EPSG::32616</OtherCRS>
            <OtherCRS>urn:ogc:def:crs:EPSG::32617</OtherCRS>
            <OtherCRS>urn:ogc:def:crs:EPSG::32618</OtherCRS>
            <ows:WGS84BoundingBox>
              <ows:LowerCorner>-180 -90</ows:LowerCorner>
              <ows:UpperCorner>180 90</ows:UpperCorner>
            </ows:WGS84BoundingBox>
            <MetadataURL
    xlink:href="http://www.ogccatservice.com/csw.cgi? service=CSW&amp;version=2.0.0&amp;
request=GetRecords&amp;constraintlanguage=CQL&amp;recordid=urn:uuid:57973502-29cb-4114-
8d64-9939627a0414"/>
            <ExtendedDescription>
              <Element name="TemporalExtent" type="xsd:date">
                <ows:Metadata
                  xlink:href="http://www.someserver.com/AboutTemporalExtent.html"/>
                <ValueList>
                  <Value>2000-01-01</Value>
                  <Value>2006-01-31</Value>
                </ValueList>
              </Element>
            </ExtendedDescription>
          </FeatureType>
        </FeatureTypeList>
        <fes:Filter_Capabilities>
          <fes:Conformance>
            <fes:Constraint name="ImplementsQuery">
              <ows:NoValues/>
              <ows:DefaultValue>TRUE</ows:DefaultValue>
            </fes:Constraint>
            <fes:Constraint name="ImplementsAdHocQuery">
              <ows:NoValues/>
              <ows:DefaultValue>TRUE</ows:DefaultValue>
            </fes:Constraint>
            <fes:Constraint name="ImplementsFunctions">
              <ows:NoValues/>
              <ows:DefaultValue>TRUE</ows:DefaultValue>
            </fes:Constraint>
            <fes:Constraint name="ImplementsMinStandardFilter">
              <ows:NoValues/>
              <ows:DefaultValue>TRUE</ows:DefaultValue>
            </fes:Constraint>
            <fes:Constraint name="ImplementsStandardFilter">
```

```
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </fes:Constraint>
    <fes:Constraint name="ImplementsMinSpatialFilter">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </fes:Constraint>
    <fes:Constraint name="ImplementsSpatialFilter">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </fes:Constraint>
    <fes:Constraint name="ImplementsMinTemporalFilter">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </fes:Constraint>
    <fes:Constraint name="ImplementsTemporalFilter">
      <ows:NoValues/>
      <ows:DefaultValue>TRUE</ows:DefaultValue>
    </fes:Constraint>
    <fes:Constraint name="ImplementsVersionNav">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </fes:Constraint>
    <fes:Constraint name="ImplementsSorting">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </fes:Constraint>
    <fes:Constraint name="ImplementsExtendedOperators">
      <ows:NoValues/>
      <ows:DefaultValue>FALSE</ows:DefaultValue>
    </fes:Constraint>
  </fes:Conformance>
  <fes:Id_Capabilities>
    <fes:ResourceIdentifier name="fes:ResourceId"/>
  </fes:Id_Capabilities>
  <fes:Scalar_Capabilities>
    <fes:LogicalOperators/>
    <fes:ComparisonOperators>
      <fes:ComparisonOperator name="PropertyIsLessThan"/>
      <fes:ComparisonOperator name="PropertyIsGreaterThan"/>
      <fes:ComparisonOperator name="PropertyIsLessThanOrEqualTo"/>
      <fes:ComparisonOperator name="PropertyIsGreaterThanOrEqualTo"/>
      <fes:ComparisonOperator name="PropertyIsEqualTo"/>
      <fes:ComparisonOperator name="PropertyIsNotEqualTo"/>
```

```
        <fes:ComparisonOperator name = "PropertyIsLike"/>
        <fes:ComparisonOperator name = "PropertyIsBetween"/>
        <fes:ComparisonOperator name = "PropertyIsNull"/>
        <fes:ComparisonOperator name = "PropertyIsNil"/>
      </fes:ComparisonOperators>
    </fes:Scalar_Capabilities>
    <fes:Spatial_Capabilities>
      <fes:GeometryOperands>
        <fes:GeometryOperand name = "gml:Point"/>
        <fes:GeometryOperand name = "gml:MultiPoint"/>
        <fes:GeometryOperand name = "gml:LineString"/>
        <fes:GeometryOperand name = "gml:MultiLineString"/>
        <fes:GeometryOperand name = "gml:Curve"/>
        <fes:GeometryOperand name = "gml:MultiCurve"/>
        <fes:GeometryOperand name = "gml:Polygon"/>
        <fes:GeometryOperand name = "gml:MultiPolygon"/>
        <fes:GeometryOperand name = "gml:Surface"/>
        <fes:GeometryOperand name = "gml:MultiSurface"/>
        <fes:GeometryOperand name = "gml:MultiGeometry"/>
        <fes:GeometryOperand name = "gml:Box"/>
        <fes:GeometryOperand name = "gml:Envelope"/>
      </fes:GeometryOperands>
      <fes:SpatialOperators>
        <fes:SpatialOperator name = "BBOX"/>
        <fes:SpatialOperator name = "Equals"/>
        <fes:SpatialOperator name = "Disjoint"/>
        <fes:SpatialOperator name = "Intersects"/>
        <fes:SpatialOperator name = "Touches"/>
        <fes:SpatialOperator name = "Crosses"/>
        <fes:SpatialOperator name = "Within"/>
        <fes:SpatialOperator name = "Contains"/>
        <fes:SpatialOperator name = "Overlaps"/>
        <fes:SpatialOperator name = "Beyond"/>
        <fes:SpatialOperator name = "DWithin"/>
      </fes:SpatialOperators>
    </fes:Spatial_Capabilities>
    <fes:Temporal_Capabilities>
      <fes:TemporalOperands>
        <fes:TemporalOperand name = "gml:validTime"/>
        <fes:TemporalOperand name = "gml:TimeInstant"/>
        <fes:TemporalOperand name = "gml:TimePeriod"/>
        <fes:TemporalOperand name = "gml:timePosition"/>
        <fes:TemporalOperand name = "gml:timeInterval"/>
        <fes:TemporalOperand name = "gml:duration"/>
```

```
        </fes:TemporalOperands>
        <fes:TemporalOperators>
          <fes:TemporalOperator name="After"/>
          <fes:TemporalOperator name="Before"/>
          <fes:TemporalOperator name="Begins"/>
          <fes:TemporalOperator name="BegunBy"/>
          <fes:TemporalOperator name="TContains"/>
          <fes:TemporalOperator name="During"/>
          <fes:TemporalOperator name="TEquals"/>
          <fes:TemporalOperator name="TOverlaps"/>
          <fes:TemporalOperator name="Meets"/>
          <fes:TemporalOperator name="OverlappedBy"/>
          <fes:TemporalOperator name="MetBy"/>
          <fes:TemporalOperator name="EndedBy"/>
        </fes:TemporalOperators>
      </fes:Temporal_Capabilities>
      <fes:Functions>
        <fes:Function name="min">
          <fes:Returns>xsd:double</fes:Returns>
          <fes:Arguments>
            <fes:Argument name="value1">
              <fes:Type>xsd:double</fes:Type>
            </fes:Argument>
            <fes:Argument name="value2">
              <fes:Type>xsd:double</fes:Type>
            </fes:Argument>
          </fes:Arguments>
        </fes:Function>
        <fes:Function name="max">
          <fes:Returns>xsd:double</fes:Returns>
          <fes:Arguments>
            <fes:Argument name="value1">
              <fes:Type>xsd:double</fes:Type>
            </fes:Argument>
            <fes:Argument name="value2">
              <fes:Type>xsd:double</fes:Type>
            </fes:Argument>
          </fes:Arguments>
        </fes:Function>
      </fes:Functions>
    </fes:Filter_Capabilities>
</WFS_Capabilities>
```

B.8 KVP 示例

B.8.1 协定

通常来说,URL 编码在没有按照预想的方式使用时要求用某些字符,例如'&'隔开。(见 IETF RFC 2396)。但出于表达的清晰性,这些字符不用特定符号隔开。

另外,这部分许多示例包含了一个 FILTER 参数,其值为一个由 ISO 19143:2010 第 7 章指定的 XML 编码的过滤器。为保证严格正确,这些示例在根元素 fes:Filter 中包含了命名空间和模式位置信息使得 XML 可能被验证。因此参数:

FILTER=<Filter><Within><PropertyName>InWaterA_1M/wkbGeom<PropertyName>
<gml:Envelope><gml:lowerCorner>10 10</gml:lowerCorner><gml:upperCorner>20
20</gml:upperCorner></gml:Envelope></Within></Filter>

将更清晰地表达为:

```
FILTER=<Filter xmlns="http://www.opengis.net/fes/2.0"
          xmlns:gml="http://www.opengis.net/gml/3.2"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.opengis.net/fes/2.0
                              http://schemas.opengis.net/filter/2.0.0/filter.xsd
                              http://www.opengis.net/gml/3.2
                              http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    <Within><PropertyName>InWaterA_1M/wkbGeom<PropertyName>
    <gml:Envelope><gml:lowerCorner>10 10</lowerCorner>
    <gml:upperCorner>20,20</gml:upperCorner></gml:Envelope>
    </Within></Filter>
```

但是为了避免混淆必要信息,本条中命名空间和模式位置属性标签已从示例中删去。另外,显示的模式位置只是位置样例,正确的模式位置可以替代位置样例。

最后,FILTER 和其他参数值被打断为多行,以清晰地表达。而实际上一个 FILTER 参数值是由单个长字符串组成。

B.8.2 DescribeFeatureType 示例

B.8.2.1 示例 1

下例请求要素类型 TreesA_1M 的模式描述。

```
http://www.someserver.com/wfs.cgi?
    SERVICE=WFS&
    VERSION=2.0.0&
    NAMESPACES=xmlns(myns, http://www.myserver.com/myns)
    REQUEST=DescribeFeatureType&
    TYPENAMES=myns:TreesA_1M
```

B.8.2.2 示例 2

下例请求要素类型 InWaterA_1M 与 BuiltUpA_1M 的模式描述。

```
http://www.someserver.com/wfs.cgi?
    SERVICE=WFS&
```

VERSION = 2.0.0&
NAMESPACES = xmlns(ns1, http://www.someserver.com/ns1), xmlns(ns2, http://someserver.com/ns2)&
REQUEST = DescribeFeatureType&
TYPENAMES = ns1:TreesA_1M, ns2:BuiltUpA_1M

B.8.3 GetPropertyValue 示例

B.8.3.1 概述

本条的示例用于查询如 B.4.1 所描述的虚拟数据库。

B.8.3.2 示例 1

查找 Fred Smith 所居住的城市,通过 GetFeatuyreById 存储的查询(见 7.9.3.6)来检索 Fred Smith 的记录。

http://www.someserver.com/wfs.cgi?
SERVICE = WFS&
VERSION = 2.0.0&
REQUEST = GetPropertyValue&
VALUEREFERENCE = myns:mailAddress/myns:Address/myns:city&
STOREDQUERY_ID = urn:ogc:def:query:OGC-WFS::GetFeatureById& ID = p4456

B.8.3.3 示例 2

查找 Fred Smith 的位置。

http://www.someserver.com/wfs.cgi?
SERVICE = WFS&
VERSION = 2.0.0&
REQUEST = GetPropertyValue&
VALUEREFERENCE = myns:location&
RESOLVE = local&
RESOLVEDEPTH = *&
TYPENAMES = myns:Person&
FILTER = <fes:Filter><fes:And><fes:PropertyIsEqualTo>
<fes:ValueReference>myns:firstName</fes:ValueReference>
<fes:Literal>Fred</fes:Literal></fes:PropertyIsEqualTo>
<fes:PropertyIsEqualTo><fes:ValueReference>myns:lastName</fes:ValueReference>
<fes:Literal>Smith</fes:Literal></fes:PropertyIsEqualTo></fes:And></fes:Filter>

B.8.3.4 示例 3

查找 Mary Smith 住房前面的道路的数目。

http://www.someserver.com/wfs.cgi?
SERVICE = WFS&
VERSION = 2.0.0&
REQUEST = GetPropertyValue&
VALUEREFERENCE = valueOf(myns:livesIn)/valueOf(myns:frontsOn)/abc:numLanes&

```
        TYPENAMES = myns:Person&
        FILTER = < fes:Filter >< fes:And >< fes:PropertyIsEqualTo >
          < fes:ValueReference > myns:firstName </fes:ValueReference >
          < fes:Literal > Fred </fes:Literal ></fes:PropertyIsEqualTo >
             < fes: PropertyIsEqualTo > < fes: ValueReference > myns: lastName </fes:
ValueReference >
          < fes:Literal > Smith </fes:Literal ></fes:PropertyIsEqualTo ></fes:And ></fes:
Filter >
```

B.8.3.5 示例 4

查找 Mary Smith 的邮政编码。

```
http://www.someserver.com/wfs.cgi?
      SERVICE = WFS&
      VERSION = 2.0.0&
      REQUEST = GetPropertyValue&
    VALUEREFERENCE = valueOf(myns:livesIn)/valueof(myns:mailAddress)/myns:postalCode&
      TYPENAMES = myns:Person&
      FILTER = < fes:Filter >< fes:And >< fes:PropertyIsEqualTo >
           < fes:ValueReference > myns:firstName </fes:ValueReference >
           < fes:Literal > Mary </fes:Literal ></fes:PropertyIsEqualTo >
           < fes: PropertyIsEqualTo > < fes: ValueReference > myns: lastName </fes:
ValueReference >
      < fes:Literal > Smith </fes:Literal ></fes:PropertyIsEqualTo ></fes:And ></fes:
Filter >
```

B.8.3.6 示例 5

查询 Fred Smith 的年龄。

```
http://www.someserver.com/wfs.cgi?
      SERVICE = WFS&
      VERSION = 2.0.0&
      REQUEST = GetPropertyValue&
      VALUEREFERENCE = myns:age&
      TYPENAMES = myns:Person&
      FILTER = < fes:Filter >< fes:And >< fes:PropertyIsEqualTo >
      < fes:ValueReference > myns:firstName </fes:ValueReference >
        < fes: Literal > Fred </fes: Literal > </fes: PropertyIsEqualTo > < fes:
PropertyIsEqualTo >
      < fes:ValueReference > myns:lastName </fes:ValueReference >
      < fes:Literal > Smith </fes:Literal ></fes:PropertyIsEqualTo ></fes:And ></fes:
Filter >
```

B.8.3.7 示例 6

查询 Fred Smith 的电话号码。

```
http://www.someserver.com/wfs.cgi?
      SERVICE = WFS&
```

```
VERSION = 2.0.0&
REQUEST = GetPropertyValue&
VALUEREFERENCE = myns:phone&
TYPENAMES = myns:Person&
FILTER = < fes:Filter >< fes:And >< fes:PropertyIsEqualTo >
    < fes:ValueReference > myns:firstName </fes:ValueReference >
    < fes:Literal > Fred </fes:Literal ></fes:PropertyIsEqualTo >
    < fes: PropertyIsEqualTo > < fes: ValueReference > myns: lastName </fes:
ValueReference >
  < fes:Literal > Smith </fes:Literal ></fes:PropertyIsEqualTo ></fes:And ></fes:
Filter >
```

B.8.3.8 示例 7

查询 Fred Smith 的第二个电话号码。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetPropertyValue&
    VALUEREFERENCE = myns:phone[2]&
    TYPENAMES = myns:Person&
    FILTER = < fes:Filter >< fes:And >< fes:PropertyIsEqualTo >
        < fes:ValueReference > myns:firstName </fes:ValueReference >
        < fes:Literal > Fred </fes:Literal ></fes:PropertyIsEqualTo >
        < fes: PropertyIsEqualTo > < fes: ValueReference > myns: lastName </fes:
ValueReference >
      < fes:Literal > Smith </fes:Literal ></fes:PropertyIsEqualTo ></fes:And ></fes:
Filter >
```

B.8.3.9 示例 8

查询 Mary Smith 住房前面的道路。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetPropertyValue&
    VALUEREFERENCE = valueOf(myns:livesIn)/myns:frontsOn&
    TYPENAMES = myns:Person&
    FILTER = < fes:Filter >< fes:And >< fes:PropertyIsEqualTo >
        < fes:ValueReference > myns:firstName </fes:ValueReference >
        < fes:Literal > Mary </fes:Literal ></fes:PropertyIsEqualTo >
        < fes: PropertyIsEqualTo > < fes: ValueReference > myns: lastName </fes:
ValueReference >
      < fes:Literal > Smith </fes:Literal ></fes:PropertyIsEqualTo ></fes:And ></fes:
Filter >
```

B.8.4 GetFeature 示例

B.8.4.1 示例 1

查询类型 InWaterA_1M 所有实例的所有属性。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    TYPENAMES = InWaterA_1M
```

B.8.4.2 示例 2

查询类型 InWaterA_1M 所有实例的部分属性。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    PROPERTYNAME = InWaterA_1M/wkbGeom,InWaterA_1M/tileId&
    TYPENAMES = InWaterA_1M
```

B.8.4.3 示例 3

查询标识符“InWaterA_1M.1013”标识的要素实例的所有属性。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    RESOURCEID = InWaterA_1M.1013
```


B.8.4.4 示例 4

查询类型 InWaterA_1M 类型的一组枚举的要素集实例。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    RESOURCEID = InWaterA_1M.1013,InWaterA_1M.1014,InWaterA_1M.1015
```

B.8.4.5 示例 5

查询类型 InWaterA_1M 的一组空间约束的要素实例的所有属性。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    TYPENAMES = InWaterA_1M&
    BBOX = 18.54,-72.3544,18.62,-72.2564
```

B.8.4.6 示例 6

查询类型 InWaterA_1M 的一组约束的要素实例的所有属性。结果集是通过一个将空间和非空间谓词相结合的过滤表达式(见 ISO 19143:2010，第 7 章)来识别的。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    PROPERTYNAME = InWaterA_1M/wkbGeom,InWaterA_1M/tileId&
    TYPENAMES = InWaterA_1M&
    FILTER = < fes:Filter >< fes:And >< fes:PropertyIsBetween >
        < fes:ValueReference > InwaterA_1M/tileId </fes:ValueReference >
        < fes:LowerBoundary >< fes:Literal > 100 </fes:Literal >
        </fes:LowerBoundary >< fes:UpperBoundary >
        < fes:Literal > 200 </fes:Literal ></fes:UpperBoundary >
        </fes:PropertyIsBetween >< fes:Intersects >
        < fes:ValueReference > Geometry </fes:ValueReference >
        < gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
        < gml:lowerCorner > 13.0983 31.5899 </gml:lowerCorner >
        < gml:upperCorner > 35.5472 42.8143 </gml:upperCorner >
        </gml:Envelope ></fes:Intersects ></fes:And ></fes:Filter >
```

B.8.4.7 示例 7

查找 Himalayas 区域周围的高度排在第 5 的山。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    TYPENAMES = Mountains&
    BBOX = 27.6669,86.4800,28.2709,87.2120&
    SORTBY = Elevation DESC&
    STARTINDEX = 5&
    COUNT = 1
```

B.8.4.8 示例 8

查询在不同命名空间下的一组要素类型的所有属性。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
NAMESPACES = xmlns(myns,http://www.someserver.com),xmlns(yourns,http://www.someotherserver.com)
    TYPENAMES = (myns:InWaterA_1M)(yourns:BuiltUpA_1M)
```

B.8.4.9 示例 9

查询一组要素实例的部分属性。本例中，对于要素实例"InWaterA_1M"，选择 wkbGeom 和 tileId

属性。对于要素实例"BuiltUpA_1M",选择所有属性。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    PROPERTY = (InWaterA_1M/wkbGeom,InWaterA_1M/tileId)(BuiltUpA_1M/ * )&
    TYPENAMES = (InWaterA_1M)(BuiltUpA_1M)
```

B.8.4.10 示例 10

查询要素类型 InWaterA_1M 与 BuiltUpA_1M 的所有位于指定范围内的要素实例的所有属性。本示例使用等价的过滤表达式而不是 BBOX 参数。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    TYPENAMES = (InWaterA_1M)(BuiltUpA_1M)&
    FILTER = (< Filter >< Within >< PropertyName > InWaterA_1M/wkbGeom
        < PropertyName >< gml:Envelope >< gml:lowerCorner > 43.5707 -79.5797 </gml:lower-
Corner >
        < gml:upperCorner > 43.8219 -79.2693 </gml:upperCorner ></gml:Envelope >
        </Within ></Filter >)(< Filter >< Within >< PropertyName >
     BuiltUpA_1M/wkbGeom < PropertyName >< gml:Envelope >< gml:lowerCorner > 43.5705 -
79.5797
        </gml:lowerCorner >< gml:upperCorner > 43.8219 -79.2693 </gml:upperCorner >
        </gml:Envelope ></Within ></Filter >)
```

B.8.4.11 示例 11

本例描述了使用 PROPERTYNAME 组件的 GetFeature 请求,PROPERTYNAME 是例 12 与例 13 进行比较的基础。

请求如下:

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    PROPERTYNAME = uk:Town/gml:name,uk:Town/gml:directedNode&
    TYPENAMES = uk:Town&
    RESOURCEID = t1
```

响应中包含 xlink:href,但是返回结果如下:

```
<? xml version = "1.0" encoding = "UTF-8"? >
< wfs:FeatureCollection
  timeStamp = "2009-02-14T01:27:44"
  numberMatched = "1"
  numberReturned = "1"
  xmlns = "http://www.someserver.com/myns"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
```

```
    xmlns:gml = "http://www.opengis.net/gml/3.2"
    xmlns:xlink = "http://www.w3.org/1999/xlink"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                  http://schema.opengis.net/wfs/2.0.0/wfs.xsd
                  http://www.opengis.net/gml/3.2
                  http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    < wfs:boundedBy >
      < gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
        < gml:lowerCorner > 49.7330 -11.5210 </gml:lowerCorner >
        < gml:upperCorner > 59.4518 -1.1409 </gml:upperCorner >
      </gml:Envelope >
    </wfs:boundedBy >
    < wfs:member >
      < Town gml:id = "t1">
        < gml:name > Bedford </gml:name >
        < gml:directedNode orientation = " + " xlink:href = " # n1"/>
      </Town >
    </wfs:member >
  </wfs:FeatureCollection >
```

B.8.4.12 示例 12

此例描述了例 11 中 GetFeature 请求的 RESOLVEDEPTH 组件与 RESOLVETIMEOUT 组件的使用。请求如下：

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    PROPERTYNAME = uk:Town/gml:name,uk:Town/gml:directedNode&
    RESOLVEDEPTH = 1&
    RESOLVETIMEOUT = 1&
    TYPENAMES = uk:Town&
    RESOURCEID = t1
```

响应中，第一层的 xlink:href 被遍历，如 7.6.4 中所描述，它返回的内容如下：

```
<? xml version = "1.0"? >
< wfs:FeatureCollection
  timeStamp = "2009-02-14T01:27:44"
  numberMatched = "1"
  numberReturned = "1"
  xmlns = "http://www.someserver.com/myns"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:gml = "http://www.opengis.net/gml/3.2"
  xmlns:xlink = "http://www.w3.org/1999/xlink"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                http://schema.opengis.net/wfs/2.0.0/wfs.xsd
```

```
                    http://www.opengis.net/gml/3.2
                    http://schemas.opengis.net/gml/3.2.1/gml.xsd">
  <wfs:boundedBy>
    <gml:Envelope srsName="urn:ogc:def:crs:EPSG::4326">
      <gml:lowerCorner>10 10</gml:lowerCorner>
      <gml:upperCorner>20 20</gml:upperCorner>
    </gml:Envelope>
  </wfs:boundedBy>
  <wfs:member>
    <Town gml:id="t1">
      <gml:name>Bedford</gml:name>
      <gml:directedNode orientation="+" xlink:href="#n1"/>
      <!-- xlink:href="#n1"/> -->
      <gml:directedNode orientation="+">
        <gml:Node gml:id="n1">
          <gml:pointProperty xlink:href="http://www.bedford.town.uk/civilwo
rks/gps.gml#townHall"/>
        </gml:Node>
      </gml:directedNode>
    </Town>
  </wfs:member>
</wfs:FeatureCollection>
```


B.8.4.13 示例 13

本例描述了例 11 中 GetFeature 请求的 RESOLVEDEPTH 与 RESOLVETIMEOUT 组件的使用。

请求如下：

```
http://www.someserver.com/wfs.cgi?
    SERVICE=WFS&
    VERSION=2.0.0&
    REQUEST=GetFeature&
    PROPERTYNAME=uk:Town/gml:name,uk:Town/gml:directedNode&
    RESOLVEDEPTH=2&
    RESOLVETIMEOUT=2&
    TYPENAMES=uk:Town&
    RESOURCEID=t1
```

在响应中，gml:directedNode 属性的第一层与第二层 xlink:hrefs 被遍历，如 7.6.4 所描述，它们返回的内容如下：

```
<?xml version="1.0"?>
<wfs:FeatureCollection
  timeStamp="2009-02-14T01:27:44"
  numberMatched="1"
  numberReturned="1"
  xmlns="http://www.someserver.com/myns"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
```

```
xmlns:gml = "http://www.opengis.net/gml/3.2"
xmlns:xlink = "http://www.w3.org/1999/xlink"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation = "http://www.opengis.net/wfs/2.0
                      http://schema.opengis.net/wfs/2.0.0/wfs.xsd
                      http://www.opengis.net/gml/3.2
                      http://schemas.opengis.net/gml/3.2.1/gml.xsd">
<wfs:boundedBy>
  <gml:Envelope srsName = "urn:ogc:def:crs:EPSG::4326">
    <gml:lowerCorner>10 10</gml:lowerCorner>
    <gml:upperCorner>20 20</gml:upperCorner>
  </gml:Envelope>
</wfs:boundedBy>
<wfs:member>
  <Town gml:id = "t1">
    <gml:name>Bedford</gml:name>
    <gml:directedNode orientation = " + " xlink:href = "#n1"/>
    <! -- xlink:href = "#n1"/> -->
    <gml:directedNode orientation = " + ">
      <gml:Node gml:id = "n1">
        <gml:pointProperty>
          <gml:Point gml:id = "townHall" srsName = "urn:ogc:def:crs:EPSG::4326">
            <gml:pos>47 34</gml:pos>
          </gml:Point>
        </gml:pointProperty>
      </gml:Node>
    </gml:directedNode>
  </Town>
</wfs:member>
</wfs:FeatureCollection>
```

B.8.4.14 示例 14

下面的示例是一组在 myns:Parks 和 myns:Lakes 两个要素类型中的查询编码：

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.2
    NAMESPACES = xmlns(myns,http://www.someserver.com/myns)&
    REQUEST = GetFeature&
    TYPENAMES = myns:Parks,myns:Lakes&
FILTER = <fes:Filter><fes:And><fes:PropertyIsEqualTo><fes:ValueReference>/myns:
Parks</fes:ValueReference><fes:Literal>Algonquin + Park</fes:Literal></fes:Property-
IsEqualTo><fes:Contains><fes:ValueReference>/myns:Parks/geometry</fes:ValueReference>
<fes:ValueReference>/myns:Lakes/geometry</fes:ValueReference></fes:Contains></fes:And
></fes:Filter>
```

B.8.4.15 示例 15

下面的示例是在 myns:Parks 和 myns:Lakes 两个要素类型中的两个单独的指定查询编码：

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.2
    NAMESPACES = xmlns(myns,http://www.someserver.com/myns)&
    REQUEST = GetFeature&
    TYPENAMES = (myns:Parks)(myns:Lakes)&
FILTER = (< fes:Filter >< fes:BBOX >< fes:ValueReference >/RS1/geometry </fes:ValueRefer-
ence >
            < gml:Envelope + srsName = "urn:ogc:def:crs:EPSG::4326">
            < gml:lowerCorner > 49.1874 -123.2778 </gml:lowerCorner >
            < gml:upperCorner > 49.3504 -122.8892 </gml:upperCorner >
            </gml:Envelope ></fes:BBOX >< fes:Filter >)(< fes:Filter >
            < fes:BBOX >< fes:ValueReference >/RS1/geometry </fes:ValueReference >
            < gml:Envelope + srsName = "urn:ogc:def:crs:EPSG::4326">
            < gml:lowerCorner > 44.6113 -63.7058 </gml:lowerCorner >
            < gml:upperCorner > 44.7256 -63.4641 </gml:upperCorner >
            </gml:Envelope ></fes:BBOX >< fes:Filter >)
```

B.8.4.16 示例 16

下面的示例是在一个指定的矩形范围内获取所有的地点。OUTPUTFORMAT 参数设置为 KML 的特殊指定值，允许通过一个支持 KML 的 Earth 浏览器响应可视化。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    TYPENAMES = PLACES&
    BBOX = 18.54,-72.3544,18.62,-72.2564&
    OUTPUTFORMAT = KML
```

B.8.4.17 示例 17

下面的示例通过 GetFeatureById 存储的查询(见 7.9.3.6)获取一个要素。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = GetFeature&
    STOREDQUERY_ID = urn:ogc:def:query:OGC-WFS::GetFeatyreById&
    ID = p4467
```

该查询的响应是一个空要素，没有包含 GetFeature 请求(见 11.3)的响应所定义的正常要素，例如，输出可能是：

```
< myns:Person
  gml:id = "p4467"
  xmlns:myns = "http://www.someserver.com/myns"
```

```
    xmlns:gml = "http://www.opengis.net/gml/3.2"
    xmlns:xlink = "http://www.w3.org/1999/xlink"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.someserver.com/myns ./myns.xsd
                  http://www.opengis.net/gml/3.2
                  http://schemas.opengis.net/gml/3.2.1/gml.xsd">
    <gml:identifier codeSpace = "http://www.canadaSIN.com"> 424679360 </gml:identifier>
    <myns:lastName> Smith </myns:lastName>
    <myns:firstName> Mary </myns:firstName>
    <myns:age> 18 </myns:age>
    <myns:sex> Female </myns:sex>
    <myns:spouse xlink:href = "#p4456"/>
    <myns:location xlink:href = "#p101"/>
    <myns:mailAddress xlink:href = "#a201"/>
    <myns:phone> 416-123-4567 </myns:phone>
    <myns:phone> 416-890-1234 </myns:phone>
    <myns:livesIn xlink:href = "#h32"/>
    <myns:isDriving xlink:href = "r1432"/>
</myns:Person>
```

B.8.5 LockFeature 示例

B.8.5.1 示例 1

下例是锁定 InWaterA_1M 要素类型的所有实例。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = LockFeature&
    TYPENAMES = InWaterA_1M
```

B.8.5.2 示例 2

下例锁定标识符为"RoadL_1M.1013"的要素。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = LockFeature&
    RESOURCEID = RoadL_1M.1013
```

使用 GetFeatureById 的存储的查询(参见 7.9.3.6),该请求的可选编码如下:

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = LockFeature&
  STOREDQUERY_ID = urn:ogc:def:query:OGC-WFS::GetFeatureById&ID = RoadL_1M.1013
```

B.8.5.3 示例 3

下例是锁定 InWaterA_1M 与 BuiltUpA_1M 要素类型的所有要素实例。

```
http://www.someserver.com/wfs.cgi?
    SERVICE = WFS&
    VERSION = 2.0.0&
    REQUEST = LockFeature&
    TYPENAMES = (InWaterA_1M)(BuiltUpA_1M)
```

B.8.5.4 示例 4

下例是锁定 InWaterA_1M 与 BuiltUpA_1M 要素类型的所有位于用户指定的目标区域内的要素实例。只有一个单一的 BBOX 值被指定，该值用于两个要素类型来识别被锁定的要素。

```
http://www.someserver.com/wfs.cgi?
   SERVICE = WFS&
   VERSION = 2.0.0&
   REQUEST = LockFeature&
   LOCKACTION = ALL&
   TYPENAMES = (INWATER_1M)(BuiltUpA_1M)&
   BBOX = 40.9821,23.4948,41.0257,23.5525
```

B.8.5.5 示例 5

除了使用两个不同的 BBOX 值来识别被锁定的要素实例之外，下面的示例和示例 4 一样。http://www.someserver.com/wfs.cgi?

```
   SERVICE = WFS&
   VERSION = 2.0.0&
   REQUEST = LockFeature&
   LOCKACTION = ALL&
   TYPENAMES = (INWATER_1M)(BuiltUpA_1M)&
   BBOX = (40.9821,23.4948,41.0257,23.5525)(40.5874,22.9013,40.6763,22.9974)
```

附 录 C
（资料性附录）
统一 XML 模式

C.1 概述

本附录将本标准中的 XML 片段统一为一个名为 wfs.xsd 的单独文件，wfs.xsd 可和一个 XML 解析器共同使用来检验 WFS 请求。

C.2 wfs.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.opengis.net/wfs/2.0"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  elementFormDefault="qualified" version="2.0.0">
  <!-- ==========================================
     包含和输入(Includes and Imports)
     ======================================== -->
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xsd:import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="http://schemas.opengis.net/xlink/1.0.0/xlinks.xsd"/>
  <xsd:import namespace="http://www.opengis.net/ows/1.1"
    schemaLocation="http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"/>
  <xsd:import namespace="http://www.opengis.net/fes/2.0"
    schemaLocation="../../filter/2.0.0/filterAll.xsd"/>
  <!-- ======================================== -->
  <!-- =   基本请求类型                        = -->
  <!-- ======================================== -->
  <xsd:complexType name="BaseRequestType" abstract="true">
    <xsd:attribute name="service"
               type="xsd:string" use="required" fixed="WFS"/>
    <xsd:attribute name="version" type="xsd:string" use="required"
      fixed="2.0.0"/>
    <xsd:attribute name="handle" type="xsd:string"/>
  </xsd:complexType>
```

```
<!-- ========================================================= -->
<!--  标准请求参数                                             -->
<!-- ========================================================= -->
<xsd:attributeGroup name="StandardPresentationParameters">
  <xsd:attribute name="startIndex"
                 type="xsd:nonNegativeInteger" default="0"/>
  <xsd:attribute name="count" type="xsd:nonNegativeInteger"/>
  <xsd:attribute name="resultType" type="wfs:ResultTypeType"
    default="results"/>
  <xsd:attribute name="outputFormat" type="xsd:string"
    default="application/gml+xml version=3.2"/>
</xsd:attributeGroup>
<xsd:simpleType name="ResultTypeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="results"/>
    <xsd:enumeration value="hits"/>
  </xsd:restriction>
</xsd:simpleType>
<!-- ========================================================= -->
<!--  解析参数                                                 -->
<!-- ========================================================= -->
<xsd:attributeGroup name="StandardResolveParameters">
  <xsd:attribute name="resolve" type="wfs:ResolveValueType" default="none"/>
  <xsd:attribute name="resolveDepth" type="wfs:positiveIntegerWithStar"
    default="*"/>
  <xsd:attribute name="resolveTimeout" type="xsd:positiveInteger"
    default="300"/>
</xsd:attributeGroup>
<xsd:simpleType name="ResolveValueType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="local"/>
    <xsd:enumeration value="remote"/>
    <xsd:enumeration value="all"/>
    <xsd:enumeration value="none"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="positiveIntegerWithStar">
  <xsd:union memberTypes="xsd:positiveInteger wfs:StarStringType"/>
</xsd:simpleType>
<xsd:simpleType name="StarStringType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="*"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
<!-- ================================================================ -->
<!-- = 标准要素输入参数                                               = -->
<!-- ================================================================ -->
<xsd:attributeGroup name="StandardInputParameters">
  <xsd:attribute name="inputFormat" type="xsd:string"
    default="application/gml+xml version=3.2"/>
  <xsd:attribute name="srsName" type="xsd:anyURI"/>
</xsd:attributeGroup>
<!-- ================================================================ -->
<!-- =  响应元数据                                                    = -->
<!-- ================================================================ -->
<xsd:attributeGroup name="StandardResponseParameters">
  <xsd:attribute name="timeStamp" type="xsd:dateTime" use="required"/>
  <xsd:attribute name="numberMatched" type="wfs:nonNegativeIntegerOrUnknown"
    use="required"/>
  <xsd:attribute name="numberReturned" type="xsd:nonNegativeInteger"
    use="required"/>
  <xsd:attribute name="next" type="xsd:anyURI"/>
  <xsd:attribute name="previous" type="xsd:anyURI"/>
</xsd:attributeGroup>
<xsd:simpleType name="nonNegativeIntegerOrUnknown">
  <xsd:union>
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="unknown"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType>
      <xsd:restriction base="xsd:nonNegativeInteger"/>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
<!-- ================================================================ -->
<!-- = 通用要素元数据元素                                             = -->
<!-- ================================================================ -->
<xsd:element name="Title">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute ref="xml:lang" default="en"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="Abstract">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute ref="xml:lang" default="en"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
<!-- ==================================== -->
<!-- =   查询元素                        = -->
<!-- ==================================== -->
<!-- ===ADHOC QUERY ======================== -->
<xsd:element name="Query" type="wfs:QueryType"
  substitutionGroup="fes:AbstractAdhocQueryExpression"/>
<xsd:complexType name="QueryType">
  <xsd:complexContent>
    <xsd:extension base="fes:AbstractAdhocQueryExpressionType">
      <xsd:attribute name="srsName" type="xsd:anyURI"/>
      <xsd:attribute name="featureVersion" type="xsd:string"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- ===存储的查询 =========================== -->
<xsd:element name="StoredQuery" type="wfs:StoredQueryType"
  substitutionGroup="fes:AbstractQueryExpression"/>
<xsd:complexType name="StoredQueryType">
  <xsd:complexContent>
    <xsd:extension base="fes:AbstractQueryExpressionType">
      <xsd:sequence>
        <xsd:element name="Parameter" type="wfs:ParameterType"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:anyURI" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ParameterType" mixed="true">
  <xsd:sequence>
    <xsd:any namespace="##other" processContents="lax" minOccurs="0"
      maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" use="required"/>
</xsd:complexType>
```

```
<!-- ================================================================= -->
<!-- =   GETCAPABILITIES 请求和响应                                    = -->
<!-- ================================================================= -->
<!-- REQUEST -->
<xsd:element name="GetCapabilities" type="wfs:GetCapabilitiesType"/>
<xsd:complexType name="GetCapabilitiesType">
  <xsd:complexContent>
    <xsd:extension base="ows:GetCapabilitiesType">
      <xsd:attribute name="service" type="ows:ServiceType" use="required"
        fixed="WFS"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- RESPONSE -->
<xsd:element name="WFS_Capabilities" type="wfs:WFS_CapabilitiesType"/>
<xsd:complexType name="WFS_CapabilitiesType">
  <xsd:complexContent>
    <xsd:extension base="ows:CapabilitiesBaseType">
      <xsd:sequence>
        <xsd:element name="WSDL" minOccurs="0">
          <xsd:complexType>
            <xsd:complexContent>
              <xsd:restriction base="xsd:anyType">
                <xsd:attributeGroup ref="xlink:simpleLink"/>
              </xsd:restriction>
            </xsd:complexContent>
          </xsd:complexType>
        </xsd:element>
        <xsd:element ref="wfs:FeatureTypeList" minOccurs="0"/>
        <xsd:element ref="fes:Filter_Capabilities" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="FeatureTypeList" type="wfs:FeatureTypeListType"/>
<xsd:complexType name="FeatureTypeListType">
  <xsd:sequence>
    <xsd:element name="FeatureType" type="wfs:FeatureTypeType"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeatureTypeType">
  <xsd:sequence>
    <xsd:element name="Name" type="xsd:QName"/>
```

```
      <xsd:element ref = "wfs:Title" minOccurs = "0" maxOccurs = "unbounded"/>
      <xsd:element ref = "wfs:Abstract" minOccurs = "0" maxOccurs = "unbounded"/>
      <xsd:element ref = "ows:Keywords" minOccurs = "0" maxOccurs = "unbounded"/>
      <xsd:choice>
        <xsd:sequence>
          <xsd:element name = "DefaultCRS" type = "xsd:anyURI"/>
          <xsd:element name = "OtherCRS" type = "xsd:anyURI" minOccurs = "0"
            maxOccurs = "unbounded"/>
        </xsd:sequence>
        <xsd:element name = "NoCRS">
          <xsd:complexType/>
        </xsd:element>
      </xsd:choice>
      <xsd:element name = "OutputFormats" type = "wfs:OutputFormatListType"
        minOccurs = "0"/>
      <xsd:element ref = "ows:WGS84BoundingBox" minOccurs = "0"
        maxOccurs = "unbounded"/>
      <xsd:element name = "MetadataURL" type = "wfs:MetadataURLType"
        minOccurs = "0" maxOccurs = "unbounded"/>
      <xsd:element name = "ExtendedDescription"
        type = "wfs:ExtendedDescriptionType" minOccurs = "0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name = "OutputFormatListType">
    <xsd:sequence maxOccurs = "unbounded">
      <xsd:element name = "Format" type = "xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name = "MetadataURLType">
    <xsd:attributeGroup ref = "xlink:simpleLink"/>
    <xsd:attribute name = "about" type = "xsd:anyURI"/>
  </xsd:complexType>
  <xsd:complexType name = "ExtendedDescriptionType">
    <xsd:sequence>
      <xsd:element ref = "wfs:Element" maxOccurs = "unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name = "Element" type = "wfs:ElementType"/>
  <xsd:complexType name = "ElementType">
    <xsd:sequence>
      <xsd:element ref = "ows:Metadata"/>
      <xsd:element ref = "wfs:ValueList"/>
    </xsd:sequence>
    <xsd:attribute name = "name" type = "xsd:string" use = "required"/>
```

```
  <xsd:attribute name = "type" type = "xsd:QName" use = "required"/>
</xsd:complexType>
<xsd:element name = "ValueList" type = "wfs:ValueListType"/>
<xsd:complexType name = "ValueListType">
  <xsd:sequence maxOccurs = "unbounded">
    <xsd:element ref = "wfs:Value"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name = "Value" type = "xsd:anyType"/>
<! — = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = —>
<! — =   DESCRIBEFEATURETYPE 请求与响应                                    = —>
<! — = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = —>
<! — REQUEST —>
<xsd:element name = "DescribeFeatureType" type = "wfs:DescribeFeatureTypeType"/>
<xsd:complexType name = "DescribeFeatureTypeType">
  <xsd:complexContent>
    <xsd:extension base = "wfs:BaseRequestType">
      <xsd:sequence>
        <xsd:element name = "TypeName" type = "xsd:QName" minOccurs = "0"
          maxOccurs = "unbounded"/>
      </xsd:sequence>
      <xsd:attribute name = "outputFormat" type = "xsd:string"
        default = "application/gml + xml version = 3.2"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<! — RESPONSE —>
<! — = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = —>
<! — For the outputFormat value of ´application/gml + xml; version = 3.2´  —>
<! — a WFS shall generate a valid XML-Schema/GML3 application schema       —>
<! — that describes that requested feature type(s).                        —>
<! — = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = —>
<! — = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = —>
<! — =   GETPROPERTYVALUE 请求与响应                                      = —>
<! — = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = —>
<! — REQUEST —>
<xsd:element name = "GetPropertyValue" type = "wfs:GetPropertyValueType"/>
<xsd:complexType name = "GetPropertyValueType">
  <xsd:complexContent>
    <xsd:extension base = "wfs:BaseRequestType">
      <xsd:sequence>
        <xsd:element ref = "fes:AbstractQueryExpression"/>
      </xsd:sequence>
      <xsd:attribute name = "valueReference" type = "xsd:string"
```

```
            use = "required"/>
          < xsd:attribute name = "resolvePath" type = "xsd:string"/>
          < xsd:attributeGroup ref = "wfs:StandardPresentationParameters"/>
          < xsd:attributeGroup ref = "wfs:StandardResolveParameters"/>
        </xsd:extension >
      </xsd:complexContent >
    </xsd:complexType >
    <! — RESPONSE —>
    < xsd:element name = "ValueCollection" type = "wfs:ValueCollectionType"/>
    < xsd:complexType name = "ValueCollectionType">
      < xsd:sequence >
        < xsd:element ref = "wfs:member" minOccurs = "0" maxOccurs = "unbounded"/>
        < xsd:element ref = "wfs:additionalValues" minOccurs = "0"/>
        < xsd:element ref = "wfs:truncatedResponse" minOccurs = "0"/>
      </xsd:sequence >
      < xsd:attributeGroup ref = "wfs:StandardResponseParameters"/>
    </xsd:complexType >
    < xsd:element name = "member" type = "wfs:MemberPropertyType"/>
    < xsd:complexType name = "MemberPropertyType" mixed = "true">
      < xsd:choice minOccurs = "0">
        < xsd:any processContents = "lax" namespace = "##other"/>
        < xsd:element ref = "wfs:Tuple"/>
        < xsd:element ref = "wfs:SimpleFeatureCollection"/>
      </xsd:choice >
      < xsd:attribute name = "state" type = "wfs:StateValueType"/>
      < xsd:attributeGroup ref = "xlink:simpleLink"/>
    </xsd:complexType >
    < xsd:element name = "Tuple" type = "wfs:TupleType"/>
    < xsd:complexType name = "TupleType">
      < xsd:sequence >
        < xsd:element ref = "wfs:member" minOccurs = "2" maxOccurs = "unbounded"/>
      </xsd:sequence >
    </xsd:complexType >
    < xsd:element name = "additionalValues">
      < xsd:complexType >
        < xsd:choice >
          < xsd:element ref = "wfs:ValueCollection"/>
          < xsd:element ref = "wfs:SimpleFeatureCollection"/>
        </xsd:choice >
      </xsd:complexType >
    </xsd:element >
    < xsd:element name = "truncatedResponse">
      < xsd:complexType >
        < xsd:sequence >
```

```
        <xsd:element ref="ows:ExceptionReport"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:simpleType name="StateValueType">
    <xsd:union>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="valid"/>
          <xsd:enumeration value="superseded"/>
          <xsd:enumeration value="retired"/>
          <xsd:enumeration value="future"/>
        </xsd:restriction>
      </xsd:simpleType>
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:pattern value="other:\w{2,}"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:union>
  </xsd:simpleType>
  <!-- ================================================================== -->
  <!-- =   GETFEATURE 请求与响应                                         = -->
  <!-- ================================================================== -->
  <xsd:element name="GetFeature" type="wfs:GetFeatureType"/>
  <xsd:complexType name="GetFeatureType">
    <xsd:complexContent>
      <xsd:extension base="wfs:BaseRequestType">
        <xsd:sequence>
          <xsd:element ref="fes:AbstractQueryExpression"
            maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="wfs:StandardPresentationParameters"/>
        <xsd:attributeGroup ref="wfs:StandardResolveParameters"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- === GETFEATUREWITHLOCK 请求 ============== -->
  <xsd:element name="GetFeatureWithLock" type="wfs:GetFeatureWithLockType"/>
  <xsd:complexType name="GetFeatureWithLockType">
    <xsd:complexContent>
      <xsd:extension base="wfs:GetFeatureType">
        <xsd:attribute name="expiry" type="xsd:positiveInteger"
          default="300"/>
```

```
        <xsd:attribute name = "lockAction" type = "wfs:AllSomeType"
          default = "ALL"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <! — = = = = PROPERTYNAME (映射子句) = = = = = = = = = = = = = = —>
  <xsd:element name = "PropertyName"
    substitutionGroup = "fes:AbstractProjectionClause">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base = "xsd:QName">
          <xsd:attributeGroup ref = "wfs:StandardResolveParameters"/>
          <xsd:attribute name = "resolvePath" type = "xsd:string"/>
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
  <! — = = = = GETFEATURE 和 GETFEATUREWITHLOCK 响应 = = = = = = = = = = = = = = = = —>
  <xsd:element name = "FeatureCollection" type = "wfs:FeatureCollectionType"
    substitutionGroup = "wfs:SimpleFeatureCollection"/>
  <xsd:complexType name = "FeatureCollectionType">
    <xsd:complexContent>
      <xsd:extension base = "wfs:SimpleFeatureCollectionType">
        <xsd:sequence>
          <xsd:element ref = "wfs:additionalObjects" minOccurs = "0"/>
          <xsd:element ref = "wfs:truncatedResponse" minOccurs = "0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref = "wfs:StandardResponseParameters"/>
        <xsd:attribute name = "lockId" type = "xsd:string"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name = "additionalObjects">
    <xsd:complexType>
      <xsd:choice>
        <xsd:element ref = "wfs:ValueCollection"/>
        <xsd:element ref = "wfs:SimpleFeatureCollection"/>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name = "SimpleFeatureCollection"
          type = "wfs:SimpleFeatureCollectionType"/>
  <xsd:complexType name = "SimpleFeatureCollectionType">
    <xsd:sequence>
```

```
    <xsd:element ref="wfs:boundedBy" minOccurs="0"/>
    <xsd:element ref="wfs:member" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="boundedBy" type="wfs:EnvelopePropertyType"/>
<xsd:complexType name="EnvelopePropertyType">
  <xsd:sequence>
    <xsd:any namespace="##other"/>
  </xsd:sequence>
</xsd:complexType>
<!-- ======================================== -->
<!-- =  列举存储的查询                        = -->
<!-- ======================================== -->
<!-- REQUEST -->
<xsd:element name="ListStoredQueries" type="wfs:ListStoredQueriesType"/>
<xsd:complexType name="ListStoredQueriesType">
  <xsd:complexContent>
    <xsd:extension base="wfs:BaseRequestType"/>
  </xsd:complexContent>
</xsd:complexType>
<!-- RESPONSE -->
<xsd:element name="ListStoredQueriesResponse"
  type="wfs:ListStoredQueriesResponseType"/>
<xsd:complexType name="ListStoredQueriesResponseType">
  <xsd:sequence>
    <xsd:element name="StoredQuery" type="wfs:StoredQueryListItemType"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="StoredQueryListItemType">
  <xsd:sequence>
    <xsd:element ref="wfs:Title" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="ReturnFeatureType" type="xsd:QName"
      maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:anyURI" use="required"/>
</xsd:complexType>
<!-- ======================================== -->
<!-- =  描述存储的查询                        = -->
<!-- ======================================== -->
<!-- REQUEST -->
<xsd:element name="DescribeStoredQueries"
  type="wfs:DescribeStoredQueriesType"/>
<xsd:complexType name="DescribeStoredQueriesType">
```

```
    <xsd:complexContent>
      <xsd:extension base="wfs:BaseRequestType">
        <xsd:sequence>
          <xsd:element name="StoredQueryId" type="xsd:anyURI" minOccurs="0"
            maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- RESPONSE -->
  <xsd:element name="DescribeStoredQueriesResponse"
    type="wfs:DescribeStoredQueriesResponseType"/>
  <xsd:complexType name="DescribeStoredQueriesResponseType">
    <xsd:sequence>
      <xsd:element name="StoredQueryDescription"
        type="wfs:StoredQueryDescriptionType" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="StoredQueryDescriptionType">
    <xsd:sequence>
      <xsd:element ref="wfs:Title" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="wfs:Abstract" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="ows:Metadata" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="Parameter" type="wfs:ParameterExpressionType"
        minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="QueryExpressionText"
              type="wfs:QueryExpressionTextType"
              minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:anyURI" use="required"/>
  </xsd:complexType>
  <!-- ========================================================== -->
  <!-- =   创建存储的查询                                        = -->
  <!-- ========================================================== -->
  <!-- REQUEST -->
  <xsd:element name="CreateStoredQuery" type="wfs:CreateStoredQueryType"/>
  <xsd:complexType name="CreateStoredQueryType">
    <xsd:complexContent>
      <xsd:extension base="wfs:BaseRequestType">
        <xsd:sequence>
          <xsd:element name="StoredQueryDefinition"
            type="wfs:StoredQueryDescriptionType" minOccurs="0"
            maxOccurs="unbounded"/>
```

```
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="ParameterExpressionType">
    <xsd:sequence>
      <xsd:element ref="wfs:Title" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="wfs:Abstract" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="ows:Metadata" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="name" type="xsd:string" use="required"/>
    <xsd:attribute name="type" type="xsd:QName" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="QueryExpressionTextType" mixed="true">
    <xsd:choice>
      <xsd:any namespace="##other" processContents="skip" minOccurs="0"
        maxOccurs="unbounded"/>
      <xsd:any namespace="##targetNamespace" processContents="skip"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:attribute name="returnFeatureTypes"
      type="wfs:ReturnFeatureTypesListType" use="required"/>
    <xsd:attribute name="language" type="xsd:anyURI" use="required"/>
    <xsd:attribute name="isPrivate" type="xsd:boolean" default="false"/>
  </xsd:complexType>
  <xsd:simpleType name="ReturnFeatureTypesListType">
    <xsd:list itemType="xsd:QName"/>
  </xsd:simpleType>
  <!-- RESPONSE -->
  <xsd:element name="CreateStoredQueryResponse"
    type="wfs:CreateStoredQueryResponseType"/>
  <xsd:complexType name="ExecutionStatusType">
    <xsd:attribute name="status" type="xsd:string" fixed="OK"/>
  </xsd:complexType>
  <xsd:complexType name="CreateStoredQueryResponseType">
    <xsd:complexContent>
      <xsd:extension base="wfs:ExecutionStatusType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ========================================================== -->
  <!-- =    终止存储的查询                                        = -->
  <!-- ========================================================== -->
  <!-- REQUEST -->
  <xsd:element name="DropStoredQuery">
```

```
    <xsd:complexType>
      <xsd:complexContent>
        <xsd:extension base="wfs:BaseRequestType">
          <xsd:attribute name="id" type="xsd:anyURI" use="required"/>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:element>
  <!-- RESPONSE -->
  <xsd:element name="DropStoredQueryResponse" type="wfs:ExecutionStatusType"/>
  <!-- ================================================================= -->
  <!-- =   LOCKFEATURE 请求和响应                                        = -->
  <!-- ================================================================= -->
  <!-- REQUEST -->
  <xsd:element name="LockFeature" type="wfs:LockFeatureType"/>
  <xsd:complexType name="LockFeatureType">
    <xsd:complexContent>
      <xsd:extension base="wfs:BaseRequestType">
        <xsd:sequence>
          <xsd:element ref="fes:AbstractQueryExpression"
            maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="lockId" type="xsd:string"/>
        <xsd:attribute name="expiry" type="xsd:positiveInteger"
          default="300"/>
        <xsd:attribute name="lockAction" type="wfs:AllSomeType"
          default="ALL"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:simpleType name="AllSomeType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="ALL"/>
      <xsd:enumeration value="SOME"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- RESPONSE -->
  <xsd:element name="LockFeatureResponse" type="wfs:LockFeatureResponseType"/>
  <xsd:complexType name="LockFeatureResponseType">
    <xsd:sequence>
      <xsd:element name="FeaturesLocked" type="wfs:FeaturesLockedType"
        minOccurs="0"/>
      <xsd:element name="FeaturesNotLocked" type="wfs:FeaturesNotLockedType"
        minOccurs="0"/>
```

```
  </xsd:sequence>
  <xsd:attribute name="lockId" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="FeaturesLockedType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element ref="fes:ResourceId"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FeaturesNotLockedType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element ref="fes:ResourceId"/>
  </xsd:sequence>
</xsd:complexType>
<!-- ========================================================== -->
<!-- =   TRANSACTION 请求与响应                                 = -->
<!-- ========================================================== -->
<!-- REQUEST -->
<xsd:element name="Transaction" type="wfs:TransactionType"/>
<xsd:complexType name="TransactionType">
  <xsd:complexContent>
    <xsd:extension base="wfs:BaseRequestType">
      <xsd:sequence>
        <xsd:sequence minOccurs="0" maxOccurs="unbounded">
          <xsd:element ref="wfs:AbstractTransactionAction"/>
        </xsd:sequence>
      </xsd:sequence>
      <xsd:attribute name="lockId" type="xsd:string"/>
      <xsd:attribute name="releaseAction" type="wfs:AllSomeType"
        default="ALL"/>
      <xsd:attribute name="srsName" type="xsd:anyURI"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="AbstractTransactionAction"
  type="wfs:AbstractTransactionActionType" abstract="true"/>
<xsd:complexType name="AbstractTransactionActionType" abstract="true">
  <xsd:attribute name="handle" type="xsd:string"/>
</xsd:complexType>
<xsd:element name="Insert" type="wfs:InsertType"
  substitutionGroup="wfs:AbstractTransactionAction"/>
<xsd:complexType name="InsertType">
  <xsd:complexContent>
    <xsd:extension base="wfs:AbstractTransactionActionType">
      <xsd:sequence>
```

```
        <xsd:any namespace="##other" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="wfs:StandardInputParameters"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="Update" type="wfs:UpdateType"
  substitutionGroup="wfs:AbstractTransactionAction"/>
<xsd:complexType name="UpdateType">
  <xsd:complexContent>
    <xsd:extension base="wfs:AbstractTransactionActionType">
      <xsd:sequence>
        <xsd:element ref="wfs:Property" maxOccurs="unbounded"/>
        <xsd:element ref="fes:Filter" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="typeName" type="xsd:QName" use="required"/>
      <xsd:attributeGroup ref="wfs:StandardInputParameters"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="Property" type="wfs:PropertyType"/>
<xsd:complexType name="PropertyType">
  <xsd:sequence>
    <xsd:element name="ValueReference">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:string">
            <xsd:attribute name="action" type="wfs:UpdateActionType"
              default="replace"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="Value" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="UpdateActionType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="replace"/>
    <xsd:enumeration value="insertBefore"/>
    <xsd:enumeration value="insertAfter"/>
    <xsd:enumeration value="remove"/>
  </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:element name = "Replace" type = "wfs:ReplaceType"
  substitutionGroup = "wfs:AbstractTransactionAction"/>
<xsd:complexType name = "ReplaceType">
  <xsd:complexContent>
    <xsd:extension base = "wfs:AbstractTransactionActionType">
      <xsd:sequence>
        <xsd:any namespace = "##other"/>
        <xsd:element ref = "fes:Filter"/>
      </xsd:sequence>
      <xsd:attributeGroup ref = "wfs:StandardInputParameters"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name = "Delete" type = "wfs:DeleteType"
  substitutionGroup = "wfs:AbstractTransactionAction"/>
<xsd:complexType name = "DeleteType">
  <xsd:complexContent>
    <xsd:extension base = "wfs:AbstractTransactionActionType">
      <xsd:sequence>
        <xsd:element ref = "fes:Filter"/>
      </xsd:sequence>
      <xsd:attribute name = "typeName" type = "xsd:QName" use = "required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name = "Native" type = "wfs:NativeType"
  substitutionGroup = "wfs:AbstractTransactionAction"/>
<xsd:complexType name = "NativeType" mixed = "true">
  <xsd:complexContent>
    <xsd:extension base = "wfs:AbstractTransactionActionType">
      <xsd:sequence>
        <xsd:any processContents = "lax" namespace = "##other" minOccurs = "0"/>
      </xsd:sequence>
      <xsd:attribute name = "vendorId" type = "xsd:string" use = "required"/>
      <xsd:attribute name = "safeToIgnore" type = "xsd:boolean" use = "required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<! -- REPONSE -->
<xsd:element name = "TransactionResponse" type = "wfs:TransactionResponseType"/>
<xsd:complexType name = "TransactionResponseType">
  <xsd:sequence>
    <xsd:element name = "TransactionSummary"
      type = "wfs:TransactionSummaryType"/>
```

```
        <xsd:element name = "InsertResults" type = "wfs:ActionResultsType"
          minOccurs = "0"/>
        <xsd:element name = "UpdateResults" type = "wfs:ActionResultsType"
          minOccurs = "0"/>
        <xsd:element name = "ReplaceResults" type = "wfs:ActionResultsType"
          minOccurs = "0"/>
      </xsd:sequence>
      <xsd:attribute name = "version" type = "xsd:string" use = "required"
        fixed = "2.0.0"/>
    </xsd:complexType>
    <xsd:complexType name = "TransactionSummaryType">
      <xsd:sequence>
        <xsd:element name = "totalInserted" type = "xsd:nonNegativeInteger"
          minOccurs = "0"/>
        <xsd:element name = "totalUpdated" type = "xsd:nonNegativeInteger"
          minOccurs = "0"/>
        <xsd:element name = "totalReplaced" type = "xsd:nonNegativeInteger"
          minOccurs = "0"/>
        <xsd:element name = "totalDeleted" type = "xsd:nonNegativeInteger"
          minOccurs = "0"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name = "ActionResultsType">
      <xsd:sequence>
        <xsd:element name = "Feature" type = "wfs:CreatedOrModifiedFeatureType"
          maxOccurs = "unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name = "CreatedOrModifiedFeatureType">
      <xsd:sequence maxOccurs = "unbounded">
        <xsd:element ref = "fes:ResourceId"/>
      </xsd:sequence>
      <xsd:attribute name = "handle" type = "xsd:string"/>
    </xsd:complexType>
    <xsd:complexType name = "EmptyType"/>
  </xsd:schema>
```

附 录 D
（规范性附录）
服务绑定

D.1 概述

本标准的实现应遵守在 OGC 06-121r3:2009 的第 11 章所描述的所有 HTTP 请求和响应规则。

D.2 HTTP GET 和 POST 绑定

本标准的实现应支持 HTTP 方法的 GET 与 POST(见 IETF RFC2616)。

表 D.1 展示了 WFS 请求编码与所支持的 HTTP 的 GET 和 POST 方法的对应关系，每个单元格的值规定了消息内容的 MIME 类型，这些消息内容应用于编码/请求方法的组合。值“Not applicable”(不可用)是指，可以支持编码/请求方法的组合，但是 MIME 类型不可用。此外，不支持 XML 编码的请求和 HTTP GET 方法的组合。

表 D.1 请求编码与传输方法

	HTTP GET 方法	HTTP POST 方法	SOAP
XML 编码的请求	不支持	text/xml	text/xml
KVP 编码的请求	不可用	application/x-www-form-urlencoded	不支持

当使用 HTTP POST 方法时，XML 编码的 WFS 请求的内容类型应设置为 *text/xml*。

当使用 HTTP POST 方法时，KVP 编码的 WFS 请求的内容类型要设置为 *application/x-www-form-urlencoded*，并且文档的内容与一个 HTTP GET 请求的查询字符串相同，即内容应是 GET 请求编码的 URL 中‘?’号后面的字符串。当然内容应用保护特殊字符编码(见 OGC 06-121r3:2009,11.3)。

当使用 HTTP GET 方法和 KVP 编码的 WFS 请求时，由于整个请求在 URL 中编码为‘?’字符之后的名一值对，所以 MIME 类型不适合。

D.3 HTTP 状态编码

服务器应在响应中设置 HTTP 状态编码(见 IETF RFC 2616:1999,6.1.1)。

如果成功处理了 WFS 操作，那么服务器应设置 HTTP 状态编码为“200”，响应消息设置为“OK”。对于相应的 WFS 操作，响应消息的主体应按照本标准所定义的 WFS 操作进行编码。

在响应 WFS 操作时，发生异常的服务器应生成如 7.5 所描述的响应，并且包含如表 D.2 中定义的 HTTP 状态编码。

表 D.2 显示了 OWS(见 OGC 06－121r2:2009,8.3)和 WFS 异常编码(见表 3)与 HTTP 状态编码(见 IETF RFC 2662:1999, 6.1.1)之间的对应关系。

表 D.2 OWS 和 WFS 异常编码与 HTTP 状态编码之间的对应关系

OGC 异常编码	HTTP 状态编码	HTTP 原由信息[a]
CannotLockAllFeatures（不能锁定所有要素）	400	Cannot lock all features（不能锁定所有要素）
DuplicateStoredQueryIdValue（存储的查询 ID 值已存在）	409	Duplicate stored query identifier（存储的查询 ID 值已存在）
DuplicateStoredQueryParameterName（存储的查询参数名已存在）	409	Duplicate stored query parameter name(存储的查询参数名已存在）
FeaturesNotLocked（要素不能被锁定）	400	Automatic locking not implemented（自动锁定不能执行）
InvalidLockId（无效的锁定 ID）	400	Invalid lock identifier（无效的锁定标识符）
InvalidValue（无效的值）	400	Invalid feature or property value（无效的要素或特性值）
LockHasExpired（锁定过期）	403	Lock identifier has expired（锁标识符过期）
OperationParsingFailed（操作解析失败）	400	Bad request（不正确的请求）
OperationProcessingFailed（操作处理失败）	403	Server processing failed（服务器操作处理失败）
OWS 异常编码		
OperationNotSupported（不支持的操作）	400	Not implemented（没有实现）
MissingParameterValue（缺少参数值）	400	Bad request（不正确的请求）
InvalidParameterValue（无效的参数值）	400	Bad request（不正确的请求）
VersionNegotiationFailed（版本协商失败）	400	Bad request（不正确的请求）
InvalidUpdateSequence（无效的更新序列）	400	Bad request（不正确的请求）
OptionNotSupported（不支持的选项）	400	Not Implemented（没有实现）
NoApplicableCode（不可用的代码）	400	Internal Server Error
[a] HTTP 原由信息只是报告(作为每个 HTTP 标准值)，不要求客户端能够分析理解原因信息。		

示例：下面的示例描述了当出现 DuplicateStoredQueryValue 异常时服务器可能生成的响应。

```
HTTP/1.1 409 Duplicate stored query identifier
Date: Wed, 23 Dec 2008 13:08:15 GMT
Content-Type: text/xml
Content-Length: 473
<? xml version = "1.0" encoding = "UTF-8"? >
< ExceptionReport
  xmlns = "http://www.opengis.net/ows/1.1"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.opengis.net/ows/1.1
                        http://schemas.opengis.net/ows/1.1.0/owsAll.xsd"
   version = "1.0.0" xml:lang = "en">
< Exception exceptionCode = "DuplicateStoredQueryIdValue"
          locator = "urn:uuid: d5054e76-6ac8-42bc-9d39-5ce1a8826518"/>
```

```
</ExceptionReport>
```

D.4 SOAP 绑定

D.4.1 概述

SOAP 是一个应用之间进行通讯的通讯协议。SOAP 定义了通过 Internet 进行通讯的应用间发送信息的格式。SOAP 具有平台无关性和语言无关性,SOAP 消息使用 XML 编码。

一个 SOAP 消息是一个包含下面元素的 XML 文档:

1) 一个必选的 soap:Envelope(信封)元素,标明该 XML 文档为一个 SOAP 消息(后面称为 SOAP Envelope)。
2) 一个可选的 soap:Header(头)元素,包含标题信息(后面称为 SOAP Header)。
3) 一个必选的 soap:Body(信息体)元素,包含请求和响应信息(后面称为 SOAP Body)。
4) 一个可选的 soap:Fault(错误)元素,提供与处理消息时所产生的错误有关的信息(后面称为 SOAP Fault)。

上述元素在 SOAP 信封 1.1 版的命名空间(http://schemas.xmlsoap.org/soap/envelope/)中声明。

遵循本标准的服务可以选择性地支持 SOAP。但是,要实现最大程度的互操作,这些服务应为基于网络的要素服务请求和响应的支持 SOAP 1.1 版(见 W3C SOAP)。

注:网络服务互操作组织(WS-I)的基础专用标准(Basic Profile)1.1 版(http://www.ws-i.org/Profiles/BasicProfile-1.1.html)包含一组公共的网络服务规范,以及这些促进互操作的规范的声明、改进、解释和补充。这个基础专用标准和其后续版本 WS-I 基础专用标准 1.2(委员会认可草案)都应要使用 W3C SOAP。

D.4.2 SOAP Envelope

一个 SOAP Envelope 应遵循在 SOAP 1.1 的第 4 部分"SOAP Envelope"指定的结构。SOAP Envelope 应在命名空间 http://schemas.xmlsoap.org/soap/envelope/中编码为 XML 1.0。

一个 SOAP Envelope 在 soap:Body 元素之后没有任何 soap:Envelope 子元素,因此所谓的 SOAP 后续部分是不允许的。

D.4.3 SOAP Header

本标准没有为 WFS 定义任何 SOAP Header 块。

另外,为了符合本标准,WFS 不应要求任何独有的 SOAP header 块。

D.4.4 SOAP Body

SOAP Body 应只拥有一个子元素。对于所有的请求和响应,除非一个操作的响应为一个 XML 模式(例如一个成功的 DescribeFeatureType 响应),这个子元素应是本标准定义的一个由 XML 编码的 WFS 操作的请求或响应。

注:这同样表示对于 WFS 的请求和响应,一个 SOAP Envelope 不使用 SOAP 第 5 部分定义的 SOAP 编码样式(可选的)。根据 WS-I 基础专用标准,不允许使用该 SOAP 编码样式,这是因为此样式会引起互操作问题。

示例 1:一个 GetFeature 请求的 SOAP 消息。

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:fes="http://www.opengis.net/fes/2.0">
  <soap:Header/>
```

```
    <soap:Body>
      <wfs:GetFeature service="WFS" version="2.0" resultType="results"
  outputFormat="application/gml+xml; version=3.2" count="50">
        <wfs:Query typeNames="myns:ROADS"/>
      </wfs:GetFeature>
    </soap:Body>
  </soap:Envelope>
```

示例 2：一个 GetFeature 响应的 SOAP 消息，包含了一个 FeatureCollection(要素集)。

```
  <soap:Envelope
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:wfs="http://www.opengis.net/wfs/2.0"
    xmlns:fes="http://www.opengis.net/fes/2.0">
    <soap:Header/>
    <soap:Body>
      <wfs:FeatureCollection
        xmlns="http://www.someserver.com/myns"
        xmlns:myns="http://www.someserver.com/myns"
        xmlns:wfs="http://www.opengis.net/wfs/2.0"
        xmlns:gml="http://www.opengis.net/gml/3.2"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://schemas.opengis.net/
wfs/2.0.0/wfs.xsd                    http://www.someserver.com/myns ex07.xsd">
        <gml:boundedBy>
          <gml:Envelope srsName="urn:ogc:def:crs:EPSG::4326">
            <gml:lowerCorner>-180.0 -90.0</gml:lowerCorner>
            <gml:upperCorner>180.0 90.0</gml:upperCorner>
          </gml:Envelope>
        </gml:boundedBy>
        <wfs:member>
          <TreesA_1M>
           ...
          </TreesA_1M>
         </wfs:member>
      </wfs:FeatureCollection>
    </soap:Body>
  </soap:Envelope>
```

D.4.5 SOAP Body 中编码的 XML 模式

如果一个 DescribeFeatureType 操作成功，那么其响应是一个 XML 模式。在 SOAP Body 中封装 XML 模式会导致一些问题，所以，当使用 SOAP 且 WFS 操作的响应为 XML 模式时，XML 模式使用 base64 编码，且编码的响应应包装在一个 XML 元素中。

当使用 SOAP 时，DescribeFeatureType 操作的响应遵循下面 XML 模式片段：

```
  <xsd:complexType name="DescribeFeatureTypeResponseType">
    <xsd:sequence>
      <xsd:element name="DescribeFeatureTypeResponse" type="xsd:base64"/>
```

```
        </xsd:sequence>
      </xsd:complexType>
```

示例：将下面 XML 模式序列化：

```
<? xml version = "1.0" encoding = "UTF - 8"? >
< xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema" xmlns:gml = "http://www.opengis.net/gml/3.2"
xmlns:myns = "http://www.somenamespace.com/myns" elementFormDefault = "qualified"
targetNamespace = "http://www.somenamespace.com/myns">
    < xsd:import namespace = "http://www.opengis.net/gml/3.2"
schemaLocation = "http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
    < xsd:complexType name = "SomeFeatureType">
    < xsd:complexContent >
        < xsd:extension base = "gml:AbstractFeatureType">
            < xsd:sequence >
              < xsd:element maxOccurs = "1" minOccurs = "0" name = "name" type = "xsd:string"/>
               < xsd:element maxOccurs = "1" minOccurs = "0" name = "title" nillable = "true" type = "xsd:
string"/>
              < xsd:element maxOccurs = "1" minOccurs = "0" name = "geometry" nillable = "true" type = "gml:
PointPropertyType"/>
            </xsd:sequence >
        </xsd:extension >
    </xsd:complexContent >
    </xsd:complexType >
    < xsd:element name = "SomeFeature" substitutionGroup = "gml:AbstractFeature"
type = "myns:SomeFeatureType"/>
</xsd:schema >
```

在 SOAP Envelope 中使用 base64 编码后，结果如下：

```
< soap:Envelope
  xmlns:soap = "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:fes = "http://www.opengis.net/fes/2.0">
  < soap:Header/>
  < soap:Body >
< wfs:DescribeFeatureTypeResponse > SB4bWxuczp4c2Q9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE 1MU2NoZW1hIiB4bWxu
czpnbWw9Imh0dHA6Ly93d3cub3Blbmdpcy5uZXQvZ21sIiB4bWxuczpteW5zPSJ odHRwOi8vd3d3LnNvbWVuYW11c3BhY2UuY29tL215bn
MiIGVsZW1lbnRGb3JtRGVmYXVsdD0icXVhbGlmaWVk IiB0YXJnZXR0YW11c3BhY2U9Imh0dHA6Ly93d3cuc29tZW5hbWVzcGFjZS5jb20
vbXlucyI + PHhzZDppbXBvcnQg bmFtZXNwYWNlPSJodHRwOi8vd3d3Lm9wZW5naXMubmV0L2dtbCIgc2NoZW1hTG9jYXRpb249Imh0dHA
6Ly9z Y2hlbWFzLm9wZW5naXMubmV0L2dtbC8zLjIuMS9nbWwueHNkIi8 + PHhzZDpjb21wbGV4VHlwZSBuYW1lPSJzb 21lRmVhdHVyZVR
5cGUiPjx4c2Q6Y29tcGxleENvbnRlbnQ + PHhzZDpleHRlbnNpb24gYmFzZT0iZ21sOkFic3RyY WN0RmVhdHVyZVR5cGUiPjx4c2Q6c2Vx
dWVuY2U + PHhzZDplbGVtZW50IG1heE9jY3Vycz0iMSIgbWluT2NjdXJ zPSIwIiBuYW1lPSJuYW11IiB0eXBlPSJ4c2Q6c3RyaW5nIi8
 + PHhzZDplbGVtZW50IG1heE9jY3Vycz0iMSIgbWluT2 NjdXJzPSIwIiBuYW1lPSJ0aXRsZSIgbmlsbGFibGU9InRydWUiIHR5cGU9In
hzZDpzdHJpbmciLz48eHNkOmVsZW 1lbnQgbWF4T2NjdXJzPSIxIiBtaW5PY2N1cnM9IjAiIG5hbWU9Imdlb21ldHJ5IiBuaWxsYWJs
ZT0idHJ1ZSIgdHlwZ T0iZ21sOlBvaW50UHJvcGVydHlUeXBlIi8 + PC94c2Q6c2VxdWVuY2U + PC94c2Q6ZXh0ZW5zaW9uPjwveHNk
Om NvbXBsZXhDb250ZW50PjwveHNkOmNvbXBsZXhUeXBlPjx4c2Q6ZWxlbWVudCBuYW1lPSJzb21lRmVhdHVyZSI gc3Vic3Rpd
HV0aW9uR3JvdXA9ImdtbDpfRmVhdHVyZSIgdHlwZT0ibXluczpzb21lRmVhdHVyZVR5cGUiLz48L3h zZDpzY2hlbWE +
</wfs:DescribeFeatureTypeResponse >
    </soap:Body >
  </soap:Envelope >
```

D.4.6 SOAP Fault

SOAP Fault 消息的结构遵循 W3C SOAP:2007 的第 4.4 部分 SOAP Fault。

< soap:faultcode >(错误代码)元素的值应具备 soap:Client(客户端)或 soap:Server(服务器)内容以表明这是一个服务异常。如果消息格式不对或者消息没有包含操作成功所需的恰当信息(例如包含一个无效的 WFS 请求),那么使用 soap:Client 值。如果消息由于一些原因不能处理,这些原因是归因于消息的处理而不是消息本身内容时,那么使用 soap:Server 值。< soap:faultstring >(错误字符串)元素应显示"A service exception was encountered."(发生服务异常)。使用这个固定字符串是因为异常的细节由 soap:detail(细节)元素的 ows:ExceptionReport(异常报告)元素(见 OGC 06-121r3:2009)指定并且在本标准的 7.5 中有所描述。

示例 : SOAP Fault 消息

```
< soap:Fault xmlns:soap = "http://schemas.xmlsoap.org/soap/envelope/">
  < soap:faultcode > soap:Server </soap:faultcode >
  < soap:faultstring > A service exception was encountered.</soap:faultstring >
  < soap:detail >
    < ows:ExceptionReport xmlns:ows = "http://www.opengis.net/ows">
        ...
    </ows:ExceptionReport >
  </soap:detail >
</soap:Fault >
```

D.4.7 SOAP HTTP 绑定

如果 WFS 支持 SOAP,那么 HTTP POST 方法应用于 SOAP 消息。

为了实现互操作,在 HTTP 请求中,SOAPAction HTTP 的头字段应为标有引号的字符串。如果在操作的绑定中没有提供 soapAction(soap 行为),那么标题字段值为空字符串(""),否则标题字段值为 WSDL 文档中指定的 soapAction 值。

既然 SOAPAction HTTP 头字段是一个提示,所以为了正确处理消息,WFS 不必依赖 SOAPAction HTTP 头字段的值。

注意,WS-I 基本专用标准 1.1 版描述如下:

"实验表明要求为 SOAPAction HTTP 头字段值标上引号增加了实例之间的互操作能力。尽管 HTTP 允许没有标上引号的标题字段值,但是一些 SOAP 实例要求标上引号。SOAPAction 对于处理器来说只是一个提示,所有关于消息内容的关键信息都在 soap:Envelope 之中。"

本标准遵循 WS-I BP1.1 的处理方法。

附 录 E
（规范性附录）
网络服务描述语言（WSDL）

E.1 概述

网络服务描述语言 1.1（WSDL 1.1）中 WSDL 定义为："一种 XML 格式，用于将网络服务描述为一组操作基于文档或基于程序信息的消息的端点。这些操作和消息是抽象描述的，然后与具体的网络协议和定义端点的消息格式进行绑定。相关的具体端点联合成抽象端点（服务）。"

对于 WFS，本标准定义了 WSDL 操作、消息以及绑定。

为提高 WFS 的互操作性，本标准定义了以下针对 WSDL 文档的附加约束：

本附件中命名空间前缀 wsdl 绑定到 http://schemas.smlsoap.org/wsdl/。

本附件中命名空间前缀 soapbind 绑定到 http://schemas.xmlsoap.org/wsdl/soap/。

E.2 WSDL 中的 WFS 操作

WFS 可以在能力描述文档中通过 WSDL 1.1:2001 第 2 部分定义的 WSDL 文档声明它支持的所有操作。服务器的能力描述文档（见 8.3.3）中使用 wfs:WSDL 元素引用所包含的 WSDL 文档。

示例 1：< wfs:WSDL xlink:href=http://www.someserver.com/wfs.wsdl/>

如本标准所定义，WFS 应为它支持的每一个 WFS 操作声明一个 WSDL 操作。输入和输出消息的格式在 SOAP Binding 中描述。

示例 2：描述为一个 WSDL 操作的 GetFeature 操作：

```
< wsdl:operation name = "wfs.getFeature">
  < wsdl:input message = "wfs - req:GetFeatureRequest"/>
  < wsdl:output message = "wfs - resp:GetFeatureResponse"/>
  < wsdl:fault name = "ServiceExceptionReport" message = "wfs - resp:ServiceExceptionReport"/>
</wsdl:operation >
```

E.3 SOAP Binding（绑定）

WSDL 文档中一个 wsdl:binding 元素使用 WSDL 1.1:2001 第 3 部分定义的 WSDL SOAP Binding。在 WSDL 文档中为 WFS 声明的消息应序列化为本标准附件 D 中定义的 SOAP 消息。

D.4.7 定义 HTTP 为 WFS 的传输协议。在 WSDL 文档的 wsdl:binding 元素中指定 SOAP 绑定和 HTTP 传输。特别指出的是，soapbind:binding 子元素的属性 transport（传输）应拥有值"http://schemas.xmlsoap.org/soap/http"。

注：这个要求不妨碍 HTTPS 的使用。

E.4 Binding 方式

〔WS-I BP 1.1〕定义了两种绑定方式：

在 W3C WSDL 1.1 中，wsdl:message(消息)元素用来表达传输数据的抽象定义，它使用 wsdl:binding 元素定义这些抽象定义如何绑定到一个具体的消息序列化中。

document-literal(文档-文字)绑定是一个 wsdl:binding 元素，其子元素 wsdl:operation(操作)元素全部都是 document-literal 操作。

document-literal 操作是 wsdl:binding 元素的 wsdl:operation 子元素，wsdl:binding 的 soapbind:body 派生元素指定 use(应用)属性的值为“literal”，并且满足下面的一条：

- 值为“document”的 style(风格)属性指定到 soapbind:operation 子元素上；
- style 属性没出现在 soapbind:operation 子元素上，并且在相应的 wsdl:binding 元素的 soapbind:binding 元素指定 style 属性值为“document”；
- 属性 style 没出现在 soapbind:operation 子元素上，也没有出现在 wsdl:binding 元素包含的 soapbind:binding 元素上。

遵循本标准的 WFS 的 WSDL 文档只能使用上面描述的文档-文字绑定。

WSDL 文档的 wsdl:binding 元素在它的每个 soapbind:body 元素和 soapbind:fault 元素中只提及已使用属性 element(元素)定义的 wsdl:part 元素。

这部分内容在 soap:Body 元素中进行序列化，并只包含一个子元素，如本标准中 D.4.5 中所描述。

对于 WFS，如果 parts()属性指定，那么一个 WSDL 文档中的 wsdl:binding 元素在它的每个 soapbind:body 元素中最多包含 parts 属性中的一个 part。

示例 1：对于 GetFeature 请求的 WSDL 消息定义。wfs:GetFeature 元素表示在 WFS 的 XML 模式中 GetFeature 定义。

```
<wsdl:message name = "GetFeatureRequest">
  <wsdl:part name = "Body" element = "wfs:GetFeature"/>
</wsdl:message>
```

示例 2：GetFeature 响应的 WSDL 消息定义

```
<wsdl:message name = "GetFeatureResponse">
  <wsdl:part name = "Body" element = "wfs:FeatureCollection"/>
</wsdl:message>
```

示例 3：提供 GetCapabilities、DescribeFeatureType 和 GetFeature 操作的 WFS 的文档文字绑定。操作的输入和输出消息在其他地方定义，见“WSDL 中的 WFS 操作”中的示例。

```
<wsdl:binding name = "wfs - SOAP" type = "wfs - req:wfs">
    <wsdl:documentation>
    WFS interface bound to SOAP over HTTP/1.1.
    </wsdl:documentation>
    <soap:binding style = "document"
            transport = "http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name = "wfs.getCapabilities">
      <soap:operation
      soapAction = "http://www.opengis.net/wfs/requests#GetCapabilities"/>
      <wsdl:input>
        <soap:body use = "literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use = "literal"/>
      </wsdl:output>
      <wsdl:fault name = "ServiceExceptionReport">
```

```
        <soap:fault use="literal" name="ServiceExceptionReport"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="wfs.describeFeatureType">
      <soap:operation
      soapAction="http://www.opengis.net/wfs/requests#DescribeFeatureType"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ServiceExceptionReport">
        <soap:fault use="literal" name="ServiceExceptionReport"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="wfs.getFeature">
      <soap:operation
      soapAction="http://www.opengis.net/wfs/requests#GetFeature"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ServiceExceptionReport">
        <soap:fault use="literal" name="ServiceExceptionReport"/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
```


E.5 服务

WFS 应在 WSDL 文档中提供一个在 WSDL 的 2.7 服务中定义的 wsdl:service 元素。

示例：下面 XML 片段是 WFS 的 service 元素的一个示例。

```
<wsdl:service name="WFS-www.myservice.com">
  <wsdl:documentation>A WFS 2.0 implementation. Includes SOAP bindings for the WFS
interfaces.</wsdl:documentation>
  <wsdl:port name="wfs-SOAP-Port" binding="wfs-soap:wfs-SOAP">
   <soap:address location="http://www.myservice.com/wfs-soap/services/wfs"/>
  </wsdl:port>
</wsdl:service>
```

E.6 使用 WSDL 描述服务

E.6.1 概述

这一条中包含 WFS 的 WSDL 描述,它同时还包含端点定义的示例,用来说明 WSDL 描述的使用。

E.6.2 wfs-xml-interfaces.wsdl

```
<? xml version = "1.0" encoding = "ISO - 8859 - 1"? >
< wsdl:definitions
  targetNamespace = "http://www.opengis.net/wfs/requests/2.0"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:ows = "http://www.opengis.net/ows/1.1"
  xmlns:wfs-req = "http://www.opengis.net/wfs/requests/2.0"
  xmlns:wfs-resp = "http://www.opengis.net/wfs/responses/2.0"
  xmlns:wsdl = "http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
  < wsdl:documentation xmlns:dc = "http://purl.org/dc/elements/1.1/">
    < dc:identifier > urn:opengis:spec:wfs:wsdl-1.1:interfaces:2.0.0 </dc:identifier >
    < dc:date > 2008-08-31 </dc:date >
    < dc:description >
      This is the normative abstract service interface definition
      for the OpenGIS Web Feature Service, v2.0.0. The WSDL 1.1 syntax is
      used to describe the interface signatures and message structures.
    </dc:description >
  </wsdl:documentation >
  < wsdl:import namespace = "http://www.opengis.net/wfs/responses/2.0"
        location = "./wfs-responses.wsdl"/>
  < wsdl:types >
    < wsdl:documentation >
      Convenience schema that defines all common WFS message elements.
    </wsdl:documentation >
    < xsd:schema targetNamespace = "http://www.opengis.net/wfs/2.0"
      xmlns:wfs = "http://www.opengis.net/wfs/2.0"
      xmlns:gml = "http://www.opengis.net/gml/3.2"
      xmlns:ogc = "http://www.opengis.net/ogc/1.1"
      xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
      elementFormDefault = "qualified" version = "1.1.0">
      < xsd:include schemaLocation = "http://schemas.opengis.net/wfs/2.0.0/wfs.xsd"/>
    </xsd:schema >
  </wsdl:types >
  < wsdl:message name = "GetCapabilitiesRequest">
    < wsdl:part name = "Body" element = "wfs:GetCapabilities"/>
  </wsdl:message >
  < wsdl:message name = "DescribeFeatureTypeRequest">
    < wsdl:part name = "Body" element = "wfs:DescribeFeatureType"/>
```

```
</wsdl:message>
<wsdl:message name="GetPropertyValueRequest">
  <wsdl:part name="Body" element="wfs:GetPropertyValue"/>
</wsdl:message>
<wsdl:message name="GetFeatureRequest">
  <wsdl:part name="Body" element="wfs:GetFeature"/>
</wsdl:message>
<wsdl:message name="GetFeatureWithLockRequest">
  <wsdl:part name="Body" element="wfs:GetFeatureWithLock"/>
</wsdl:message>
<wsdl:message name="LockFeatureRequest">
  <wsdl:part name="Body" element="wfs:LockFeature"/>
</wsdl:message>
<wsdl:message name="TransactionRequest">
  <wsdl:part name="Body" element="wfs:Transaction"/>
</wsdl:message>
<wsdl:message name="ListStoredQueriesRequest">
  <wsdl:part name="Body" element="wfs:ListStoredQueries"/>
</wsdl:message>
<wsdl:message name="DescribeStoredQueriesRequest">
  <wsdl:part name="Body" element="wfs:DescribeStoredQueries"/>
</wsdl:message>
<wsdl:message name="CreateStoredQueryRequest">
  <wsdl:part name="Body" element="wfs:CreateStoredQuery"/>
</wsdl:message>
<wsdl:message name="DropStoredQueryRequest">
  <wsdl:part name="Body" element="wfs:DropStoredQuery"/>
</wsdl:message>
<wsdl:portType name="wfs">
  <wsdl:operation name="wfs.getCapabilities">
    <wsdl:input message="wfs-req:GetCapabilitiesRequest"/>
    <wsdl:output message="wfs-resp:GetCapabilitiesResponse"/>
    <wsdl:fault name="ServiceExceptionReport"
      message="wfs-resp:ServiceExceptionReport"/>
  </wsdl:operation>
  <wsdl:operation name="wfs.describeFeatureType">
    <wsdl:input message="wfs-req:DescribeFeatureTypeRequest"/>
    <wsdl:output message="wfs-resp:DescribeFeatureTypeResponse"/>
    <wsdl:fault name="ServiceExceptionReport"
      message="wfs-resp:ServiceExceptionReport"/>
  </wsdl:operation>
  <wsdl:operation name="wfs.getPropertyValue">
    <wsdl:input message="wfs-req:GetPropertyValueRequest"/>
    <wsdl:output message="wfs-resp:GetPropertyValueResponse"/>
```

```
    <wsdl:fault name = "ServiceExceptionReport"
      message = "wfs-resp:ServiceExceptionReport"/>
  </wsdl:operation>
  <wsdl:operation name = "wfs.getFeature">
    <wsdl:input message = "wfs-req:GetFeatureRequest"/>
    <wsdl:output message = "wfs-resp:GetFeatureResponse"/>
    <wsdl:fault name = "ServiceExceptionReport"
      message = "wfs-resp:ServiceExceptionReport"/>
  </wsdl:operation>
  <wsdl:operation name = "wfs.getFeatureWithLock">
    <wsdl:input message = "wfs-req:GetFeatureWithLockRequest"/>
    <wsdl:output message = "wfs-resp:GetFeatureWithLockResponse"/>
    <wsdl:fault name = "ServiceExceptionReport"
      message = "wfs-resp:ServiceExceptionReport"/>
  </wsdl:operation>
  <wsdl:operation name = "wfs.lockFeature">
    <wsdl:input message = "wfs-req:LockFeatureRequest"/>
    <wsdl:output message = "wfs-resp:LockFeatureResponse"/>
    <wsdl:fault name = "ServiceExceptionReport"
      message = "wfs-resp:ServiceExceptionReport"/>
  </wsdl:operation>
  <wsdl:operation name = "wfs.transaction">
    <wsdl:input message = "wfs-req:TransactionRequest"/>
    <wsdl:output message = "wfs-resp:TransactionResponse"/>
    <wsdl:fault name = "ServiceExceptionReport"
      message = "wfs-resp:ServiceExceptionReport"/>
  </wsdl:operation>
  <wsdl:operation name = "wfs.listStoredQueries">
    <wsdl:input message = "wfs-req:ListStoredQueriesRequest"/>
    <wsdl:output message = "wfs-resp:ListStoredQueriesResponse"/>
    <wsdl:fault name = "ServiceExceptionReport"
      message = "wfs-resp:ServiceExceptionReport"/>
  </wsdl:operation>
  <wsdl:operation name = "wfs.describeStoredQueries">
    <wsdl:input message = "wfs-req:DescribeStoredQueriesRequest"/>
    <wsdl:output message = "wfs-resp:DescribeStoredQueriesResponse"/>
    <wsdl:fault name = "ServiceExceptionReport"
      message = "wfs-resp:ServiceExceptionReport"/>
  </wsdl:operation>
  <wsdl:operation name = "wfs.createStoredQuery">
    <wsdl:input message = "wfs-req:CreateStoredQueryRequest"/>
    <wsdl:output message = "wfs-resp:CreateStoredQueryResponse"/>
    <wsdl:fault name = "ServiceExceptionReport"
      message = "wfs-resp:ServiceExceptionReport"/>
```

```
    </wsdl:operation>
    <wsdl:operation name="wfs.dropStoredQuery">
      <wsdl:input message="wfs-req:DropStoredQueryRequest"/>
      <wsdl:output message="wfs-resp:DropStoredQueryResponse"/>
      <wsdl:fault name="ServiceExceptionReport"
        message="wfs-resp:ServiceExceptionReport"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

E.6.3 wfs-kvp-interfaces.wsdl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<wsdl:definitions targetNamespace="http://www.opengis.net/wfs/requests/kvp/2.0"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wfs-req-kvp="http://www.opengis.net/wfs/requests/kvp/2.0"
  xmlns:wfs-kvp="http://www.opengis.net/wfs-kvp/2.0"
  xmlns:wfs-req="http://www.opengis.net/wfs/requests/kvp/2.0"
  xmlns:wfs-resp="http://www.opengis.net/wfs/responses/2.0"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:documentation xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:identifier>urn:opengis:spec:wfs:wsdl-1.1:interfaces:2.0.0</dc:identifier>
    <dc:date>2008-08-31</dc:date>
    <dc:description>
      This is the normative abstract service interface definition for
      the OpenGIS Web Feature Service, v2.0.0. The WSDL 1.1 syntax is
      used to describe the interface signatures and message structures.
    </dc:description>
  </wsdl:documentation>
  <wsdl:import namespace="http://www.opengis.net/wfs/responses/2.0"
    location="./wfs-responses.wsdl"/>
  <wsdl:types>
    <wsdl:documentation>
      Convenience schema that defines all common WFS message elements.
    </wsdl:documentation>
    <xsd:schema targetNamespace="http://www.opengis.net/wfs/2.0"
      xmlns:wfs="http://www.opengis.net/wfs/2.0"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified" version="1.1.0">
      <xsd:import namespace="http://www.opengis.net/wfs-kvp/2.0"
        schemaLocation="./wfs-kvp.xsd"/>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="GetCapabilitiesRequest">
    <wsdl:part name="Body" element="wfs-kvp:GetCapabilities"/>
```

```
    </wsdl:message>
    <wsdl:message name="DescribeFeatureTypeRequest">
      <wsdl:part name="Body" element="wfs-kvp:DescribeFeatureType"/>
    </wsdl:message>
    <wsdl:message name="GetPropertyValueRequest">
      <wsdl:part name="Body" element="wfs-kvp:GetPropertyValue"/>
    </wsdl:message>
    <wsdl:message name="GetFeatureRequest">
      <wsdl:part name="Body" element="wfs-kvp:GetFeature"/>
    </wsdl:message>
    <wsdl:message name="GetFeatureWithLockRequest">
      <wsdl:part name="Body" element="wfs-kvp:LockFeature"/>
    </wsdl:message>
    <wsdl:message name="LockFeatureRequest">
      <wsdl:part name="Body" element="wfs-kvp:LockFeature"/>
    </wsdl:message>
    <wsdl:portType name="wfs">
      <wsdl:operation name="wfs.getCapabilities">
        <wsdl:input message="wfs-req-kvp:GetCapabilitiesRequest"/>
        <wsdl:output message="wfs-resp:GetCapabilitiesResponse"/>
        <wsdl:fault name="ServiceExceptionReport"
          message="wfs-resp:ServiceExceptionReport"/>
      </wsdl:operation>
      <wsdl:operation name="wfs.describeFeatureType">
        <wsdl:input message="wfs-req-kvp:DescribeFeatureTypeRequest"/>
        <wsdl:output message="wfs-resp:DescribeFeatureTypeResponse"/>
        <wsdl:fault name="ServiceExceptionReport"
          message="wfs-resp:ServiceExceptionReport"/>
      </wsdl:operation>
      <wsdl:operation name="wfs.getPropertyValue">
        <wsdl:input message="wfs-req-kvp:GetPropertyValueRequest"/>
        <wsdl:output message="wfs-resp:GetPropertyValueResponse"/>
        <wsdl:fault name="ServiceExceptionReport"
          message="wfs-resp:ServiceExceptionReport"/>
      </wsdl:operation>
      <wsdl:operation name="wfs.getFeature">
        <wsdl:input message="wfs-req-kvp:GetFeatureRequest"/>
        <wsdl:output message="wfs-resp:GetFeatureResponse"/>
        <wsdl:fault name="ServiceExceptionReport"
          message="wfs-resp:ServiceExceptionReport"/>
      </wsdl:operation>
      <wsdl:operation name="wfs.getFeatureWithLock">
        <wsdl:input message="wfs-req-kvp:GetFeatureWithLockRequest"/>
        <wsdl:output message="wfs-resp:GetFeatureResponse"/>
```

```
      <wsdl:fault name = "ServiceExceptionReport"
        message = "wfs-resp:ServiceExceptionReport"/>
    </wsdl:operation>
    <wsdl:operation name = "wfs.lockFeature">
      <wsdl:input message = "wfs-req-kvp:LockFeatureRequest"/>
      <wsdl:output message = "wfs-resp:LockFeatureResponse"/>
      <wsdl:fault name = "ServiceExceptionReport"
        message = "wfs-resp:ServiceExceptionReport"/>
    </wsdl:operation>
    <wsdl:operation name = "wfs.listStoredQueries">
      <wsdl:input message = "wfs-req-kvp:ListStoredQueriesRequest"/>
      <wsdl:output message = "wfs-resp:ListStoredQueriesResponse"/>
      <wsdl:fault name = "ServiceExceptionReport"
        message = "wfs-resp:ServiceExceptionReport"/>
    </wsdl:operation>
    <wsdl:operation name = "wfs.describeStoredQueries">
      <wsdl:input message = "wfs-req-kvp:DescribeStoredQueriesRequest"/>
      <wsdl:output message = "wfs-resp:DescribeStoredQueriesResponse"/>
      <wsdl:fault name = "ServiceExceptionReport"
        message = "wfs-resp:ServiceExceptionReport"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

E.6.4 wfs- responses.wsdl

```
<? xml version = "1.0" encoding = "ISO-8859-1"? >
<wsdl:definitions
  targetNamespace = "http://www.opengis.net/wfs/responses/2.0"
  xmlns:wfs-util = "http://www.opengis.net/wfs-util/2.0"
  xmlns:wfs = "http://www.opengis.net/wfs/2.0"
  xmlns:ows = "http://www.opengis.net/ows/1.1"
  xmlns:wsdl = "http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
  <wsdl:documentation xmlns:dc = "http://purl.org/dc/elements/1.1/">
    <dc:date > 2008-08-31 </dc:date >
    <dc:description >
      This WSDL document defines the response messages and types
      for the WFS.
    </dc:description >
  </wsdl:documentation >
  <wsdl:types >
    <xsd:schema targetNamespace = "http://www.opengis.net/wfs/2.0"
      xmlns:wfs = "http://www.opengis.net/wfs/2.0"
      xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
      elementFormDefault = "qualified" version = "1.1.0">
```

```
        <xsd:include schemaLocation = "http://schemas.opengis.net/wfs/2.0.0/wfs.xsd"/>
        <xsd:import namespace = "http://www.opengis.net/wfs-util/2.0"
          schemaLocation = "./wfs-util.xsd"/>
        <xsd:import namespace = "http://www.opengis.net/ows/1.1"
          schemaLocation = "http://schemas.opengis.net/ows/1.1.0/owsExceptionReport.xsd"/>
      </xsd:schema>
    </wsdl:types>
    <wsdl:message name = "ServiceExceptionReport">
      <wsdl:part element = "ows:ExceptionReport" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "GetCapabilitiesResponse">
      <wsdl:part element = "wfs:WFS_Capabilities" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "DescribeFeatureTypeResponse">
      <wsdl:part element = "wfs-util:DescribeFeatureTypeResponse" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "GetPropertyValueResponse">
      <wsdl:part element = "wfs:ValueCollection" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "GetFeatureResponse">
      <wsdl:part element = "wfs:FeatureCollection" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "GetFeatureWithLockResponse">
      <wsdl:part element = "wfs:FeatureCollection" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "LockFeatureResponse">
      <wsdl:part element = "wfs:LockFeatureResponse" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "TransactionResponse">
      <wsdl:part element = "wfs:TransactionResponse" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "ListStoredQueriesResponse">
      <wsdl:part element = "wfs:ListStoredQueriesResponse" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "DescribeStoredQueriesResponse">
      <wsdl:part element = "wfs:DescribeStoredQueriesResponse" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "CreateStoredQueryResponse">
      <wsdl:part element = "wfs:CreateStoredQueryResponse" name = "Body"/>
    </wsdl:message>
    <wsdl:message name = "DropStoredQueryResponse">
      <wsdl:part element = "wfs:DropStoredQueryResponse" name = "Body"/>
    </wsdl:message>
  </wsdl:definitions>
```

E.6.5 wfs-http-bindings.wsdl

```
<? xml version = "1.0" encoding = "ISO-8859-1"? >
< wsdl:definitions
  targetNamespace = "http://www.opengis.net/wfs/http/2.0"
  xmlns:wfs-req = "http://www.opengis.net/wfs/requests/2.0"
  xmlns:wsdl = "http://schemas.xmlsoap.org/wsdl/"
  xmlns:http = "http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:mime = "http://schemas.xmlsoap.org/wsdl/mime/">
  < wsdl:documentation xmlns:dc = "http://purl.org/dc/elements/1.1/">
    < dc:description >
      HTTP/1.1 protocol bindings for WFS interfaces.
    </dc:description >
    < dc:date > 2008-08-31 </dc:date >
  </wsdl:documentation >
  < wsdl:import namespace = "http://www.opengis.net/wfs/requests/2.0"
          location = "./wfs-xml-interfaces.wsdl"/>
  < wsdl:binding name = "wfs-POST" type = "wfs-req:wfs">
    < wsdl:documentation >
      wfs interface bound to the HTTP/1.1 POST method.
    </wsdl:documentation >
    < http:binding verb = "POST"/>
    < wsdl:operation name = "wfs.getCapabilities">
      < http:operation location = "{id}"/>
      < wsdl:input >
        < mime:mimeXml/>
      </wsdl:input >
      < wsdl:output >
        < mime:mimeXml/>
      </wsdl:output >
      < wsdl:fault name = "ServiceExceptionReport">
        < soap:fault use = "literal" name = "ServiceExceptionReport"/>
      </wsdl:fault >
    </wsdl:operation >
    < wsdl:operation name = "wfs.describeFeatureType">
      < http:operation location = "{id}"/>
      < wsdl:input >
        < mime:mimeXml/>
      </wsdl:input >
      < wsdl:output >
        < mime:mimeXml/>
      </wsdl:output >
      < wsdl:fault name = "ServiceExceptionReport">
        < soap:fault use = "literal" name = "ServiceExceptionReport"/>
```

```
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.getPropertyValue">
    <http:operation location="{id}"/>
    <wsdl:input>
      <mime:mimeXml/>
    </wsdl:input>
    <wsdl:output>
      <mime:mimeXml/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.getFeature">
    <http:operation location="{id}"/>
    <wsdl:input>
      <mime:mimeXml/>
    </wsdl:input>
    <wsdl:output>
      <mime:mimeXml/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.getFeatureWithLock">
    <http:operation location="{id}"/>
    <wsdl:input>
      <mime:mimeXml/>
    </wsdl:input>
    <wsdl:output>
      <mime:mimeXml/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.lockFeature">
    <http:operation location="{id}"/>
    <wsdl:input>
      <mime:mimeXml/>
    </wsdl:input>
    <wsdl:output>
```

```
      <mime:mimeXml/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.transaction">
    <http:operation location="{id}"/>
    <wsdl:input>
      <mime:mimeXml/>
    </wsdl:input>
    <wsdl:output>
      <mime:mimeXml/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.listStoredQueries">
    <http:operation location="{id}"/>
    <wsdl:input>
      <mime:mimeXml/>
    </wsdl:input>
    <wsdl:output>
      <mime:mimeXml/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.describeStoredQueries">
    <http:operation location="{id}"/>
    <wsdl:input>
      <mime:mimeXml/>
    </wsdl:input>
    <wsdl:output>
      <mime:mimeXml/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.createStoredQuery">
    <http:operation location="{id}"/>
```

```
        <wsdl:input>
          <mime:mimeXml/>
        </wsdl:input>
        <wsdl:output>
          <mime:mimeXml/>
        </wsdl:output>
        <wsdl:fault name="ServiceExceptionReport">
          <soap:fault use="literal" name="ServiceExceptionReport"/>
        </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="wfs.dropStoredQuery">
        <http:operation location="{id}"/>
        <wsdl:input>
          <mime:mimeXml/>
        </wsdl:input>
        <wsdl:output>
          <mime:mimeXml/>
        </wsdl:output>
        <wsdl:fault name="ServiceExceptionReport">
          <soap:fault use="literal" name="ServiceExceptionReport"/>
        </wsdl:fault>
      </wsdl:operation>
    </wsdl:binding>
  </wsdl:definitions>
```


E.6.6 wfs-kvp-bindings.wsdl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<wsdl:definitions targetNamespace="http://www.opengis.net/wfs/http/kvp/2.0"
  xmlns:wfs-req-kvp="http://www.opengis.net/wfs/requests/kvp/2.0"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/">
  <wsdl:documentation xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:description>
      HTTP/1.1 protocol bindings for WFS interfaces.
    </dc:description>
    <dc:date>2008-08-31</dc:date>
  </wsdl:documentation>
  <wsdl:import namespace="http://www.opengis.net/wfs/requests/kvp/2.0"
    location="./wfs-kvp-interfaces.wsdl"/>
  <wsdl:binding name="wfs-GET" type="wfs-req-kvp:wfs">
    <wsdl:documentation>
      wfs interface bound to the HTTP/1.1 GET method.
    </wsdl:documentation>
```

```
<http:binding verb = "GET"/>
<wsdl:operation name = "wfs.getCapabilities">
  <http:operation location = "{id}"/>
  <wsdl:input>
    <http:urlEncoded/>
  </wsdl:input>
  <wsdl:output>
    <mime:mimeXml/>
  </wsdl:output>
  <wsdl:fault name = "ServiceExceptionReport">
    <soap:fault use = "literal" name = "ServiceExceptionReport"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name = "wfs.describeFeatureType">
  <http:operation location = "{id}"/>
  <wsdl:input>
    <http:urlEncoded/>
  </wsdl:input>
  <wsdl:output>
    <mime:mimeXml/>
  </wsdl:output>
  <wsdl:fault name = "ServiceExceptionReport">
    <soap:fault use = "literal" name = "ServiceExceptionReport"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name = "wfs.getPropertyValue">
  <http:operation location = "{id}"/>
  <wsdl:input>
    <http:urlEncoded/>
  </wsdl:input>
  <wsdl:output>
    <mime:mimeXml/>
  </wsdl:output>
  <wsdl:fault name = "ServiceExceptionReport">
    <soap:fault use = "literal" name = "ServiceExceptionReport"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name = "wfs.getFeature">
  <http:operation location = "{id}"/>
  <wsdl:input>
    <http:urlEncoded/>
  </wsdl:input>
  <wsdl:output>
    <mime:mimeXml/>
```

```
      </wsdl:output>
      <wsdl:fault name="ServiceExceptionReport">
        <soap:fault use="literal" name="ServiceExceptionReport"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="wfs.getFeatureWithLock">
      <http:operation location="{id}"/>
      <wsdl:input>
        <http:urlEncoded/>
      </wsdl:input>
      <wsdl:output>
        <mime:mimeXml/>
      </wsdl:output>
      <wsdl:fault name="ServiceExceptionReport">
        <soap:fault use="literal" name="ServiceExceptionReport"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="wfs.lockFeature">
      <http:operation location="{id}"/>
      <wsdl:input>
        <http:urlEncoded/>
      </wsdl:input>
      <wsdl:output>
        <mime:mimeXml/>
      </wsdl:output>
      <wsdl:fault name="ServiceExceptionReport">
        <soap:fault use="literal" name="ServiceExceptionReport"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="wfs.listStoredQueries">
      <http:operation location="{id}"/>
      <wsdl:input>
        <http:urlEncoded/>
      </wsdl:input>
      <wsdl:output>
        <mime:mimeXml/>
      </wsdl:output>
      <wsdl:fault name="ServiceExceptionReport">
        <soap:fault use="literal" name="ServiceExceptionReport"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="wfs.describeStoredQueries">
      <http:operation location="{id}"/>
      <wsdl:input>
```

```
        <http:urlEncoded/>
      </wsdl:input>
      <wsdl:output>
        <mime:mimeXml/>
      </wsdl:output>
      <wsdl:fault name="ServiceExceptionReport">
        <soap:fault use="literal" name="ServiceExceptionReport"/>
      </wsdl:fault>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

E.6.7 wfs-soap-bindings.wsdl

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<wsdl:definitions
  targetNamespace="http://www.opengis.net/wfs/soap/2.0"
  xmlns:wfs-req="http://www.opengis.net/wfs/requests/2.0"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:date>2008-08-31</dc:date>
    <dc:description>
      Adds SOAP (over HTTP) protocol bindings for WFS 2.0.0 interfaces.
    </dc:description>
  </wsdl:documentation>
  <wsdl:import namespace="http://www.opengis.net/wfs/requests/2.0"
    location="./wfs-xml-interfaces.wsdl"/>
  <wsdl:binding name="wfs-SOAP" type="wfs-req:wfs">
    <wsdl:documentation>
      WFS interface bound to SOAP over HTTP/1.1.
    </wsdl:documentation>
    <soap:binding style="document"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="wfs.getCapabilities">
      <soap:operation
    soapAction="http://www.opengis.net/wfs/requests#GetCapabilities"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
      <wsdl:fault name="ServiceExceptionReport">
        <soap:fault use="literal" name="ServiceExceptionReport"/>
```

```
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.describeFeatureType">
    <soap:operation
  soapAction="http://www.opengis.net/wfs/requests#DescribeFeatureType"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.getPropertyValue">
    <soap:operation
  soapAction="http://www.opengis.net/wfs/requests#GetPropertyValue"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.getFeature">
    <soap:operation
  soapAction="http://www.opengis.net/wfs/requests#GetFeature"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.getFeatureWithLock">
    <soap:operation
  soapAction="http://www.opengis.net/wfs/requests#GetFeatureWithLock"/>
```

```
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.lockFeature">
    <soap:operation
  soapAction="http://www.opengis.net/wfs/requests#LockFeature"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.transaction">
    <soap:operation
  soapAction="http://www.opengis.net/wfs/requests#Transaction"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.listStoredQueries">
    <soap:operation
  soapAction="http://www.opengis.net/wfs/requests#ListStoredQueries"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
```

```
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.describeStoredQueries">
    <soap:operation
  soapAction="http://www.opengis.net/wfs/requests#DescribeStoredQueries"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.createStoredQuery">
    <soap:operation
  soapAction="http://www.opengis.net/wfs/requests#CreateStoredQuery"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="wfs.dropStoredQuery">
    <soap:operation
  soapAction="http://www.opengis.net/wfs/requests#DropStoredQuery"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="ServiceExceptionReport">
      <soap:fault use="literal" name="ServiceExceptionReport"/>
    </wsdl:fault>
  </wsdl:operation>
```

```
  </wsdl:binding>
</wsdl:definitions>
```

E.6.8 辅助文件

E.6.8.1 wfs-kvp.xsd

```
<? xml version = "1.0" encoding = "ISO-8859-1"? >
< xsd:schema
  targetNamespace = "http://www.opengis.net/wfs-kvp/2.0"
  xmlns:wfs-kvp = http://www.opengis.net/wfs-kvp/2.0
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "qualified"
  version = "1.1.0">
  < xsd:annotation >
    < xsd:documentation >
      This file defines the abstract schema of keyword-value pair
      encoding of WFS operations. This is NOT the wire format.
      The types in this file are meant to convey the structure.
    </xsd:documentation >
  </xsd:annotation >
  < xsd:simpleType name = "CommaSeparatedList">
    < xsd:restriction base = "xsd:string">
      < xsd:pattern
        value = "\w(,\w) * |
          ´(´ \w(,\w) * )´|
          ´´´( \w(,\w) * ´)´ (,´(´ \w(,\w) * ´)´) * ´)´ "/>
    </xsd:restriction >
  </xsd:simpleType >
  < xsd:simpleType name = "BBoxType">
    < xsd:restriction base = "xsd:string">
      < xsd:pattern value = "(\d,\d)(,\d,\d) * "/>
    </xsd:restriction >
  </xsd:simpleType >
  < xsd:simpleType name = "All_N_ValueType">
    < xsd:restriction base = "xsd:string">
      < xsd:pattern value = "ALL | \d"/>
    </xsd:restriction >
  </xsd:simpleType >
  < xsd:simpleType name = "All_Some_ValueType">
    < xsd:restriction base = "xsd:string">
      < xsd:enumeration value = "ALL"/>
      < xsd:enumeration value = "SOME"/>
    </xsd:restriction >
  </xsd:simpleType >
  < xsd:simpleType name = "ResolveValueType">
```

```
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="local"/>
      <xsd:enumeration value="remote"/>
      <xsd:enumeration value="both"/>
      <xsd:enumeration value="none"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- GETCAPABILITIES 类型定义 ==========================-->
  <xsd:complexType name="GetCapabilitiesType">
    <xsd:sequence>
      <xsd:element name="service" type="xsd:string" minOccurs="1"
        maxOccurs="1" fixed="WFS"/>
      <xsd:element name="request" type="xsd:string" minOccurs="1"
        maxOccurs="1" fixed="GetCapabilities"/>
      <xsd:element name="acceptversions" type="wfs-kvp:CommaSeparatedList"
        minOccurs="0"/>
      <xsd:element name="sections" type="wfs-kvp:CommaSeparatedList"
        minOccurs="0"/>
      <xsd:element name="updatesequence" type="xsd:string" minOccurs="0"/>
      <xsd:element name="acceptformats" type="wfs-kvp:CommaSeparatedList"
        minOccurs="0"/>
      <xsd:element name="namespace" type="wfs-kvp:CommaSeparatedList"
        minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <!-- DESCRIBEFEATURETYPE 类型定义 ======================-->
  <xsd:complexType name="DescribeFeatureTypeRequestType">
    <xsd:sequence>
      <xsd:element name="service" type="xsd:string" minOccurs="1"
        maxOccurs="1" fixed="WFS"/>
      <xsd:element name="version" type="xsd:string" minOccurs="1"
        maxOccurs="1" fixed="2.0.0"/>
      <xsd:element name="request" type="xsd:string" minOccurs="1"
        maxOccurs="1" fixed="DescribeFeatureType"/>
      <xsd:element name="typeName" type="wfs-kvp:CommaSeparatedList"
        minOccurs="0"/>
      <xsd:element name="outputFormat" type="xsd:string" minOccurs="0"
        default=" application/gml+xml; version=3.2"/>
      <xsd:element name="namespace" type="wfs-kvp:CommaSeparatedList"
        minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <!-- GETPROPERTYVALUE 类型定义 ==========================-->
  <xsd:complexType name="GetPropertyValueRequestType">
```

```
      <xsd:sequence>
        <xsd:element name="service" type="xsd:string" minOccurs="1"
          maxOccurs="1" fixed="WFS"/>
        <xsd:element name="version" type="xsd:string" minOccurs="1"
          maxOccurs="1" fixed="2.0.0"/>
        <xsd:element name="request" type="xsd:string" minOccurs="1"
          maxOccurs="1" fixed="GetPropertyValue"/>
        <xsd:element name="valuereference" type="xsd:string"/>
        <xsd:element name=" count" type="xsd:positiveInteger" minOccurs="0"/>
        <xsd:element name="srsname" type="xsd:anyURI" minOccurs="0"/>
        <xsd:choice>
          <xsd:element name=" resourceid" type="wfs-kvp:CommaSeparatedList"/>
          <xsd:sequence>
<xsd:element name="typenames" type="xsd:string"/>
            <xsd:element name="aliases" type="wfs-kvp:CommaSeparatedList"
              minOccurs="0"/>
            <xsd:choice>
              <xsd:element name="filter" type="xsd:string"/>
              <xsd:element name="bbox" type="wfs-kvp:BBoxType"/>
            </xsd:choice>
          </xsd:sequence>
        </xsd:choice>
        <xsd:element name="resolve" type="wfs-kvp:ResolveValueType"
          minOccurs="0" default="none"/>
        <xsd:element name="resolvePath" type="xsd:string" minOccurs="0"
          default="none"/>
        <xsd:element name="resolveDepth" type="xsd:string" minOccurs="0"
          default="*"/>
        <xsd:element name=" resolveTimeout" type="xsd:positiveInteger"
          minOccurs="0" default="300"/>
        <xsd:element name="namespace" type="wfs-kvp:CommaSeparatedList"
          minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
    <!-- GETFEATURE类型定义=================================-->
    <xsd:complexType name="GetFeatureRequestType">
      <xsd:sequence>
        <xsd:element name="service" type="xsd:string" minOccurs="1"
          maxOccurs="1" fixed="WFS"/>
        <xsd:element name="version" type="xsd:string" minOccurs="1"
          maxOccurs="1" fixed="2.0.0"/>
        <xsd:element name="request" type="xsd:string" minOccurs="1"
          maxOccurs="1" fixed="GetFeature"/>
        <xsd:element name="outputFormat" type="xsd:string" minOccurs="0"
```

```
        default = " application/gml + xml; version = 3.2"/>
      < xsd:element name = " count" type = "xsd:positiveInteger"
        minOccurs = "0"/>
      < xsd:element name = "resulttype" minOccurs = "0" default = "results">
        < xsd:simpleType >
          < xsd:restriction base = "xsd:string">
            < xsd:enumeration value = "results"/>
            < xsd:enumeration value = "hits"/>
          </xsd:restriction >
        </xsd:simpleType >
      </xsd:element >
      < xsd:element name = "featureversion" type = "wfs-kvp:All_N_ValueType"
        minOccurs = "0"/>
      < xsd:element name = "srsname" type = "xsd:anyURI" minOccurs = "0"/>
      < xsd:element name = "propertyname" type = "wfs-kvp:CommaSeparatedList"
        minOccurs = "0"/>
      < xsd:choice >
        < xsd:element name = " resourceid" type = "wfs-kvp:CommaSeparatedList"/>
        < xsd:sequence >
          < xsd:element name = "typenames" type = "wfs-kvp:CommaSeparatedList"/>
          < xsd:element name = "aliases" type = "wfs-kvp:CommaSeparatedList"/>
          < xsd:choice >
            < xsd:element name = "filter" type = "xsd:string"/>
            < xsd:element name = "bbox" type = "wfs-kvp:BBoxType"/>
          </xsd:choice >
        </xsd:sequence >
      </xsd:choice >
      < xsd:element name = "sortby" type = "xsd:string" minOccurs = "0"/>
      < xsd:element name = "resolve" type = "wfs-kvp:ResolveValueType"
        minOccurs = "0" default = "none"/>
      < xsd:element name = "resolvePath" type = "xsd:string" minOccurs = "0"
        default = "none"/>
      < xsd:element name = "resolveDepth" type = "xsd:string" minOccurs = "0"
        default = " * "/>
      < xsd:element name = " resolveTimeout" type = "xsd:positiveInteger"
        minOccurs = "0" default = "300"/>
      < xsd:element name = "namespace" type = "wfs-kvp:CommaSeparatedList"
        minOccurs = "0"/>
    </xsd:sequence >
  </xsd:complexType >
  <! -- GETFEATUREWITHLOCK 类型定义 = = = = = = = = = = = = = = = = = = = = = —>
  < xsd:complexType name = "GetFeatureWithLockRequestType">
    < xsd:sequence >
      < xsd:element name = "service" type = "xsd:string" minOccurs = "1"
```

```
    maxOccurs = "1" fixed = "WFS"/>
  < xsd:element name = "version" type = "xsd:string" minOccurs = "1"
    maxOccurs = "1" fixed = "2.0.0"/>
  < xsd:element name = "request" type = "xsd:string" minOccurs = "1"
    maxOccurs = "1" fixed = "GetFeatureWithLock"/>
  < xsd:element name = "outputFormat" type = "xsd:string" minOccurs = "0"
    default = " application/gml + xml; version = 3.2"/>
  < xsd:element name = " count" type = "xsd:positiveInteger"
    minOccurs = "0"/>
  < xsd:element name = "resulttype" minOccurs = "0" default = "results">
    < xsd:simpleType >
      < xsd:restriction base = "xsd:string">
        < xsd:enumeration value = "results"/>
        < xsd:enumeration value = "hits"/>
      </xsd:restriction >
    </xsd:simpleType >
  </xsd:element >
  < xsd:element name = "expiry" type = "xsd:positiveInteger" minOccurs = "0"/>
  < xsd:element name = "featureversion" type = "wfs-kvp:All_N_ValueType"
    minOccurs = "0"/>
  < xsd:element name = "srsname" type = "xsd:anyURI" minOccurs = "0"/>
  < xsd:element name = "propertyname" type = "wfs-kvp:CommaSeparatedList"
    minOccurs = "0"/>
  < xsd:choice >
    < xsd:element name = " resourceid" type = "wfs-kvp:CommaSeparatedList"/>
    < xsd:sequence >
      < xsd:element name = "typenames" type = "wfs-kvp:CommaSeparatedList"/>
      < xsd:element name = "aliases" type = "wfs-kvp:CommaSeparatedList"/>
      < xsd:choice >
        < xsd:element name = "filter" type = "xsd:string"/>
        < xsd:element name = "bbox" type = "wfs-kvp:BBoxType"/>
      </xsd:choice >
    </xsd:sequence >
  </xsd:choice >
  < xsd:element name = "sortby" type = "xsd:string" minOccurs = "0"/>
  < xsd:element name = "resolve" type = "wfs-kvp:ResolveValueType"
    minOccurs = "0" default = "none"/>
  < xsd:element name = "resolvePath" type = "xsd:string" minOccurs = "0"
    default = "none"/>
  < xsd:element name = "resolveDepth" type = "xsd:string" minOccurs = "0"
    default = " * "/>
  < xsd:element name = " resolveTimeout" type = "xsd:positiveInteger"
    minOccurs = "0" default = "300"/>
  < xsd:element name = "namespace" type = "wfs-kvp:CommaSeparatedList"
```

```
        minOccurs = "0"/>
    </xsd:sequence>
  </xsd:complexType>
  <! — LOCKFEATURERE T 类型定义 = = = = = = = = = = = = = = = = = = = = = = = = = —>
  < xsd:complexType name = "LockFeatureRequestType">
    < xsd:sequence >
      < xsd:element name = "service" type = "xsd:string" minOccurs = "1"
        maxOccurs = "1" fixed = "WFS"/>
      < xsd:element name = "version" type = "xsd:string" minOccurs = "1"
        maxOccurs = "1" fixed = "2.0.0"/>
      < xsd:element name = "request" type = "xsd:string" minOccurs = "1"
        maxOccurs = "1" fixed = "LockFeature"/>
      < xsd:element name = "lockaction" type = "wfs-kvp:All_Some_ValueType"
        minOccurs = "0"/>
      < xsd:element name = "expiry" type = "xsd:positiveInteger" minOccurs = "0"/>
      < xsd:choice >
        < xsd:element name = " resourceid" type = "wfs-kvp:CommaSeparatedList"/>
        < xsd:sequence >
          < xsd:element name = "typenames" type = "wfs-kvp:CommaSeparatedList"/>
          < xsd:choice >
            < xsd:element name = "filter" type = "xsd:string"/>
            < xsd:element name = "bbox" type = "wfs-kvp:BBoxType"/>
          </xsd:choice >
        </xsd:sequence >
      </xsd:choice >
      < xsd:element name = "namespace" type = "wfs-kvp:CommaSeparatedList"
        minOccurs = "0"/>
    </xsd:sequence >
  </xsd:complexType >
  <! — LISTSTOREDQUERIES 类型定义 = = = = = = = = = = = = = = = = = = = = = = = = —>
  < xsd:complexType name = "ListStoredQueriesRequestType">
    < xsd:sequence >
      < xsd:element name = "service" type = "xsd:string" minOccurs = "1"
        maxOccurs = "1" fixed = "WFS"/>
      < xsd:element name = "version" type = "xsd:string" minOccurs = "1"
        maxOccurs = "1" fixed = "2.0.0"/>
      < xsd:element name = "request" type = "xsd:string" minOccurs = "1"
        maxOccurs = "1" fixed = "ListStoredQueries"/>
      < xsd:element name = "namespace" type = "wfs-kvp:CommaSeparatedList"
        minOccurs = "0"/>
    </xsd:sequence >
  </xsd:complexType >
  <! — DESCRIBESTOREDQUERIES 类型定义 = = = = = = = = = = = = = = = = = = = = = = = = —>
  < xsd:complexType name = "DescribedStoredQueriesRequestType">
```

```
  <xsd:sequence>
    <xsd:element name="service" type="xsd:string" minOccurs="1"
      maxOccurs="1" fixed="WFS"/>
    <xsd:element name="version" type="xsd:string" minOccurs="1"
      maxOccurs="1" fixed="2.0.0"/>
    <xsd:element name="request" type="xsd:string" minOccurs="1"
      maxOccurs="1" fixed="DescribedStoredQueries"/>
    <xsd:element name="storedQueryId" type="wfs-kvp:CommaSeparatedList"
      minOccurs="0"/>
    <xsd:element name="namespace" type="wfs-kvp:CommaSeparatedList"
      minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<!-- DROPSTOREDQUERY 类型定义 ========== -->
<xsd:complexType name="DropStoredQueryRequestType">
  <xsd:sequence>
    <xsd:element name="service" type="xsd:string" minOccurs="1"
      maxOccurs="1" fixed="WFS"/>
    <xsd:element name="version" type="xsd:string" minOccurs="1"
      maxOccurs="1" fixed="2.0.0"/>
    <xsd:element name="request" type="xsd:string" minOccurs="1"
      maxOccurs="1" fixed="DropStoredQuery"/>
    <xsd:element name="storedQueryId" type="xsd:string" maxOccurs="1"
      minOccurs="1"/>
    <xsd:element name="namespace" type="wfs-kvp:CommaSeparatedList"
      minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:element name="GetCapabilities" type="wfs-kvp:GetCapabilitiesType"/>
<xsd:element name="DescribeFeatureType"
  type="wfs-kvp:DescribeFeatureTypeRequestType"/>
<xsd:element name="GetPropertyValue"
  type="wfs-kvp:GetPropertyValueRequestType"/>
<xsd:element name="GetFeature" type="wfs-kvp:GetFeatureRequestType"/>
<xsd:element name="GetFeatureWithLock" type="wfs-kvp:GetFeatureRequestType"/>
<xsd:element name="LockFeature" type="wfs-kvp:LockFeatureRequestType"/>
<xsd:element name="ListStoredQueries"
  type="wfs-kvp:ListStoredQueriesRequestType"/>
<xsd:element name="DescribeStoredQueries"
  type="wfs-kvp:DescribeStoredQueriesRequestType"/>
<xsd:element name="DropStoredQuery"
  type="wfs-kvp:DropStoredQueryRequestType"/>
```

```
</xsd:schema>
```

E.6.8.2 wfs.util.xsd

```
<? xml version = "1.0" encoding = "ISO-8859-1"? >
< xsd:schema
  targetNamespace = "http://www.opengis.net/wfs-util/2.0"
  xmlns:util = "http://www.opengis.net/wfs-util/2.0"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "qualified">
  < xsd:annotation >
    < xsd:documentation >
      This file defines the type of the DescribeFeatureType response,
      because WSDL requires that types be explicitly defined in the
      types section, perhaps thru imports.
    </xsd:documentation >
  </xsd:annotation >
  < xsd:import namespace = "http://www.w3.org/2001/XMLSchema"
        schemaLocation = "http://www.w3.org/2001/XMLSchema.xsd"/>
  < xsd:complexType name = "DescribeFeatureTypeResponseType">
    < xsd:sequence >
      < xsd:element ref = "xsd:schema"/>
    </xsd:sequence >
  </xsd:complexType >
  < xsd:element name = "DescribeFeatureTypeResponse"
      type = "util:DescribeFeatureTypeResponseType"/>
</xsd:schema >
```

E.6.9 示例

E.6.9.1 概述

E.6.9.2 – E.6.9.5 中包含用于定义 WFS 的端点的样例文件,其 WFS 支持 HTTP GET,POST 和 SOAP(普通 XML)。

E.6.9.2 示例-endpoints.wsdl 文件

```
<? xml version = "1.0" encoding = "ISO-8859-1"? >
< wsdl:definitions
  targetNamespace = "http://www.myservice.com/wfs/2.0"
  xmlns:wsdl = "http://schemas.xmlsoap.org/wsdl/"
  xmlns:http = "http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:wfs-http = "http://www.opengis.net/wfs/http/2.0"
  xmlns:wfs-http-kvp = "http://www.opengis.net/wfs/http/kvp/2.0"
  xmlns:wfs-soap = "http://www.opengis.net/wfs/soap/2.0"
  xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/">
  < wsdl:documentation xmlns:dc = "http://purl.org/dc/elements/1.1/">
    < dc:date > 2008-08-31 </dc:date >
```

```
    <dc:description>
      This WSDL document defines the service-specific properties
      of a MyService WFS implementation; it specifies available
      endpoints and alternative bindings.
    </dc:description>
  </wsdl:documentation>
  <wsdl:import namespace="http://www.opengis.net/wfs/soap/2.0"
    location="./wfs-soap-bindings.wsdl"/>
  <wsdl:import namespace="http://www.opengis.net/wfs/http/2.0"
    location="./wfs-http-bindings.wsdl"/>
  <wsdl:import namespace="http://www.opengis.net/wfs/http/kvp/2.0"
    location="./wfs-kvp-bindings.wsdl"/>
  <wsdl:service name="WFS-www.myservice.com">
    <wsdl:documentation>
      A WFS-2.0 implementation. Includes alternative SOAP bindings
      for the WFS interfaces.
    </wsdl:documentation>
    <wsdl:port name="wfs-SOAP-Port" binding="wfs-soap:wfs-SOAP">
      <soap:address
        location="http://www.myservice.com/wfs-soap/services/WFSSOAP"/>
    </wsdl:port>
    <wsdl:port name="wfs-POST-Port" binding="wfs-http:wfs-POST">
      <http:address location="http://www.myservice.com/"/>
    </wsdl:port>
    <wsdl:port name="wfs-GET-Port" binding="wfs-http-kvp:wfs-GET">
      <http:address location="http://www.myservice.com/"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

E.6.9.3 示例-POST-endpoints.wsdl 文件

```
<? xml version="1.0" encoding="ISO-8859-1"? >
<wsdl:definitions
  targetNamespace="http://www.myservice.com/wfs/2.0"
  xmlns:wfs-http="http://www.opengis.net/wfs/http/2.0"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/">
  <wsdl:documentation xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:date>2008-08-31</dc:date>
    <dc:description>
      This WSDL document defines the service-specific properties
      of a MyService WFS implementation; it specifies available
      endpoints and alternative bindings.
    </dc:description>
  </wsdl:documentation>
```

```
  <wsdl:import namespace="http://www.opengis.net/wfs/http/2.0"
          location="./wfs-http-bindings.wsdl"/>
  <wsdl:service name="WFS-www.myservice.com">
    <wsdl:documentation>A WFS-2.0 implementation.</wsdl:documentation>
    <wsdl:port name="wfs-POST-Port" binding="wfs-http:wfs-POST">
      <http:address location="http://www.myservice.com/"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

E.6.9.4 示例-GET-endpoints.wsdl 文件

```
<? xml version="1.0" encoding="ISO-8859-1"? >
<wsdl:definitions
  targetNamespace="http://www.myservice.com/wfs/2.0"
  xmlns:wfs-http="http://www.opengis.net/wfs/http/kvp/2.0"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:date>2008-08-31</dc:date>
    <dc:description>
      This WSDL document defines the service-specific properties
      of a MyService WFS implementation; it specifies available endpoints
      and alternative bindings.
    </dc:description>
  </wsdl:documentation>
  <wsdl:import namespace="http://www.opengis.net/wfs/http/kvp/2.0"
    location="./wfs-kvp-bindings.wsdl"/>
  <wsdl:service name="WFS-www.myservice.com">
    <wsdl:documentation>A WFS-2.0 implementation.</wsdl:documentation>
    <wsdl:port name="wfs-GET-Port" binding="wfs-http:wfs-GET">
      <http:address location="http://www.myservice.com/"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

E.6.9.5 示例-SOAP-endpoints.wsdl 文件

```
<? xml version="1.0" encoding="ISO-8859-1"? >
<wsdl:definitions targetNamespace="http://www.myservice.com/wfs/2.0"
  xmlns:wfs-soap="http://www.opengis.net/wfs/soap/2.0"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <wsdl:documentation xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:date>2008-08-31</dc:date>
    <dc:description>This WSDL document defines the service-specific properties
```

```
        of a MyService WFS implementation; it specifies available
        endpoints and alternative bindings.</dc:description>
    </wsdl:documentation>
    <wsdl:import namespace="http://www.opengis.net/wfs/soap/2.0"
      location="./wfs-soap-bindings.wsdl"/>
    <wsdl:service name="WFS-www.myservice.com">
      <wsdl:documentation>A WFS-2.0 implementation. Includes alternative SOAP bindings for
the WFS interfaces.</wsdl:documentation>
      <wsdl:port name="wfs-SOAP-Port" binding="wfs-soap:wfs-SOAP">
        <soap:address location="http://www.myservice.com/wfs-soap/services/WFSSOAP"/>
      </wsdl:port>
    </wsdl:service>
  </wsdl:definitions>
```

附 录 F
（资料性附录）
抽象模型

F.1 概述

本附录为 WFS 提供了丰富的抽象模型。WFS 是一个服务，用户能够通过该服务选取由该服务管理的资源子集。假定每个 WFS 实例与一个资源集相关联。

抽象模型(AM)定义了包含在 WFS 请求和响应中的信息，并且定义了在 OGC 请求中所有允许的表达式的值。

F.2 抽象资源模型

F.2.1 概述

本附录介绍资源的抽象模型，该模型独立于 WFS。一组这样的资源总是与一个 WFS 相关联，通过一组针对这些资源的操作(见 F.6)定义 WFS 的行为。

一个资源就是一个集合中的一个元素，所有资源包含 0 个或多个描述资源的特性，特性值总是另外一个资源。但是一些资源是不能拥有特性的(例如一个简单的字符串值)。

资源根据它们拥有指定的一系列特性来进行分类，即拥有相同的一系列特性的资源属于同一个类型。

特性的值可以是文字的或非文字的资源，有时候将特性看作资源的函数非常有用，例如，资源 Road(道路)的 numLanes(车道数目)特性可以写成一个函数 numLanes(Road)|→3.

多种不同类型的资源可以组成一个资源集，WFS 的目的是允许用户选择指定的资源集中的一个子集，资源集中的资源即是 ISO 19109 定义的 ISO 通用要素模型中的要素。这里，抽象模型采用术语“资源”而非“要素”是希望该模型既可以阐述类型又可以描述地理要素。但是，当描述 WFS 时可以认为资源等同于要素。

本抽象模型由 F.2.2 所描述的存取器函数(Accessor Function)正式定义。

F.2.2 基本存取函数

下面为描述资源的基本存取函数：

- 基于资源和特性的存取

Name (Resource | Property) → String (名称(资源|特性)→字符串)

name()存取函数返回资源和特性的名称。对于 WFS，名称可以是指要素名称(例如 abc:Road)或特性名称(例如 abc:numLanes)。

- 基于资源的存取

1) id (Resource) → String | Null (标识符(资源)→字符串|空)

id()存取函数返回资源的标识符。

2) typeName (Resource) → TypeNameString (类型名称(资源)→类型名称字符串)

返回的 TypeNameString 是资源的类型名称，并且应完整地标识与资源相关的特性清单，也可以写成 Type (Feature) → TypeNameString。

3) typeDef (Resource) → type definition (类型定义(资源)→类型定义)

一个类型定义包含丰富的信息,以完整地标识资源的所有特性。

4) versionNumber (Resource) → numeric value (版本号(资源)→数字值)(资源的版本)

5) versionTimestamp (Resource) → timestamp value 版本时间戳(资源)→时间戳的值

6) properties (Resource) → Property * (特性(资源)→特性 *)

properties()存取函数返回资源 property (Resource,PropertyName) → Property * (特性(资源,特性名称)→特性 *)的所有特性:

property()存取应用于资源,返回 0 个或者多个指定特性名称的特性。

PropertyName 是一个请求的特性名称的字符串值。注意一个资源可以拥有相同 PropertyName 的多个实例(见后面 cardinality(关联基数))。

- 基于特性的存取

1) valueOf (property) → Value + (求值(特性)→值+)

valueOf()存取器应用于一个特性,返回这个特性的值,特性值设置为一个或者多个资源。这些资源可以是非文字或者文字。

例如,要素 Road (见 F.2.1)的特性函数"numLanes"可以表示为:

numLanes (Road) → valueOf(Property (Road,numLanes))

2) typeDef(Property)→type definition (类型定义(特性)→类型定义)

一个类型定义包含足够的信息,以完整地标识作为特性值的资源的类型定义。

示例:有一个拥有特性 numLanes,heightAtCenter,span 和 crosses 的资源 LionsGateBridge。同样还有一个资源 BurrardInlet,它是 crosses 特性的值。即:crosses (LionsGateBridge) → BurrardInlet。那么 TypeDef (crosses) → TypeDef (BurrardInlet)。

3) cardinality (Property) → multiplicity (关联基数(特性)→多次出现)

一个特性可以在资源中出现多次,一个特性在资源中可能出现的次数由 cardinality()函数决定。

multiplicity 是一个排列类型,由一组非负整数定义,其值可以为{0 或多个,1 或多个,n(一个非负整数),(n,m)(从 n 到 m)}中的一个。

F.3 通用要素模型(General Feature Model,GFM)到 WFS 抽象模型的映射

WFS 抽象模型是针对资源而言的,这与 GFM 针对的要素等同。在该附录后文中所有类似的资源都是指要素。

F.4 标识符

一个 GML 对象是一个应用模式中声明的一个元素,此应用模式的内容模型直接或间接源自 gml:AbstractGML 的内容模式。每个 GML 对象都有一个必选的 gml:id。gml:id 应是一个 WFS 资源的固定标识符,除了插入操作,它不能被其他 WFS 操作改变,并且如果相关资源被删除,其 gml:id 不能重新赋予其他资源。gml:id 由拥有这个资源的 WFS 控制。可以在一个领域中定义 gml:id 的全局唯一值。

F.5 valueOf()函数

valueOf()存取器函数提供在决定特性值时可能会遇到的任何引用的解析。例如,在 GML 中,一个特性可能拥有通过一个 xlink:href 指定的特性值。这种情况下,valueOf()函数应有效地遍历和解析所有这样的引用。

F.6 WFS 操作

F.6.1 概述

WFS 的行为由一个要素(资源)集合(见 C.2)定义,在该要素集合上定义了下面一组抽象存取器函数。

a) featureTypeNameList()(要素类型名称列表)
b) featureType()(要素类型)
c) feature()(要素)
d) propertyValue()(特性值)
e) lock()(锁定)
f) transaction()(事务)
   1) insert()(插入)
   2) update()(更新)
   3) delete()(删除)
   4) replace()(替换)
g) stored query ops(存储的查询操作)
   1) createStoredQuery()(创建存储的查询)
   2) dropStoredQuery()(删除存储的查询)
   3) listStoredQueries()(列举存储的查询)
   4) describeStoredQueries()(描述存储的查询)

这些函数与 valueOf()函数(见 F.5)完整地描述了 WFS 的行为。

F.6.2 featureTypeNameList()函数

该函数返回一个服务支持的要素类型的名称列表(TypeName(Feature))。

在 GML 中,这是一列 QName(XML 模式类型 QName,例如 myns:address)。这个函数没有任何自变量。

F.6.3 featureType()函数

TFeatureDef 表示与 WFS 相关的要素类型的定义(见 F.2.2 中 TypeDef 存取器函数)。

featureType (TypeNameString *) → type definition *,其中 type definition 属于 $T_{FeatureDef}$

TypeNameString 是表示由 Type(Resource)函数返回的 TypeName 的字符串。

这个函数返回对要素类型名称的要素类型定义(以及对这些要素类型的所有特性的定义)。

此要素类型名称字符串集(函数自变量)可以为空,这样的情况下将返回一个完整的包含服务支持的所有要素类型定义的 $T_{FeatureDef}$ 集。

在一个返回 GML 的 WFS 中,要素类型名称为 QName 类型,并且返回的类型定义集使用 GML 应用模式来表达。

F.6.4 查询函数

F.6.4.1 feature() 函数

feature (Query +) → Feature * (要素(查询+)→要素 *)

feature()函数输入一个或多个查询自变量,返回 0 个或多个要素。

F.6.4.2 查询表达式

Query:: =AdHocQuery | StoredQuery(即席查询|存储的查询)

一个查询(Query)可以为一个即席查询表达式或者一个参数化的存储的查询表达式。不管是哪一种情况,一个查询返回 0 个或多个要素。

F.6.4.3 即席查询

F.6.4.3.1 通用表达式

AdHocQuery::=Sort • Resolve • Project • Filter (即席查询::=排序 • 解析 • 映射 • 过滤)

F.6.4.3.2 排序

Sort(Feature+, ValueReference, ASC | DESc) +) → Feature+ (排序(要素+,(引用值,升序|降序)+)→要素+

此排序命令由一个或者多个 Sort 表达式指定。每个 Sort 表达式包含一个值引用,要素根据这个值引用进行排序;同时还包含一个排序调节器,用来指明返回结果是以升序(ASC)还是降序(DESC)进行排序。默认值为 ASC。

如果 WFS 是基于 XML 实现的,那么 ValueReference 是一个 XPath 表达式。

F.6.4.3.3 引用解析

Resolve: (Feature +, resolutionDepth) → Feature + (解析:(要素+,解析深度)→要素+)

与 WFS 相关联的要素集可以包含资源,其资源特性可以引用要素集内的其他资源,也可以引用不由此 WFS 管理的外部资源。此函数输入一个要素,然后对每个特性使用 ValueOf()函数。因为一个特性值可能包含更深层次的引用,所以这个过程是重复的(即接下来 ValueOf()函数被应用到 Property 值中的所有资源的所有特性),直到 resolutionDepth 指定的层次。

F.6.4.3.4 映射

Project: (Feature +, desiredProperties) → Feature+ (映射:(要素+,所需特性)→要素+

如 F.2.2 所述,一个要素可以拥有某些特性,这些特性的多重性是 0 或多个。这些特性可以是可选的,并可以通过使用 Project 函数忽略这些特性。

Project 是获取一个或多个要素的函数,其自变量 desiredProperties 指定从返回要素的特性集得到所想要的多重性为 0 或多个的特性。总是返回所有必选的特性(multiplicity=(1, 1 或多个, n(n>0), (n, m), n>0)。

如果 WFS 是使用 XML 实现的,那么使用 XPath 表达式指定的 desiredProperties。

F.6.4.3.5 过滤器

Filter 是一个在抽象查询模型中定义的过滤器函数。它应用于与 WFS 相关联的要素集,用来选择满足查询表达式中包含的过滤器表达式的所有要素。

F.6.4.4 存储的查询表达式

StoredQuery: (QueryIdentifier, QueryParameter *) —> Feature + (存储的查询:(查询标识符,查询参数 *)→要素+)

一个 StoredQuery 由 QueryIdentifier 标识。它可被认为是一个函数,其自变量由相关联的

QueryParameters 表规定。它返回一组要素。

F.6.5 propertyValue()函数

propertyValue (Feature +, Query) → Value * (特性值(要素+,查询)→要素*)

这个函数返回满足函数自变量中指定 Query 的 Property 值。函数 propertyValue()的语义是由函数 Resolve(),Prune()和 Query()共同确定。即:

propertyValue::= Resolve • Prune • Query (特性值::=解析 • 删减 • 查询)

查询依次遍历适用的要素并返回满足所包含的过滤器表达式(参见 F.6.4.3.5)的所有要素。

Prune()函数输入 Query 返回的要素集和提供的指定所需特性的 XPath 表达式,返回 Property 值。

prune: (Feature +, XPathExp) → Value + (删减:(要素+,XPath 表达式)→值

接下来 Resolve 函数遍历返回的要素,对这些要素的所有特性使用 valueOf()函数。

F.6.6 Lock()函数

lock: (Query) → Feature * (锁定:(查询)→要素*)

lock 函数锁定满足提供的 Query(见 F.6.4.2)的要素集。

接下来,一个封装的函数可以对返回的要素集使用 id()函数,生成一个要素标识符集。

锁定要素集可以阻止任何 WFS 客户端对这些要素进行事务操作,但是获得权限的客户端除外。

F.6.7 transaction()函数

F.6.7.1 通用表达式

transaction: (insert *,update *,delete *,replace *) → boolean (事务:(插入*,更新*,删除*,替换*)→布尔)

transaction 函数自动实现指定的 insert,update 和 delete 系列操作。如果系列操作都成功实现,那么它返回“true”,否则返回“false”。

F.6.7.2 insert()插入函数

insert: (Feature +) → boolean (插入:(要素+)→布尔)

insert 函数在 WFS 要素集中添加一个或多个要素。

F.6.7.3 update()更新函数

update 函数可以 replace(替换)、remove(移除)或 insert(插入)要素集的特性值。

update: (UpdateOperator +, Filter) → boolean (更新:(更新操作+,过滤器)→布尔

Filter 选择满足过滤器条件的要素集,然后由更新操作更新这些要素。每一个 UpdateOperator 可以是一个 Replace,Remove 或 Insert:

UpdateOperator:: =Replace | Remove | Insert (更新操作::=替换|删除|插入)

- Replace (Feature, PropertyReference, NewProperty) → Feature (替换(要素,特性引用,新特性)→要素)

PropertyReference 是对提供的要素特性的引用。

对于一个 WFS 实例,PropertyReference 在具体的模型中可以实现为一个指定的路径表达式(例如 XPath 或对象路径)。Replace 操作使用提供的新特性更换引用的特性。

- Remove (Feature, PropertyReference) → Feature (删除(要素,特性引用)→要素)

Delete 操作删除给定要素的指定特性(由 PropertyReference 指定)。

- Insert (Feature, PropertyReference, NewProperty, (Before | After) ?) → Feature (插入(要素,特性引用,新特性,(之前|之后)?)→要素)

insert 操作给要素插入提供 NewProperty 特性。(Before | After)指定相对于引用特性的插入位置。位置区分符的缺省表示新特性在引用的特性之后插入。

F.6.7.4 delete()删除函数

delete (Filter) → boolean (删除(过滤器)→布尔)

delete 函数删除满足查询条件的所有要素。

F.6.7.5 replace()替换函数

replace (Feature +, Filter) → boolean 替换(要素+,过滤器)→布尔

replace 函数为 WFS 添加提供的要素,并赋予插入的要素的版本值,版本值是在过滤器所标识的要素的基础上计算得到的版本值,插入的要素为标识的要素的最新版本。

F.6.8 存储的查询表达式的操作

F.6.8.1 createStoredQuery()函数

createStoredQuery (QueryIdentifier,returnType +, QueryParameter *, Query) →boolean (创建存储的查询(查询标识符,返回类型+,查询参数*,查询)→布尔)

createStoredQuery( )函数定义了一个要存储的在服务器数据库中的参数化查询。QueryIdentifier 参数指定存储的查询存储在服务器数据库时的标识符。returnType 参数指定存储的查询返回的要素类型。QueryParameters 参数指定 0 个或多个自变量,当进行存储的查询时,将使用这些自变量。Query 参数指定服务器存储的查询表达式。函数返回"TRUE"表示存储的查询表达式已经成功保存。

F.6.8.2 dropStoredQuery()函数

dropStoredQuery (QueryIdentifier) → boolean (删除存储的查询(查询标识符)→布尔)

dropStoredQuery( )函数从服务器数据库中移除带有指定 QueryIdentifier 的存储的查询。函数返回"TRUE"表示存储的查询已成功移除。

F.6.8.3 listStoredQueries()函数

listStoredQueries( )→list of indetifier (列举存储的查询()→标识符列表)

listStoredQueries( )函数返回一列存储在服务器数据库中的存储的查询标识符。

F.6.8.4 describeStoredQueries()函数

describeStoredQueries (QueryIdentifier *) → list of query expression descriptions (描述存储的查询(查询标识符*)→查询表达式描述列表)

describeStoredQueries()函数提供存储在服务器数据库中的查询表达式的详细描述。QueryIdentifier 参数指定描述的一个或者多个存储的查询的标识符。如果没有指定任何 QueryIdentifier,那么将描述所有的存储的查询表达式。

F.7 WFS 操作

表 F.1 将 WFS 抽象模型映射到本标准指定章和子章条定义的 WFS 操作。

表 F.1 将 WFS 抽象数据模型映射到 WFS 操作

抽象模型函数	WFS 操作	章 条
featureTypeNameList()	GetCapabilities	8
featureType()	DescribeFeatureType	9
feature()	GetFeature	11
propertyValue()	GetPropertyValue	10
lock()	Lock / GetFeatureWithLock	12、13
transaction()	Transaction	15
createStoredQuery()	CreateStoredQuery	14.5
dropStoredQuery()	DropStoredQuery	14.6
listStoredQueries()	ListStoredQueries	14.3
describeStoredQueries()	DescribedStoredQueries	14.4

F.8 概念模式

WFS 接口的概念模式是基于本目录和目录 E，ISO 19143:2010 指定的抽象查询模式，使用 UML 记号表达。图 F.1 描述了 WFS 的基本接口。

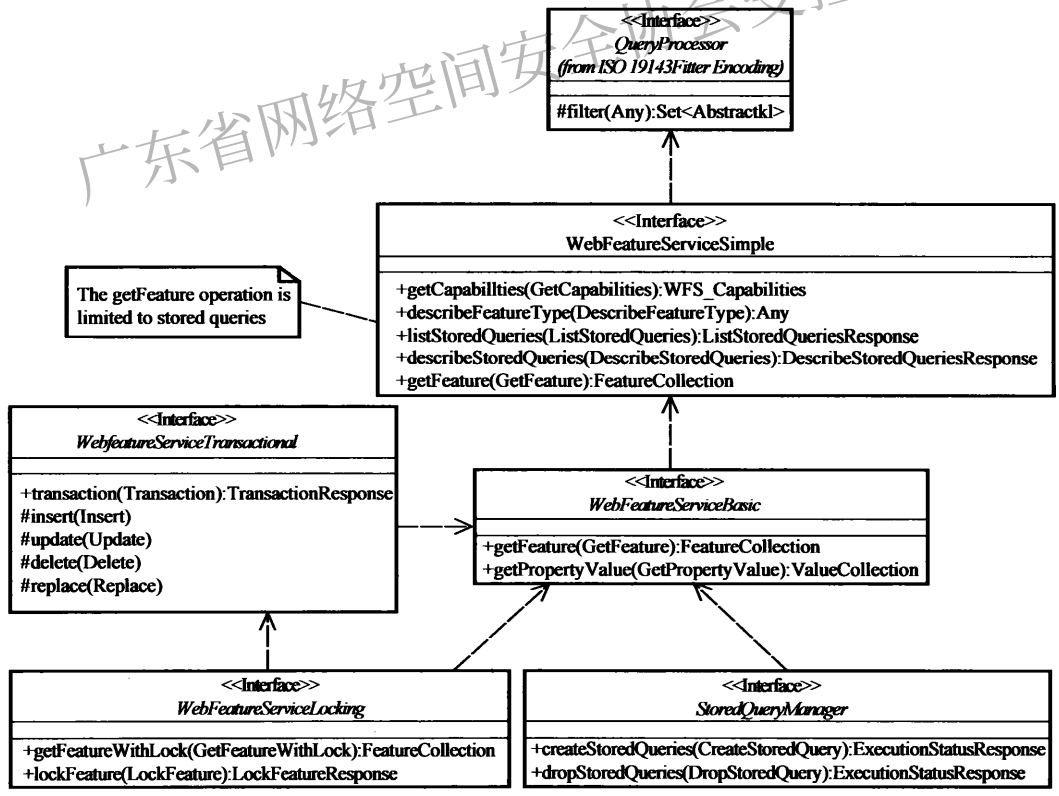

图 F.1 WFS 接口概述

附 录 G
（资料性附录）
操作及参数名称的中英文对照

G.1 概述

为了方便本标准的使用，本章列举本标准中定义的主要操作名称及其主要参数或元素名称的中英文对照表。

G.2 主要操作名称的中英文对照

英文名称	中文名称	章 条
GetCapabilities()	获取服务能力	8
DescribeFeatureType ()	描述要素类型	9
GetPropertyValue()	获取特性值	10
GetFeature()	获取要素	11
LockFeature ()	锁定要素	12
GetFeatureWithLock ()	获取要素并锁定	13
Transaction ()	事务操作	15
CreateStoredQuery ()	创建存储的查询	14.5
DropStoredQuery ()	删除存储的查询	14.6
ListStoredQueries ()	列举存储的查询	14.3
DescribeStoredQueries ()	描述存储的查询	14.4

G.3 主要参数和元素名称的中英文对照

参数的英文名称不分大小写，但是其参数值区分大小写。

英文名称	中文名称
Abstract	摘要
AbstractQueryExpression	抽象的查询表达式
AcceptFormats	可接受的格式
AcceptVersions	可接受的版本
BBOX	外接矩形
Count	计数
DefaultCRS	缺省空间参照系
EXPIRY	期限

（续）

英文名称	中文名称
ExtendedDescription	扩展的描述
FeatureTypeList	要素类型列表
Filter	过滤器
Filter_Language	过滤器_语言
inputFormat	输入格式
IsPrivate	是否私有
Keywords	关键字
language	语言
LOCKACTION	锁定动作
lockId	锁标识符
Locking	锁定
MetadataURL	元数据资源标识地址
Name	名称
Namespaces	命名空间
NoCRS	没有坐标参照系
OtherCRS	其他空间参照系
OutputFormat	输出格式
Projection Clause	映射子句
PropertyName	特性名称
QueryExpressionText	查询表达式文本
releaseAction	释放动作
Resolve	解析
ResolveDepth	解析深度
ResolveTimeout	解析终止时间
ResourceId	资源标识符
RESOVEPATH	解析路径
Resulttype	结果类型
SortBy	排序
SRSName	空间参照系名称
StandardInputParameters	标准输入参数
StandardPresentationParameters	标准表达参数
StandardResolveParameters	标准解析参数
Startindex	起始索引
state	状态

（续）

英文名称	中文名称
StoredQuery_ID	存储的查询表达式标识符
STOREDQUERY_ID	存储的查询标识符
storedquery_parameter	存储的查询参数
Title	标题
Typename	类型名称
VALUEREFERENCE	值引用
vendorId	提供商的标识符
Vendor-specific parameters	提供商指定参数
Version	版本号
Wfs_capabilities	WFS_服务能力
WGS84BoundingBox	WGS84 外接矩形

参 考 文 献

[1] GB/T 17694—2009 地理信息 术语 (ISO 19104:2008,IDT)

[2] GB/T 19710—2005 地理信息 元数据(ISO 19115,MOD)

[3] GB/T 25597—2010 地理信息 网络地图服务接口(ISO 19142—2010,IDT)

[4] ISO 8601:2004 Data elements and interchange formats Information interchange Representation of dates and times

[5] ISO 19101:2002 Geographic information Reference model

[6] ISO 19109:2005 Geographic information Rules for application schema

[7] ISO 19111:2007 Geographic information Spatial referencing by coordinates

[8] ISO 19119:2005 Geographic information Services

[9] CGI/1.1 The Common Gateway Interface, National Centre for Supercomputing Applications

[10] IETF RFC 2045 Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies (November 1996)

[11] IETF RFC 2396 Uniform Resource Identifiers (URI), Generic Syntax, (August 1998)

[12] OGC 02-058 Web Feature Service V1.0.0 (17 May 2002)

[13] OGC 04-094 Web Feature Service Implementation Specification V1.1.0 (03 May 2005)

[14] OGC 07-092r2 Definition identifier URNs in OGC namespace (22 August 2008)

[15] OGC 09-144r1 MIME Media Types for GML, OGC® Policy Document (08 February 2010)

[16] W3C XML Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation (4 February 2004)

[17] W3C XLink, XML Linking Language (XLink) Version 1.0, W3C Recommendation (27 June 2001)

中华人民共和国
国家标准
地理信息 基于网络的要素服务
GB/T 30169—2013/ISO 19142:2010
*
中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100029)
北京市西城区三里河北街16号(100045)
网址 www.spc.net.cn
总编室:(010)64275323 发行中心:(010)51780235
读者服务部:(010)68523946
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230 1/16 印张 18.5 字数 522 千字
2014年5月第一版 2014年5月第一次印刷
*
书号:155066·1-48904 定价 140.00 元

GB/T 30169-2013