DigitalGlobe

Web Feature Service Developer Guide

Cloud Services | August 2013

Table of Contents

List of Figures **3**
List of Tables **4**
1 Introduction **5**
1.1 About This Document 5
1.2 Targeted Audience 5
1.3 What is WFS? 5
1.4 References 5
2 Open Geospatial Consortium (OGC) **6**
2.1 About OGC 6
2.2 The OGC Process 6
2.3 OGC Standards and Specification 6
2.4 OGC Standards 7
2.5 Abstract Specification 7
2.6 OGC Reference Model (ORM) 7
3 Geography Markup Language (GML) **8**
3.1 Introduction to GML 8
3.2 Overview of GML Schema 8
3.3 GML Schema Features 9
3.3.1 AbstractFeatureType 9
3.3.2 AbstractFeature 9
3.4 Overview 9
3.4.1 Dictionary schema 10
3.5 GML-Based Request 13
4 Web Feature Service **15**
4.1 Introduction 15
4.2 WFS Client-Server Architecture 15
4.3 WFS Service Details 15
4.4 Basic Service Elements 16
4.4.1 HTTP Request 16
4.4.2 HTTP Response 19
4.4.3 Request Parameters 20
4.4.4 Request Parameter Rules 22
4.5 Integration Procedure 22
4.5.1 Using WFS-GetFeature to Access Additional Metadata 27
4.5.2 Using WFS TileMatrixFeature Request 28
4.5.3 Using WFS FinishedFeature Request 28
4.6 Service Exceptions 31
4.7 WFS Layers 31
4.8 API Reference 32
5 Common Query Language **38**
5.1 CQL Filter Principle 38
5.2 How to Combine Two Attributes in a CQL Filter 38
Glossary **42**
Index **44**

List of Figures

Figure 4.1 A Typical Structure of a DGCS-WFS Application 15
Figure 4.2 Sample WFS Client-Server Application 15
Figure 4.3 Sample WFS Client-Server Application 18
Figure 4.4 BBOX with Output SRS Specified 21
Figure 4.5 BBOX with Input SRS Specified 22

List of Tables

Table 2.1 OGC Document Types....6
Table 4.1 A General GET Request....18
Table 4.2 Reserved Characters in HTTP Get Query....18
Table 4.3 Values for Output Format Attribute....20
Table 4.4 OGC Layers....31
Table 4.5 Understanding URL Parameters for WFS GetCapabilities Request....32
Table 4.6 GetFeature Request Parameters....32
Table 4.7 DescribeFeatureType Request Parameters....33
Table 4.8 Elements of Feature Type “FinishedFeature”....34
Table 4.9 Elements of FeatureType ImageinMosaice Feature....36
Table 5.1 CQL Filter Parameters....39

1 Introduction

1.1 About This Document

This document covers the concepts of Web Feature Service (WFS), Open Geospatial Consortium (OGC) standards for WFS, capabilities of WFS and ways to integrate DigitalGlobe Cloud Services (DGCS)-WFS in GIS-based custom application development.

1.2 Targeted Audience

This document is targeted to help developers of GIS-based custom application development. Developers new to WFS can read about the DGCS-WFS framework, capabilities, integration procedures and development best-practices to design methods for creating innovative world-class GIS applications.

1.3 What is WFS?

The OGC Web Feature Service Interface Standard (WFS) provides an interface allowing requests for geographical features across the web using platform-independent calls.

The basic Web Feature Service allows querying and retrieval of features. A transactional Web Feature Service (WFS-T) allows creation, deletion, and updating of features.

The WFS describes discovery, query, or data transformation operations. The client generates the request and posts it to a web feature server using HTTPS. The web feature server then executes the request. The WFS specification uses HTTPS as the distributed computing platform, although this is not a hard requirement.

1.4 References

- http://www.opengeospatial.org/standards
- http://en.wikipedia.org/wiki/Web_Feature_Service
- http://en.wikipedia.org/wiki/GIS#OGC_standards
- http://www.wikipedia.org/
- http://en.wikipedia.org/wiki/Geography_Markup_Language

2 Open Geospatial Consortium (OGC)

2.1 About OGC

The OGC is an international voluntary consensus standards organization that originated in 1994. In the OGC, more than 400 commercial, governmental, nonprofit and research organizations worldwide collaborate in a consensus process encouraging development and implementation of open standards for geospatial content and services, GIS data processing and data sharing.

A predecessor organization, the Open GRASS Foundation (OGF), started in 1992. From 1994 to 2004 the organization also used the name Open GIS Consortium.

2.2 The OGC Process

The OGC exists to enable a fast, effective, inclusive, user-driven process to develop, test, demonstrate, and promote the use of geospatial information and services by using OpenGIS® Standards.

The OGC has defined the standards around different GIS Cloud Services by following the process of identifying and addressing existing problems in the GIS industry. The process is:

1. Identify the Problem
2. Craft a Solution
3. Evaluate the Proposed Solution
4. Implement Standards

One of the major problems identified and addressed is interoperability. The following were discussed and prioritized as part of identifying and addressing the interoperability problem.

- Sharing maps on the Web.
- Delivering data to different systems easily.
- Common language to speak about geospatial data or services.
- Finding and pulling together data from our automated sensors.

2.3 OGC Standards and Specification

OGC® Standards and Specifications are technical documents that detail interfaces or encodings. Software developers use these documents to build support for the interfaces or encodings into their products and services. These specifications are the main “products” of the Open Geospatial Consortium and have been developed by the membership to address specific interoperability challenges. The OGC documents are offered at no cost and are available to everyone. Refer to Table 2.1 for a list of documents currently available on the OGC website.

TABLE 2.1 OGC DOCUMENT TYPES

OGC DOCUMENT TYPE	DESCRIPTION
OpenGIS Implementation Standard	A document containing an OGC consensus, technology-dependent standard for application programming interfaces and related standards based on the Abstract Specification or domain-specific extensions to the Abstract Specification. There are five subtypes: Interface, Encoding, Profile, Application Profile, and Application Schema.
Abstract Specification	A document (or set of documents) containing an OGC consensus, technology-independent standard for application programming interfaces and related standards based on object-oriented or other IT-accepted concepts. It describes and/or models an application environment for interoperable geoprocessing and geospatial data and services products.
Best Practices	A document referring to the use and/or implementation of an adopted OGC document. Best Practices Documents are an official position of the OGC and thus represent an endorsement of the content of the paper.
Discussion Papers	A document containing discussion of some technology or standard area for release to the public. Discussion Papers are not the official position of the OGC and contain a statement to that effect.
White Papers	A publication released by the OGC to the Public that states a position on a social, political, technical or other subject, often including a high-level explanation of an architecture or framework of a solution.

2.4 OGC Standards

OGC Standards are written for a more technical audience and detail the interface structure between software components. An interface specification is considered to be at the implementation level of detail if, when implemented by two different software engineers in ignorance of each other, the resulting components plug and play with each other through that interface.

2.5 Abstract Specification

The OGC Technical Committee (TC) has developed architecture in support of its vision of geospatial technology and data interoperability called the OGC Abstract Specification. The Abstract Specification provides the conceptual foundation for most OGC specification development activities. Open interfaces and protocols are built and referenced against the Abstract Specification, thus enabling interoperability between different brands and different kinds of spatial processing systems. The Abstract Specification provides a reference model for the development of OGC Implementation Specifications.

2.6 OGC Reference Model (ORM)

The OGC Reference Model (ORM) provides a framework for the ongoing work of the OGC. The ORM describes the OGC Standards Baseline (SB) focusing on the relationships between the OpenGIS Specification documents. The OGC SB consists of the approved OGC Abstract and Implementation Specifications as well as OGC Best Practices documents. Best Practices documents are official positions of the OGC members and quite often are provided as supporting technical information for the adopted Specifications.

Advantages or the purposes of ORM are:

- Provides an overview of OGC Standards Baseline
- Provides insight into the current state of the work of the OGC
- Serves as a basis for coordination and understanding of the documents in OGC SB
- Provides a useful resource for defining architectures for specific applications

➔ For detailed information on OGC standards and specifications, visit http://www.opengeospatial.org/standards.

3 Geography Markup Language (GML)

3.1 Introduction to GML

The Geography Markup Language (GML) is the XML grammar defined by the Open Geospatial Consortium (OGC) to express geographical features. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet. Note that the concept of feature in GML is a very general one and includes not only conventional "vector" or discrete objects, but also coverages and sensor data. The ability to integrate all forms of geographic information is key to the utility of GML.

GML was conceived and evolved for a variety of reasons, the most important being:

- To provide a language for expressing geographic entities – to create application specific geographic vocabularies.
- To enable the encoding of geographic information consistent with these vocabularies.
- To support geospatial queries and transactions across the Internet.

GML is feature-centric. Features are entities – things that describe aspects of the real world from the perspective of a particular application community – whether circumscribed by geography or function or both. GML vocabularies are created by communities of interest. These vocabularies are called GML Application Schemas. If you look at such an Application Schema you will find real world objects like Buildings, Roads, Buoys, Navigation Aids, Airline Flight Paths, Vehicles and Railway Switches. Each such object is defined in the schema by listing its properties. For example, a Building might be described by:

```
<abc:Building gml:id='b143'>
      <abc:height>40</abc:height>
      <abc:footprint>
            <gml:polygon></gml:polygon>
      </abc:footprint>
</abc:Building>
```

Note that the Building (feature) has two properties, namely height and footprint. The height property in this case has an integer value (number of stories), while the footprint property has a Polygon (shape) for a value.

GML application schemas can be the basis of standards in themselves – such as S57GML, cityGML, geoRSS GML and AIXM, or they can be informal creations for only a very small community. Which is the case is up to the community.

GML application schemas should NOT be confused with GML profiles. A GML profile is a subset of GML, defined usually by the subset tool (part of the GML specification), consisting of selected element, attribute and type declarations and all dependent components from the GML core schemas (the schemas defined by the GML specification). Application schemas can be built on GML profiles. Some GML profiles are also specifications and this includes the GML Simple Features Profile, the Point Profile, the GML Profile for GMLJP2 and the GML Profile for GeoRSS.

GML was developed to support geographic requests and transactions and this usage predates the WFS developed for this purpose. When you send a request for geographic data – e.g. "find all water wells within this county" – you need a means to express water well and county and the geometric extend of the county. In WFS you use GML for this purpose. When you want to send a transaction such as "change the shape of the Holmes River to the following …" you need a way to express the river's geometry and GML provides this mechanism in the WFS.

3.2 Overview of GML Schema

GML specifies XML encodings of a number of the conceptual classes defined in the ISO 19100 series of International Standards and the OpenGIS Abstract Specification in conformance with these standards and specifications.

In many cases, the mapping from the conceptual classes to XML is straightforward, while in some cases the mapping is more complex.

In addition, GML provides XML encodings for additional concepts not yet modeled in the ISO 19100 series of International Standards or the OpenGIS Abstract Specification. Examples include moving objects, simple observations or value objects. Additional conceptual classes corresponding to these extensions are also specified in Annex D.

The GML schema comprises the components (XML elements, attributes, simple types, complex types, attribute groups, groups, etc.) that are described in this International Standard. The XML encoding conforms to ISO 19118.

3.3 GML Schema Features

A GML feature is a feature encoded using GML. For example, a road, a river, a person, a vehicle, an administrative area, an event, etc.

The feature schema provides a framework for the creation of GML features and feature collections.

3.3.1 ABSTRACTFEATURETYPE

The basic feature model is given by the `gml:AbstractFeatureType`, defined in the schema as follows:

```
<complexType name="AbstractFeatureType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:boundedBy" minOccurs="0"/>
        <element ref="gml:location" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

The content model for `gml:AbstractFeatureType` adds two specific properties suitable for geographic features to the content model defined in `gml:AbstractGMLType`.

The value of the `gml:boundedBy` property describes an envelope that encloses the entire feature instance, and is primarily useful for supporting rapid searching for features that occur in a particular location.

The value of the `gml:location` property describes the extent, position or relative location of the feature. `gml:location` is deprecated as part of the standard content model of `gml:AbstractFeatureType`.

3.3.2 ABSTRACTFEATURE

The element `gml:AbstractFeature` is declared as follows:

```
<element name="AbstractFeature"
           type="gml:AbstractFeatureType"
           abstract="true"
           substitutionGroup="gml:AbstractGML"/>
```

This abstract element serves as the head of a substitution group which may contain any elements whose content model is derived from `gml:AbstractFeatureType`. This may be used as a variable in the construction of content models.

`gml:AbstractFeature` may be thought of as anything that is a GML featurell and may be used to define variables or templates in which the value of a GML property is “any feature”. This occurs in particular in a GML feature collection where the feature member properties contain one or multiple copies of `gml:AbstractFeature` respectively.

The Other features which are used are boundedBy, BoundingShapeType, EnvelopeWithTimePeriod, EnvelopeWithTimePeriodType, locationName, locationReference,FeaturePropertyType, FeatureArrayPropertyType.

3.4 Overview

Many applications require definitions of terms which are used within instance documents as the values of certain properties or as reference information to tie properties to standard information values in some way. Units of measure and descriptions of measurable phenomena are two particular examples.

It will often be convenient to use definitions provided by external authorities. These may already be packaged for delivery in various ways, both online and offline. In order that they may be referred to from GML documents it is generally necessary that a URI be available for each definition. Where this is the case then it is usually preferable to refer to these directly.

Alternatively, it may be convenient or necessary to capture definitions in XML, either embedded within an instance document containing features or as a separate document. The definitions may be transcriptions from an external source, or may be new definitions for a local purpose. In order to support this case, some simple components are provided in GML in the form of:

- A generic `gml:Definition`, which may serve as the basis for more specialized definitions.
- A generic `gml:Dictionary`, which allows a set of definitions or references to definitions to be collected.

These components may be used directly, but also serve as the basis for more specialized definition elements in GML, in particular: coordinate operations (Clause 12), coordinate reference systems (Clause 12), datums (Clause 12), temporal reference systems (Clause 14), and units of measure (Clause 16).

Note that the GML definition and dictionary components implement a simple nested hierarchy of definitions with identifiers. The latter provide handles which may be used in the description of more complex relationships between terms. However, the GML dictionary components are not intended to provide direct support for complex taxonomies, ontologies or thesauri. Specialized XML tools are available to satisfy the more sophisticated requirements.

➔ The dictionary schema document is identified by the following location-independent name (using URN syntax): urn:x-ogc:specification:gml:schema-xsd:dictionary:3.2.1.

3.4.1 DICTIONARY SCHEMA

Definition, DefinitionType, remarks

The basic `gml:Definition` element specifies a definition, which can be included in or referenced by a dictionary. It is declared as follows:

```
<element name="Definition" type="gml:DefinitionType"
substitutionGroup="gml:AbstractGML"/>
<complexType name="DefinitionBaseType">
  <complexContent>
    <restriction base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:metaDataProperty"
                 minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:description" minOccurs="0"/>
        <element ref="gml:descriptionReference" minOccurs="0"/>
        <element ref="gml:identifier"/>
        <element ref="gml:name"
                 minOccurs="0" maxOccurs="unbounded"/>
       </sequence>
         <attribute ref="gml:id" use="required"/>
      </restriction>
   </complexContent>
</complexType>
<complexType name="DefinitionType">
   <complexContent>
     <extension base="gml:DefinitionBaseType">
       <sequence>
         <element ref="gml:remarks" minOccurs="0"/>
       </sequence>
     </extension>
   </complexContent>
</complexType>
<element name="remarks" type="string"/>
```

The content model for a generic definition is a derivation from `gml:AbstractGMLType`.

The `gml:description` property element shall hold the definition if this can be captured in a simple text string, or the `gml:descriptionReference` property element may carry a link to a description elsewhere.

The `gml:identifier` element shall provide one identifier identifying this definition. The identifier shall be unique within the dictionaries using this definition.

The `gml:name` elements shall provide zero or more terms and synonyms for which this is the definition.

The `gml:remarks` element shall be used to hold additional textual information that is not conceptually part of the definition but is useful in understanding the definition.

Dictionary, DictionaryType

Sets of definitions may be collected into dictionaries or collections. These are declared in the schema as follows:

```
<element name="Dictionary" type="gml:DictionaryType"
substitutionGroup="gml:Definition"/>
<complexType name="DictionaryType">
  <complexContent>
    <extension base="gml:DefinitionType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="gml:dictionaryEntry"/>
        <element ref="gml:indirectEntry"/>
      </choice>
    <attributeGroup ref="gml:AggregationAttributeGroup"/>
  </extension>
 </complexContent>
</complexType>
```

A `gml:Dictionary` is a non-abstract collection of definitions.

The `gml:Dictionary` content model adds a list of `gml:dictionaryEntry` and *`gml:indirectEntry`* *(deprecated)* properties that contain *or reference* `gml:Definition` objects. A database handle (`gml:id` attribute) is required, in order that this collection may be referred to. The standard `gml:identifier`, `gml:description`, `gml:descriptionReference` and `gml:name` properties are available to reference or contain more information about this dictionary. The `gml:description` and `gml:descriptionReference` property elements may be used for a description of this dictionary. The derived `gml:name` element may be used for the name(s) of this dictionary.

dictionaryEntry, DictionaryEntryType

These elements contain or refer to the definitions which are members of a dictionary. The element `gml:dictionaryEntry` is declared as follows:

```
<element name="dictionaryEntry" type="gml:DictionaryEntryType"/>
<complexType name="DictionaryEntryType">
  <complexContent>
    <extension base="gml:AbstractMemberType">
      <sequence minOccurs="0">
        <element ref="gml:Definition"/>
      </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </complexContent>
</complexType>
```

The content model follows the standard GML property pattern, so a `gml:dictionaryEntry` may either contain or refer to a single `gml:Definition`. Since `gml:Dictionary` is substitutable for `gml:Definition`, the content of an entry may itself be a lower-level dictionary.

Note that if the value is provided by reference, this definition does *not* carry a handle (`gml:id`) in this context, so does *not* allow external references to this specific definition in this context. When used in this way the referenced definition will usually be in a dictionary in the same XML document.

Using definitions and dictionaries

Dictionaries and definitions are GML objects, so may be found in independent GML data instance documents.

In application schemas it might be useful to attach a `gml:Dictionary` or `gml:Definitions` to a feature collection in order to record definitions used in properties of members of the collection.

The following example shows two instances of dictionaries:

```
<gml:Dictionary gml:id="rockTypes">
    <gml:description>
        A simple dictionary of rock types using components from gmlBase
      </gml:description>
      <gml:identifier codeSpace="http://www.abc.org/terms">
        Rock Types
      </gml:identifier>
      <gml:dictionaryEntry>
         <gml:Definition gml:id="granite">
             <gml:description>
                 A igneous rock normally composed of quartz, two feldspars and
optional mica
             </gml:description>
             <gml:identifier codeSpace="http://www.abc.org/terms">
                 Granite
             </gml:identifier>
             </gml:Definition>
     </gml:dictionaryEntry>
     <gml:dictionaryEntry>
             <gml:Definition gml:id="sst">
             <gml:description>
                  A detrital sedimentary rock normally composed of siliceous grains
             </gml:description>
                  <gml:identifier codeSpace="http://www.abc.org/terms">
                  Sandstone
                  </gml:identifier>
             </gml:Definition>
      </gml:dictionaryEntry>
      <gml:dictionaryEntry
xlink:href="http://my.big.org/definitions/geology/limestone"/>
</gml:Dictionary>
<gml:Dictionary gml:id="AbridgedGMLdictionary">
<gml:identifier codeSpace="http://www.opengis.net/gml/3.2">
GML Dictionary
</gml:identifier>
<gml:dictionaryEntry>
      <gml:Definition gml:id="term4.1">
            <gml:description>
conceptual schema for data required by one or more applications
</gml:description>
            <gml:identifier codeSpace="http://www.isotc211.org/19101">
application schema
</gml:identifier>
      </gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
<gml:Definition gml:id="term4.2">
            <gml:description>
application schema written in XML Schema in accordance with the rules specified in ISO
19136
</gml:description>

Continued…
```

```
…Continued
<gml:identifier codeSpace="http://www.opengis.net/gml/3.2">
GML application schema
</gml:identifier>
          </gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
      <gml:Definition gml:id="term4.3">
            <gml:description>
semantic relationship between two or more classifiers that specifies connections among
their instances
 </gml:description>
            <gml:identifer
codeSpace="http://www.uml.org/1.3">association</gml:identifier>
      </gml:Definition>
</gml:dictionaryEntry>
<gml:dictionaryEntry>
      <gml:Definition gml:id="term4.4">
            <gml:description>name-value pair contained in an element</gml:description>
            <gml:identifer
codeSpace="http://www.w3.org/XML/1998/namespace">attribute</gml:identifier>
      </gml:Definition>
</gml:dictionaryEntry>
<!--.. -->
</gml:Dictionary>
```

3.5 GML-Based Request

The request to the WFS for any of the operations can be sent as a GML Based request instead of the plain Get URL. The Key and value pair of parameters that we send to the WFS server is now sent through the GML-based request in the following way:

```
<?xml version="1.0" encoding="utf-8"?>
<GetFeature
xmlns="http://www.opengis.net/wfs"xmlns:DigitalGlobe=http://www.digitalglobe.com
xmlns:ogc="http://www.opengis.net/ogc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:gml="http://www.opengis.net/gml" service="WFS" version="1.1.0"
outputFormat="text/xml; subtype=gml/3.1.1" maxFeatures="100" handle="" >
      <Query typeName="DigitalGlobe:FinishedFeature" srsName="urn:x-
ogc:def:crs:EPSG:4326">
         <ogc:Filter>
             <ogc:Intersects>
                <ogc:PropertyName>geometry</ogc:PropertyName>
                <gml:Envelope srsName="urn:x-ogc:def:crs:EPSG:4326"

xmlns:gml="http://www.opengis.net/gml">
                   <gml:lowerCorner>-90 -180</gml:lowerCorner>
               <gml:upperCorner>90 180</gml:upperCorner>
             </gml:Envelope>
          </ogc:Intersects>
       </ogc:Filter>
     </Query>
</GetFeature>
```

The <GetFeature> element contains one or more <Query> elements, each of which in turn contains the description of a query. The results of all queries contained in a GetFeature request are concatenated to produce the result set.

The outputFormat attribute defines the format to use to generate the result set. The default value is GML2.

The optional maxFeatures attribute can be used to limit the number of features that a GetFeature request retrieves. Once the maxFeatures limit is reached, the result set is truncated at that point.

Each individual query packaged in a GetFeature request is defined using the <Query> element. The <Query> element defines which feature type to query, what properties to retrieve and what constraints (spatial and non-spatial) to apply to those properties.

The typeName attribute is used to indicate the name of the feature type or class to be queried.

The featureVersion attribute is included in order to accommodate systems that support feature versioning. A value of ALL indicates that all versions of a feature should be fetched. Otherwise, an integer, n, can be specified to return the nth version of a feature. The version numbers start at 1, which is the oldest version. If a version value larger than the largest version number is specified, then the latest version is returned. The default action shall be for the query to return the latest version. Systems that do not support versioning can ignore the parameter and return the only version that they have.

The <PropertyName> element is used to enumerate the feature properties that should be selected during a query and whose values should be included in the response to a GetFeature request. A client application can determine the properties of a feature by making a DescribeFeatureType request before composing a GetFeature request. The DescribeFeatureType operation [sec. 8] will generate a GML application schema defining the schema of the feature type. The client can then select the properties to be fetched. In addition, the client can determine which feature properties are mandatory and must be fetched in order for the WFS to be able to generate an instance of the feature type that will validate againt the generated GML application schema. In the event that a WFS encounters a query that does not select all mandatory properties of a feature, the WFS will internally augment the property name list to include all necessary property names. A WFS client must thus be prepared to deal with a situation where it receives more property values than it requests.

If no <PropertyName> elements are specified, then all feature properties should be fetched.

The <Filter> element can be used to define constraints on a query. Both spatial and/or non-spatial constraints can be specified as described in the Filter Encoding Specification [3]. If no <Filter> element is contained within the <Query> element, then the query is unconstrained and all feature instances should be retrieved.

4 Web Feature Service

4.1 Introduction

The DigitalGlobe Web Feature Service defines a set of functions that clients may use to return actual features with geometry and attributes that can be used in any type of geospatial analysis.

Any client making requests that conform to the OGC WFS specification can interact with DGCS WFS server. Web-based client/server architecture is a typical example of the structure of a web feature service application, as illustrated in Figure 4.1 below:

FIGURE 4.1 A TYPICAL STRUCTURE OF A DGCS-WFS APPLICATION

In DGCS - WFS scenario, the client application requests desired information from the web feature service server. The WFS server retrieves from the feature store the appropriate information including vector metadata, imagery footprints and responds to the request with this information in a GML format.

4.2 WFS Client-Server Architecture

The architecture shown in Figure 4.2 depicts a sample integration of WFS client and server applications. Client Viewer is a series of HTML pages running inside a web browser that can interact with WFS server via client application through HTTPS calls. WFS client manages the interactions with WFS interfaces through HTTPS requests and dynamically generates HTML that can run in a Web browser.

WFS server accepts requests from WFS client and viewer client in the form of HTTPS URL strings, and returns results encoded as XML, GIF, GML, and so on. The database stores geo-feature data that can be accessed and utilized by the WFS server to generate GML documents or draw maps.

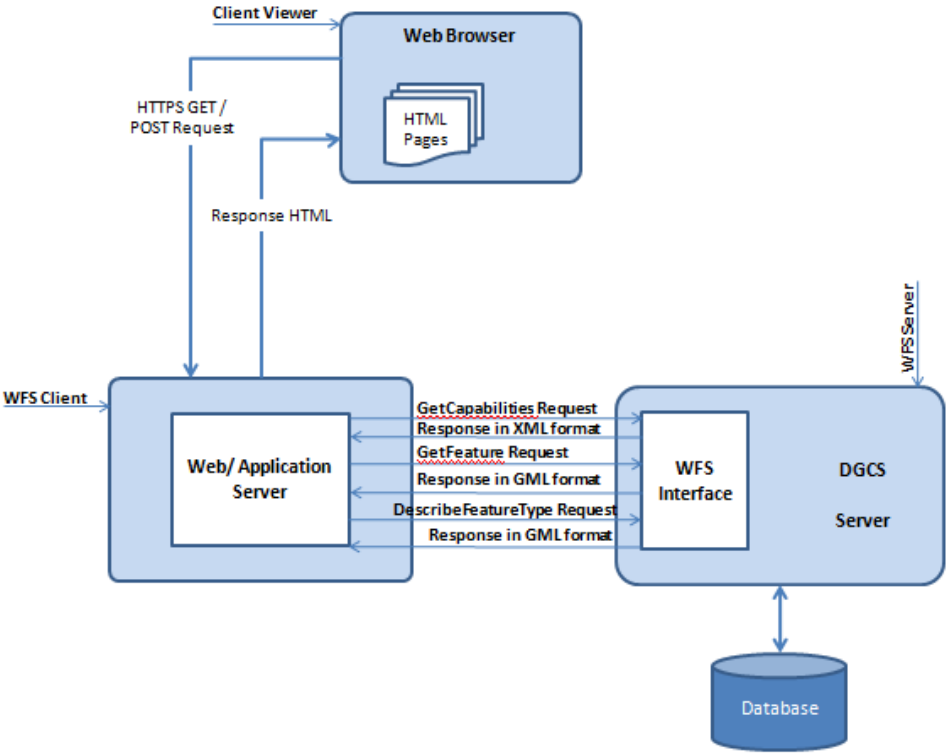

FIGURE 4.2 SAMPLE WFS CLIENT-SERVER APPLICATION

4.3 WFS Service Details

The DigitalGlobe Web Feature Service provides vector metadata, including imagery footprints, in GML format. The DigitalGlobe WFS supports the following OGC-defined operations.

GetCapabilities

The GetCapabilities request is used to determine the supported FeatureTypes.

DescribeFeatureType

The DescribeFeatureType request is used to discover the properties available for a supported FeatureType.

GetFeature

The GetFeature request is used to get the properties of one or more instances of a supported FeatureType.

The DigitalGlobe WFS service categorizes features as one of the following types:

- FinishedFeature. This feature type provides users with access to all finished product data via a single feature.
- Tile Matrix Feature. This feature type provides users with access to all of the available metadata, for all of the feature instances that occur in a particular tile.
- ImageInMosaicFeature. This feature type provides users with access to the seam line information that is available with mosaicked finished products.

4.4 Basic Service Elements

This section specifies aspects of Web Feature Server behavior that are independent of particular operations or are common to several operations.

4.4.1 HTTP REQUEST

HTTP functions as a request-response protocol in the client-server computing model. In HTTP, a web browser, for example, acts as a client, while an application running on a computer hosting a web site functions as a server. The client submits an HTTP request message to the server. The server, which stores content, or provides resources, such as HTML files and images, or generates such content as required, or performs other functions on behalf of the client, returns a response message to the client. A response contains completion status information about the request and may contain any content requested by the client in its message body.

An HTTP Uniform Resource Locator (URL) locates the Online Resource of each operation supported by a service instance. The URL may be different for each operation, or the same, at the discretion of the service provider.

HTTP supports two request methods: GET and POST. One or both of these methods may be defined for a particular web service and offered by a service instance. The use of the Online Resource URL differs in each case.

HTTP GET

An Online Resource URL intended for HTTP GET requests, is, in fact, only a URL prefix to which additional parameters must be appended in order to construct a valid Operation request. A URL prefix is defined as an opaque string including the protocol, hostname, optional port number, path, a question mark '?', and, optionally, one or more server-specific parameters ending in an ampersand '&'. The prefix uniquely identifies the particular service instance.

A client can append the necessary request parameters as name/value pairs in the form "name=value&". The resulting URL must be valid according to the HTTP Common Gateway Interface (CGI) standard, which mandates the presence of '?' before the sequence of query parameters and the '&' between each parameter.

Table 4.1 summarizes the components of an operation request URL.

The URL prefix must end in either a '**?**' (in the absence of additional server-specific parameters) or a '**&**'**.** In practice, however, Clients **should** be prepared to add a necessary trailing '**?**' or '**&**' before appending the operation parameters defined as per DG-WFS specification in order to construct a valid request URL. Refer to Table 4.2 for reserved characters as per HTTP rules.

TABLE 4.1 A GENERAL GET REQUEST

URL COMPONENT	DESCRIPTION
http://host[:port]/path?{name[=value]&}	URL prefix of service operation. [ ] denotes 0 or 1 occurrence of an optional part; {} denotes 0 or more occurrences. The prefix is entirely at the discretion of the service provider.
name=value&	One or more standard request parameter name/value pairs defined by a Web Feature Service. The actual list of required and optional parameters is mandated for each operation is described later in the document.

TABLE 4.2 RESERVED CHARACTERS IN HTTP GET QUERY

CHARACTER	RESERVED USAGE
?	Separator indicating start of query string.
&	Separator between parameters in query string.
=	Separator between name and value of parameter
/	Separator between MIME type and subtype in format parameter value.
:	Separator between Namespace and Identifier in SRS parameter value.
,	Separator between individual values in list-oriented parameters.

HTTP POST

An Online Resource URL intended for HTTP POST requests is a complete and valid URL to which clients transmit encoded requests in the *body* of the POST document. DGCS-WFS do not require additional parameters to be appended to the URL in order to construct a valid target for the Operation request. Refer to Figure 4.3 to see a sample of a HTTP Post request.

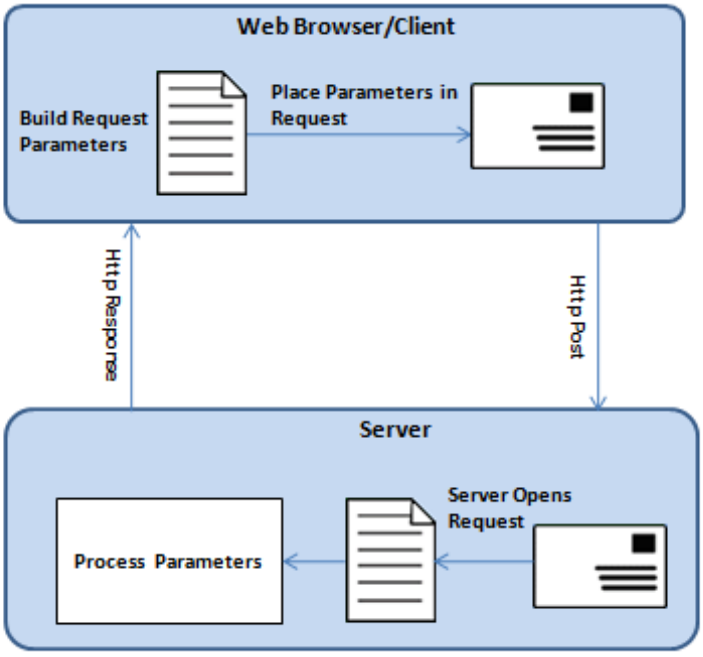

FIGURE 4.3 SAMPLE WFS CLIENT-SERVER APPLICATION

Advantages of HTTP Post instead of HTTP Get:

- The Parameter's name and value are visible to the user and to anyone who is looking at the URL in the browser.
- GET requests are passed as the URL string and are therefore limited by the URL length limit specified by the browser.
- HTTP Post method can upload files to the server.

HTTPS

DigitalGlobe offers Web Feature Service using HTTPS. HTTPS is HTTP over a secure communication channel which allows encrypted information to be transferred between machines over the World Wide Web.

The use of HTTPS does not affect the description of the requests and responses described in this document but may require additional actions to be taken on both the client and the service in order to initiate secure communication.

4.4.2 HTTP RESPONSE

Upon receiving a valid HTTP request, the service sends a response corresponding exactly to the request as detailed based on parameters for the specific operations.

Response objects will be accompanied by other HTTP entity headers as appropriate and to the extent possible. In particular, the Expires and Last-Modified headers provide important information for caching; Content-Length may be used by clients to know when data transmission is complete and to efficiently allocate space for results, and Content-Encoding or Content-Transfer-Encoding may be necessary for proper interpretation of the results. If the request is invalid, the service will issue a Service Exception which is explained in detail in Section 0.

Output Formats

The optional **outputFormat** attribute specifies the format of the response to a web service request. The default value is text/xml; **subtype=gml/3.1.1** indicating that a valid GML3 document, that validates against a valid GML3 application schema, must be generated. For backward compatibility, the values **GML2** or **text/xml; subtype=gml/2.1.2** may be specified indicating that a valid GML2 document that validates against a valid GML2 application schema, must be generated. The following table summarizes the possible values for the **outputFormat** attribute:

TABLE 4.3 VALUES FOR OUTPUT FORMAT ATTRIBUTE

OUTPUTFORMAT VALUE	DESCRIPTION
GML2	This value is kept for backward compatibility and indicates that an XML instance document must be generated that validates against a GML2 application schema.
text/xml; subtype=gml/2.1.2	Same as GML2.
text/xml; subtype=gml/3.1.1	This value indicates that an XML instance document must be generated that validates against a GML3 application schema. This is the default values of the **outputFormat** attribute if the attribute is not specified in the GetFeature request.

4.4.3 REQUEST PARAMETERS

As per the specification standards of WFS, a client application has to form the HTTPS-based URL dynamically, based on requirement or operation it has to perform. The following are the list of important parameters which are part of WFS URL.

Base URL

For every request to DigitalGlobe WFS server, client needs to append parameters to the base URL. DigitalGlobe provides the base URL for WFS, which is used as the common base URL.

Base URL:

https://services.digitalglobe.com/catalogservice/wfsaccess

Username and Password are required only for some accounts. All others require a ConnectID.

CONNECTID

ConnectID is a parameter name which needs to be appended along with base URL mentioned above with appropriate value. Value for these parameters is a unique 32 digit alphanumeric value. It is a mandatory parameter which should be part of every request client makes with server. Please contact DigitalGlobe to get your unique ConnectID.

ConnectID format: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

x → alpha numeric number

SERVICE

This parameter defines the type of service the client is requesting. As mentioned above, DigitalGlobe provides different services like WMS, WFS, WMTS and WCS and the client need to provide an appropriate value based on the service to be requested. The value for this parameter always is "**WFS**" for WFS clients.

Example: service=WFS

VERSION

The "**version**" parameter specifies the protocol version number. The version number indicates the specification defined by OGC. The format of version number contains three positive integers, separated by decimal points, in the form "**x.y.z"**. The numbers "y" and "z" will never exceed 99. Each Feature Service provided by DigitalGlobe numbered independently as per respective OGC specification standards. The latest version of DigitalGlobe WFS implemented for the OGC specification is 1.1.0.

The version number appears in following two places:

- In the response XML of GetCapabilities request describing WFS service
- In the parameter list of client requests to the WFS service.

In response to the GetCapabilities request containing a version number, WFS server responds with output that conforms to that version of the specification, or negotiates a mutually agreeable version if the requested version is not implemented on the server. If no version number is specified in the request, the server responds with the

highest version it understands and label the response accordingly. Please refer to the OGC specification for WFS on negotiation rules.

Example: version=1.1.0 (Recommended until DigitalGlobe implements new version per OGC specification)

REQUEST

The REQUEST parameter indicates which service operation is being invoked. The value shall be the name of one of the operations offered by DigitalGlobe Web Feature Service. Please refer to Section 4.2 for different operations supported by DigitalGlobe WFS, along with the descriptions.

Example: request=GetCapabilities

EXCEPTIONS

The EXCEPTIONS parameter in a request indicates the format in which the Client wishes to be notified of Service Exceptions. The only value of the EXCEPTIONS parameter that is defined for WFS Web Service is "application/vnd.ogc.se_xml", which means "Service Exception XML". Individual error messages appear as <ServiceException> elements within the <ServiceExceptionReport> in Service Exception XML. Please refer to Section 0 for more details on Service Exceptions.

BBOX (Bounding Box)

The Bounding Box (BBOX) is a set of four comma-separated decimal, scientific notation or integer values represents the geo referenced bounding parameters of Area Of Interest (AOI). These values specify the minimum X, minimum Y, maximum X, and maximum Y ranges, in that order, expressed in units of the Spatial Reference System (SRS) of the request, such that a rectangular area is defined in those units.

The four bounding box values indicate the outside edges of a rectangle: minimum X is the left edge, maximum X the right, minimum Y the bottom, and maximum Y the top (see Figure 4.4). The relation of the Bounding Box to the image pixel matrix is shown: the bounding box goes around the "outside" of the pixels of the image rather than through the centers of the border pixels. In this context, individual pixels have an area.

Rules to follow while defining BBOX:

- A Bounding Box should not have zero area.
- Minimum X should be less than or equal to the Maximum X and Minimum Y should be less than or equal to the Maximum Y.

Input and output BBOX SRS values generate different BBOX order output.

Specifying the Output BBOX SRS

To specify the output BBOX SRS, add it as a separate parameter (e.g. &srsName=EPSG:4326). In this method, the order of BBOX is minY, minX, maxY, maxX (or minlat, minlong, maxlat, maxlong). See the sample URL below for context. Refer to Figure 4.4 for the resulting BBOX.

Sample URL:
https://services.digitalglobe.com/catalogservice/wfsaccess?SERVICE=WFS&REQUEST=GetFeature%20&typeName=DigitalGlobe:FinishedFeature&VERSION=1.1.0&CONNECTID=<CONNECTID>&BBOX=30.250383448012,-97.7684708569195,30.3119534076606,-97.7092977813448&WIDTH=3000&HEIGHT=3000&**srsName=EPSG:4326**

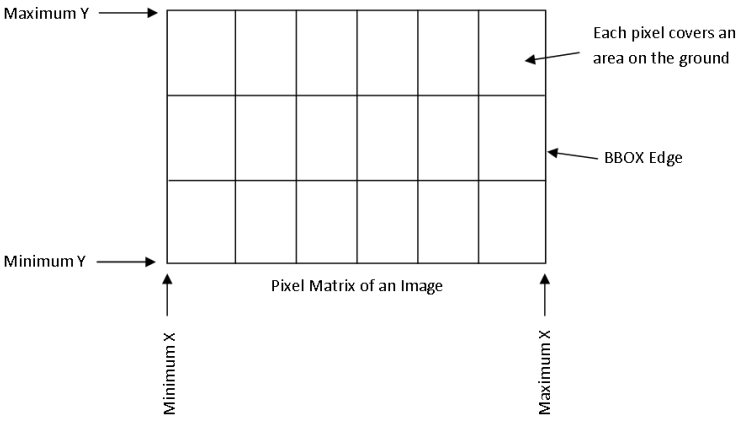

FIGURE 4.4 BBOX WITH OUTPUT SRS SPECIFIED

Specifying the Input BBOX SRS

To specify the output BBOX SRS, include it as part of the BBOX (e.g. &BBOX=39.346038,-105.424805,40.131711,-104.545898,EPSG:4326). With this method, the order of BBOX is minX, minY, maxX, maxY (or minlong, minlat, maxlong, maxlat). See the sample URL below for context. Refer to Figure 4.5 for the resulting BBOX.

Sample URL:
https://services.digitalglobe.com/catalogservice/wfsaccess?SERVICE=WFS&REQUEST=GetFeature%20&typeName=DigitalGlobe:FinishedFeature&VERSION=1.1.0&CONNECTID=<CONNECTID>&**BBOX**=-97.7684708569195,30.250383448012,-97.7092977813448,30.3119534076606,**EPSG:4326**&WIDTH=3000&HEIGHT=3000

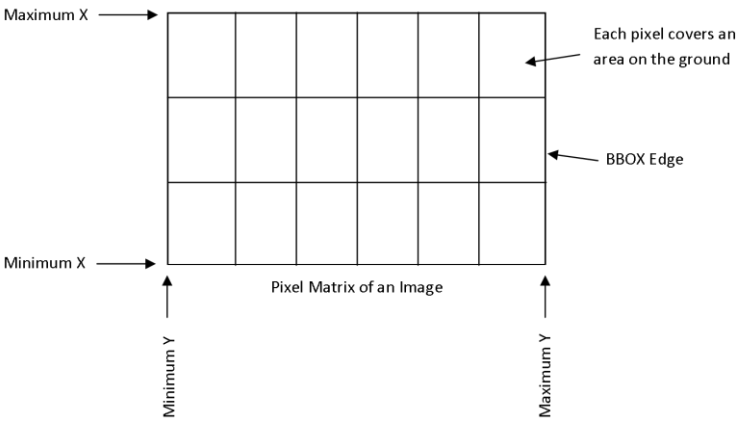

FIGURE 4.5 BBOX WITH INPUT SRS SPECIFIED

Not Specifying an SRS

If no SRS is specified, the system will use the default value of EPSG:4326.

TYPE NAME

If additional metadata about either all features in a particular tile, or a single feature included in a particular tile is needed, it can be retrieved by the *typeName* from the *getFeature* Request.

4.4.4 REQUEST PARAMETER RULES

While forming request URL, client applications should follow certain rules as described below:

- Parameter names are not case-sensitive, but parameter values are case-sensitive. Parameter names are typically shown in uppercase for typographical clarity, not as a requirement.
- Parameters in a request may be specified in any order.
- When request parameters are duplicated with conflicting values, the response from the server may be undefined.
- Parameters consisting of lists (for example, BBOX, LAYERS and STYLES in WFS requests) shall use the comma (“,”) as the separator between items in the list. Additional white space shall not be used to delimit list items.
- Two successive commas indicate an empty item, as does a leading comma or a trailing comma. An empty list (“”) shall be interpreted either as a list containing no items or as a list containing a single empty item, depending on context.

4.5 Integration Procedure

A WFS client application is a program that communicates with the DGCS WFS server using the three functions GetCapabilities, GetFeature and DescribeFeatureType, as noted earlier. More specifically, in a typical WFS client/server interaction, the following steps can be followed:

STEP-1

The client first has to request **GetCapabilities** from the WFS server in order to determine what the WFS server can do and what features the WFS server can provide.

Example Request:
https://services.digitalglobe.com/catalogservice/wfsaccess?SERVICE=WFS&REQUEST=GetCa

pabilities&VERSION=1.1.0&connectid=<CONNECTID>&username=<username>&password=<password>

Username and Password parameters may not be applicable depending on your account type. Replace <CONNECTID> with the ConnectID provided by DigitalGlobe. Parameters do not have to be entered in the order shown.

Understanding URL

The previous URL contains Base URL and few parameters as explained in Section 4.4. The key parameter for this request is "**request=GetCapabilities**" which fetches the capabilities of Web Feature Service and response back in the form of XML data.

Response

In response to a GetCapabilities request, the DGCS WFS server produces an eXtensible Markup Language (XML) document containing the WFS server's service metadata, describing all the operations it supports, and providing information about the available features. The client application has to parse the XML capabilities document to retrieve the necessary information used to request for a feature. The Document Object Model (DOM) is a widely used and efficient XML parser, which can be utilized to parse the XML document and retrieve the information. The DOM represents an XML document as a tree of nodes that can be easily traversed and edited with its standard interfaces.

Response XML to GetCapabilities request contains the following details:

- WFS Service details like Name, Title, URL.
- Contact Information Person, Organization, Address, Telephone, Fax and Email.
- WFS Capabilities like GetCapabilties, GetFeature and DescribeFeatureInfo along with respective formats and URLs.

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:WFS_Capabilities xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opengis.net/wfs" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:ows="http://www.opengis.net/ows" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:DigitalGlobe="http://www.digitalglobe.com" version="1.1.0"
xsi:schemaLocation="http://www.opengis.net/wfs
http://services.digitalglobe.com:80/wfsservice/schemas/wfs/1.1.0/wfs.xsd"
updateSequence="103">
      <ows:ServiceIdentification>
            <ows:Title>DigitalGlobe Web Feature Service</ows:Title>
<ows:Abstract/>
<ows:Keywords>
            <ows:Keyword>WFS</ows:Keyword>
            <ows:Keyword>WFS</ows:Keyword>
            </ows:Keywords>
            <ows:ServiceType>WFS</ows:ServiceType>
            <ows:ServiceTypeVersion>1.1.0</ows:ServiceTypeVersion>
<ows:Fees>NONE</ows:Fees>
            <ows:AccessConstraints>NONE</ows:AccessConstraints>
      </ows:ServiceIdentification>
 <ows:ServiceProvider>
            <ows:ProviderName>DigitalGlobe Inc</ows:ProviderName>
            <ows:ServiceContact>
<ows:IndividualName>Customer Service Department</ows:IndividualName>
<ows:PositionName>Customer Service Department</ows:PositionName>
               <ows:ContactInfo>
                  <ows:Phone>
                     <ows:Voice>800.496.1225</ows:Voice>
                     <ows:Facsimile>303.684.4562</ows:Facsimile>
                        </ows:Phone>
                        <ows:Address>
                          <ows:City/>
                          <ows:AdministrativeArea/>
                          <ows:PostalCode/>
                          <ows:Country/>
                        </ows:Address>
                </ows:ContactInfo>
            </ows:ServiceContact>
      </ows:ServiceProvider>
<ows:OperationsMetadata>
      <ows:Operation name="GetCapabilities">
            <ows:DCP>
                  <ows:HTTP>
                        <ows:Get
xlink:href="https://services.digitalglobe.com/catalogservice/wfsaccess?CONNECTID=<
CONNECTID>"/>
  <ows:Post xlink:href=
"https://services.digitalglobe.com/catalogservice/wfsaccess?CONNECTID=<CONNECTID>"
/>
                  </ows:HTTP>
            </ows:DCP>
<ows:Parameter name="AcceptVersions">
                  <ows:Value>1.0.0</ows:Value>
                  <ows:Value>1.1.0</ows:Value>
            </ows:Parameter>
            <ows:Parameter name="AcceptFormats">
                  <ows:Value>text/xml</ows:Value>
</ows:Parameter>
      </ows:Operation>
      <ows:Operation name="DescribeFeatureType">
            <ows:DCP>
            <ows:HTTP>
Continued…
```

```
…Continued
                        <ows:Get xlink:href=
"https://services.digitalglobe.com/catalogservice/wfsaccess?CONNECTID=<CONNECTID>"/>
                        <ows:Post xlink:href=
"https://services.digitalglobe.com/catalogservice/wfsaccess?CONNECTID=<CONNECTID>"/>
</ows:HTTP>
            </ows:DCP>
name="outputFormat">
                  <ows:Value>text/xml; subtype=gml/3.1.1</ows:Value>
            </ows:Parameter>
      </ows:Operation>
      <ows:Operation name="GetFeature">
            <ows:DCP>
                  <ows:HTTP>
                  <ows:Get xlink:href=
"https://services.digitalglobe.com/catalogservice/wfsaccess?CONNECTID=
<CONNECTID>"/>
                  <ows:Post xlink:href=
"https://services.digitalglobe.com/catalogservice/wfsaccess?CONNECTID=<CONNECTID>"/>
                  </ows:HTTP>
            </ows:DCP>
            <ows:Parameter name="resultType">
<ows:Value>results</ows:Value>
            <ows:Value>hits</ows:Value>
<ows:Parameter
</ows:Parameter>
<ows:Parameter name="outputFormat">
          <ows:Value>text/xml; subtype=gml/3.1.1</ows:Value>
          <ows:Value>text/xml; subtype=gml/2.1.2</ows:Value>
</ows:Parameter>
<ows:Constraint name="LocalTraverseXLinkScope">
<ows:Value>2</ows:Value>
</ows:Constraint>
</ows:Operation>
</ows:OperationsMetadata>
<FeatureTypeList>
<Operations>
<Operation>Query</Operation>
          </Operations>
          <FeatureType xmlns:DigitalGlobe="http://www.digitalglobe.com">
              <Name>DigitalGlobe:FinishedFeature</Name>
              <Title>FinishedFeature</Title>
<Abstract>Generated from CatalogService</Abstract>
           <ows:Keywords>
<ows:Keyword>CatalogService</ows:Keyword>
                  <ows:Keyword>FinishedFeature</ows:Keyword>
           </ows:Keywords>
<DefaultSRS>urn:x-ogc:def:crs:EPSG:4326</DefaultSRS>
<ows:WGS84BoundingBox>
                  <ows:LowerCorner>-180.0 -90.0</ows:LowerCorner>
                  <ows:UpperCorner>180.0 90.0</ows:UpperCorner>
                </ows:WGS84BoundingBox>
          </FeatureType>
</FeatureTypeList>
<ogc:Filter_Capabilities>
<ogc:Spatial_Capabilities>
          <ogc:GeometryOperands>
<ogc:GeometryOperand>gml:Polygon</ogc:GeometryOperand>
          </ogc:GeometryOperands>

Continued…
```

```
…Continued
<ogc:SpatialOperators>
<ogc:SpatialOperator name="Intersects"/>
</ogc:SpatialOperators>
</ogc:Spatial_Capabilities>
<ogc:Scalar_Capabilities>
          <ogc:LogicalOperators/>
          <ogc:ComparisonOperators>
<ogc:ComparisonOperator>EqualTo</ogc:ComparisonOperator>
<ogc:ComparisonOperator>Between</ogc:ComparisonOperator>
          </ogc:ComparisonOperators>
<ogc:ArithmeticOperators>
      <ogc:SimpleArithmetic/>
<ogc:Functions>
<ogc:FunctionNames>
                  <ogc:FunctionName nArgs="2">
                        stackingViewport
                  </ogc:FunctionName>
            </ogc:FunctionNames>
      </ogc:Functions>
</ogc:ArithmeticOperators>
</ogc:Scalar_Capabilities>
<ogc:Id_Capabilities>
<ogc:FID/>
<ogc:EID/>
</ogc:Id_Capabilities>
</ogc:Filter_Capabilities>
</wfs:WFS_Capabilities>
```

STEP-2

The client can request **GetFeature** with the WFS server's capabilities information in order to get the Feature information. Once a user has obtained a description of the supported FeatureType, the GetFeature request is used to access the metadata associated with one or more feature instances. Subsets of Features can be obtained by providing filtering information in the request. Available filtering capability includes:

- Filtering on a list of specific Feature IDs
- Filtering based on a geographic area (Bounding Box)
- Filtering based on Feature property values (e.g., only return features with a Cloud Cover of less than 5%)

> Example Request:
> https://services.digitalglobe.com/catalogservice/wfsaccess?SERVICE=WFS&**REQUEST=GetFeature**&maxFeatures=25&typeName=DigitalGlobe:FinishedFeature&VERSION=1.1.0&connectid=**<CONNECTID>**&BBOX=39.346038,-105.424805,40.131711,-104.545898
>
> Username and Password parameters may not be applicable depending on your account type. Replace <CONNECTID> with the ConnectID provided by DigitalGlobe. Parameters do not have to be entered in the order shown.

Understanding URL

The Parameters available in the GetFeature request are shown in Table 4.6. The client provides the following: information in a Key-Value Pair (KVP) format, where the "name" field is the key, and the "value" field is the value; the data is supplied in the format "key=value"; for example, "service=WFS".

Response

In response to a valid GetFeature request, the WFS returns a GML document containing zero or more Features that match the request criteria. Each Feature is described by a list of properties contained in a GML document. The available properties are defined in the PropertyName field of the GetFeature Request above. An example Feature, in GML format, is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:FeatureCollection xmlns:DigitalGlobe="http://www.digitalglobe.com"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:ows="http://www.opengis.net/ows"
xmlns:wfs="http://www.opengis.net/wfs" numberOfFeatures="1" timeStamp="2010-10-
29T12:34:08.004Z"
xsi:schemaLocation="http://www.digitalglobe.com
https://services.digitalglobe.com/catalogservice/wfsaccess?CONNECTID=<CONNECTID>&amp;service=WFS&amp;version=1.1.0&amp;request=DescribeFeatureType&amp;typeName=DigitalGlobe:FinishedFeature http://www.opengis.net/wfs
http://services.digitalglobe.com:80/wfsservice/schemas/wfs/1.1.0/wfs.xsd">
Continued…
```

```
…Continued

<gml:featureMember>
        <DigitalGlobe:featureId>3e437526446504b1b8c2e1b9033a3375</DigitalGlobe:fe
atureId>
        <DigitalGlobe:geometry>
                <gml:Polygon srsName="urn:x-ogc:def:crs:EPSG:4326">
                <gml:exterior>
                     <gml:LinearRing>
<gml:posList>39.94609069824219 -105.49018859863281 40.139949798583984 -
105.49018859863281 40.139949798583984 -105.23181915283203 39.94609069824219 -
105.23181915283203 39.94609069824219 -105.49018859863281</gml:posList>
                     </gml:LinearRing>
                 </gml:exterior>
              </gml:Polygon>
          </DigitalGlobe:geometry>
            <DigitalGlobe:offNadirAngle>28.562355</DigitalGlobe:offNadirAngle>
          <DigitalGlobe:sunElevation>49.5492</DigitalGlobe:sunElevation>
          <DigitalGlobe:cloudCover>7.0088723E-4</DigitalGlobe:cloudCover>
          <DigitalGlobe:sunAzimuth>154.9179</DigitalGlobe:sunAzimuth>
<DigitalGlobe:source>QB02</DigitalGlobe:source>
          <DigitalGlobe:sourceUnit>Strip</DigitalGlobe:sourceUnit>
            <DigitalGlobe:productType>Pan Sharpened Natural
Color</DigitalGlobe:productType>
            <DigitalGlobe:groundSampleDistance>0.6</DigitalGlobe:groundSampleDista
nce>
<DigitalGlobe:groundSampleDistanceUnit>Meter</DigitalGlobe:groundSampleDistanceUni
t>
<DigitalGlobe:dataLayer>crisis_event</DigitalGlobe:dataLayer>
    <DigitalGlobe:legacyDescription/>
<DigitalGlobe:outputMosaic>false</DigitalGlobe:outputMosaic>
<DigitalGlobe:colorBandOrder>RGB</DigitalGlobe:colorBandOrder>
<DigitalGlobe:assetName>FINISHED</DigitalGlobe:assetName>
<DigitalGlobe:assetType>PRODUCT_GEOMETRY</DigitalGlobe:assetType>
<DigitalGlobe:legacyId>101001000C4DBF00</DigitalGlobe:legacyId>
<DigitalGlobe:factoryOrderNumber>052405468-10</DigitalGlobe:factoryOrderNumber>
<DigitalGlobe:acquisitionDate>2010-09-17
17:51:33.28</DigitalGlobe:acquisitionDate>
<DigitalGlobe:perPixelX>5.399999998E-6</DigitalGlobe:perPixelX>
<DigitalGlobe:perPixelY>-5.399999998E-6</DigitalGlobe:perPixelY>
<DigitalGlobe:crsFromPixels>EPSG:4326</DigitalGlobe:crsFromPixels>
<DigitalGlobe:url/>
<DigitalGlobe:ageDays>41</DigitalGlobe:ageDays>
<DigitalGlobe:formattedDate>2010-09-17</DigitalGlobe:formattedDate>
</gml:featureMember>
</wfs:FeatureCollection>
```

4.5.1 USING WFS-GETFEATURE TO ACCESS ADDITIONAL METADATA

If additional metadata about either all features in a particular tile, or a single feature included in a particular tile is needed, it can be retrieved from the DigitalGlobe Cloud Services using a WFS **getFeature** Request.

4.5.2 USING WFS TILEMATRIXFEATURE REQUEST

To obtain all of the available metadata, for all of the feature instances that occur in a particular tile, use the WFS *getFeature* request, with a *typeName* of **TileMatrixFeature**.

> Example Request:
> https://services.digitalglobe.com/catalogservice/wfsaccess?request=GetFeature&typeName=**DigitalGlobe:TileMatrixFeature**&SERVICE=WFS&VERSION=1.1.0&connectid=<CONNECTID>&CQL_FILTER=layer='DigitalGlobe:ImageryTileService'and%20tileMatrixSet%20=%20'EPSG:3857'%20and%20tileMatrix%20=%20'EPSG:3857:17'%20and%20row%20=%2058685%20and%20column%20=%2039235
>
> Replace <CONNECTID> with your ConnectID provided by DigitalGlobe.
>
> CQL_FILTER parameters define the exact tile of interest, by using:
> the layer: DigitalGlobe:ImageryTileService
> the tileMatrixSet: EPSG:3857
> the tileMatrix: EPSG:3857:17
> the tile row:58685 for this example
> the tile column: 39235 for this example.

The response is a DigitalGlobe TileMatrixFeature, which provides both summary metadata for the defined tile and detailed metadata for each feature that appears in the tile.

> ➔ Note: The data returned would represent the CURRENT tile that would be produced by the DigitalGlobe Cloud Services system. In order to ensure that this matches the customer tile for the desired metadata, the customer must compare the *tileIdentifier* in the WFS return, with the tileIdentifier for metadata being requested. If the tileIdentifiers do NOT match, then the data in the DigitalGlobe system has changed since the initial tile delivery.

4.5.3 USING WFS FINISHEDFEATURE REQUEST

To obtain the detailed metadata for one or more features contained in a tile, use the WFS *getFeature* request, with a *typeName* of **FinishedFeature**.

> Example Request:
> https://services.digitalglobe.com/catalogservice/wfsaccess?SERVICE=WFS&request=GetFeature&version=1.1.0&typeName=**DigitalGlobe:FinishedFeature**&srsName=EPSG:3857&outputFormat=text/xml;subtype=gml/3.1.1&featureId=(FEATUREID)&connectid=<CONNECTID>
>
> Replace <CONNECTID> with your ConnectID provided by DigitalGlobe.
>
> (FEATUREID) is a comma-delimited list of featureids for which metadata is desired; these can be obtained from the <featureInTileIdentifier> field of the tile_master.lst file.

For the complete list of supported outputFormat types, use the WFS "GetCapabilities".

STEP-3

To get the description of the supported **FeatureType**, the user can use the following URL:

> Example Request:
> https://services.digitalglobe.com/catalogservice/wfsaccess?service=WFS&request=DescribeFeatureType&version=1.1.0&typeName=**DigitalGlobe:FinishedFeature**&connectid=<CONNECTID>&username=<username>&password=<password>
>
> Replace <CONNECTID> with your ConnectID provided by DigitalGlobe.
> Parameters do not have to be entered in the order shown.

Understanding URL

The Parameters available in the DescribeFeatureType request are shown in Table 4.7. The client provides the following information in a Key-Value Pair (KVP) format, where the "name" field is the key, and the "value" field is the value; the data is supplied in the format "key=value"; for example, "service=WFS".

Response

In response to a DescribeFeatureType request, a GML document is returned that defines the elements of the requested Feature Type. The elements of each supported feature type are defined in Table 4.7 and Table 4.8. Where returned data is dependent on the value of elements within the feature, the dependency is noted in the Description column. Example output is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:DigitalGlobe="http://www.digitalglobe.com" xmlns:gml="http://www.opengis.net/gml"
elementFormDefault="qualified" targetNamespace="http://www.digitalglobe.com">
      <xsd:import namespace="http://www.opengis.net/gml"
schemaLocation="http://services.digitalglobe.com:80/wfsservice/schemas/gml/3.1.1/base/gml
.xsd"/>
      <xsd:complexType name="FinishedFeatureType">
            <xsd:complexContent>
                  <xsd:extension base="gml:AbstractFeatureType">
                        <xsd:sequence>
                              <xsd:element maxOccurs="1" minOccurs="0" name="featureId"
nillable="true" type="xsd:string"/>
<xsd:element maxOccurs="1" minOccurs="0" name="geometry" nillable="true"
type="gml:PolygonPropertyType"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="offNadirAngle"
nillable="true" type="xsd:string"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="sunElevation"
nillable="true" type="xsd:string"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="cloudCover"
nillable="true" type="xsd:string"/>
                  <xsd:element maxOccurs="1" minOccurs="0" name="sunAzimuth"
nillable="true" type="xsd:string"/>
                  <xsd:element maxOccurs="1" minOccurs="0" name="source" nillable="true"
type="xsd:string"/>
                  <xsd:element maxOccurs="1" minOccurs="0" name="sourceUnit"
nillable="true" type="xsd:string"/>
                  <xsd:element maxOccurs="1" minOccurs=”0" name=”productType”
nillable=”true” type=”xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs="0" name=”groundSampleDistance” nillable=”true”
type=”xsd:string”/>
                  <xsd:element maxOccurs=”1" minOccurs=”0”
name="groundSampleDistanceUnit" nillable=”true” type=”xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”dataLayer” nillable=”true”
type="xsd:string"/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”legacyDescription” nillable=”true”
type="xsd:string"/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”outputMosaic” nillable=”true”
type="xsd:string"/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”legacyId” nillable=”true”
type="xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”colorBandOrder” nillable=”true”
type="xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”assetName” nillable=”true”
type=”xsd:string”/>


Continued
```

```
…Continued

<xsd:element maxOccurs=”1” minOccurs=”0” name=”assetType” nillable=”true”
type=”xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”factoryOrderNumber” nillable=”true”
type=”xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”acquisitionDate” nillable=”true”
type=”xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”perPixelX” nillable=”true”
type=”xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”perPixelY” nillable=”true”
type=”xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”crsFromPixels” nillable=”true”
type=”xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”url” nillable=”true” type=”xsd:string”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name="ageDays” nillable=”true” type=”xsd:int”/>
<xsd:element maxOccurs=”1” minOccurs=”0” name=”formattedDate” nillable=”true”
type=”xsd:string”/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:element name=”FinishedFeature” substitutionGroup=”gml:_Feature”
type=”DigitalGlobe:FinishedFeatureType”/>
</xsd:schema>
```

Example output follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:DigitalGlobe="http://www.digitalglobe.com"
xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified"
targetNamespace="http://www.digitalglobe.com">
      <xsd:import namespace="http://www.opengis.net/gml"
schemaLocation="http://services.digitalglobe.com:80/wfsservice/schemas/gml/3.1.1/base/
gml.xsd"/>
      <xsd:complexType name="ImageInMosaicFeatureType">
            <xsd:complexContent>
                  <xsd:extension base="gml:AbstractFeatureType">
                        <xsd:sequence>
              <xsd:element maxOccurs="1" minOccurs="0" name="featureId"
nillable="true" type="xsd:string"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="parentFeatureId"
nillable="true" type="xsd:string"/>
               <xsd:element maxOccurs="1" minOccurs="0" name="dataLayer"
nillable="true" type="xsd:string"/>
               <xsd:element maxOccurs="1" minOccurs="0" name="geometry"
nillable="true" type="gml:PolygonPropertyType"/>
               <xsd:element maxOccurs="1" minOccurs="0" name="assetName"
nillable="true" type="xsd:string"/>
               <xsd:element maxOccurs="1" minOccurs="0" name="assetType"
nillable="true" type="xsd:string"/>
               <xsd:element maxOccurs="1" minOccurs="0" name="url" nillable="true"
type="xsd:string"/>

Continued…
```

```
…Continued
                <xsd:element maxOccurs="1" minOccurs="0" name="legacyId"
nillable="true" type="xsd:string"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="acquisitionDate"
nillable="true" type="xsd:string"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="cloudCover"
nillable="true" type="xsd:string"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="offNadirAngle"
nillable="true" type="xsd:string"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="sunElevation"
nillable="true" type="xsd:string"/>
                <xsd:element maxOccurs="1" minOccurs="0" name="sunAzimuth"
nillable="true" type="xsd:string"/>
                 <xsd:element maxOccurs="1" minOccurs="0" name="source" nillable="true"
type="xsd:string"/>
                            </xsd:sequence>
                      </xsd:extension>
            </xsd:complexContent>
      </xsd:complexType>
      <xsd:element name="ImageInMosaicFeature" substitutionGroup="gml:_Feature"
type="DigitalGlobe:ImageInMosaicFeatureType"/>
</xsd:schema>
```

4.6 Service Exceptions

In the event that a web feature service encounters an error while processing a request or receives an unrecognized request, it will generate an XML document indicating that an error has occurred.

An **<ExceptionReport>** element will contain one or more WFS processing exceptions specified using the <**Exception**> element. The mandatory version attribute is used to indicate the version of the service exception report schema. For this version of the specification, this value is fixed at 1.1.0. Individual exception messages are contained within the **<ExceptionText>** element.

The following is an example of an exception report. This exception indicates that the first insert statement failed because of a missing closing XML tag in the request.

```
<?xml version="1.0" ?>
<ExceptionReport version="1.1.0" xmlns="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="owsExceptionReport.xsd">
      <Exception code="999" locator="INSERT STMT 01">
            <ExceptionText>
                  parse error: missing closing tag for element Geom
            </ExceptionText>
      </Exception>
</ExceptionReport>
```

4.7 WFS Layers

TABLE 4.4 OGC LAYERS

OGC LAYER(S)	DESCRIPTION
FinishedFeature	The FinishedFeature Layer provides all available metadata for all content to which the account has access. This includes mosaic and single-strip products, where mosaic products are treated with either single values or ranges of metadata values based on the components of the feature.
ImageInMosaic	This layer provides the detailed metadata for a mosaic product. For example, if multiple images are used to make a cell or other mosaic product, this layer allows determination of the metadata for a particular point within the mosaic. ImageInMosaic is available for some, but not all mosaic products.
TileMatrixFeature	TileMatrixFeature provides the metadata for a single tile that would be retrieved via WMTS. Queries utilize the row/column and zoom constructs to return GML which identifies what features are part of that tile and the geometries are cut to the polygonal outline of the grid.

4.8 API Reference

API reference gives the list of all possible request parameters for every WFS operation and detailed information about corresponding response.

The client should provide the respective information in a Key-Value Pair (KVP) format for every WFS request, where the “name” field is the key, and the "value" field is the value; the data is supplied in the format “key=value”; for example, “service=WFS”.

GetCapabilities

The following table shows all possible request parameters for GetCapabilities operation of WFS server.

TABLE 4.5 UNDERSTANDING URL PARAMETERS FOR WFS GETCAPABILITIES REQUEST

PARAMETER NAME	PARAMETER VALUE	DESCRIPTION
CONNECTID*	**<CONNECTID>** provided by Digital Globe	Value for this parameter is a unique 32 digit alphanumeric value given by DigitalGlobe (Explained under “CONNECTID” of Section 4.4.3). A valid CONNECTID is mandatory for every request.
SERVICE*	WFS	Refer to “SERVICE” in Section 4.4.3.
REQUEST*	GetCapabilities	The value for this parameter should always be “**GetCapabilities**” for step-1.
VERSION*	1.1.0	Refer to “VERSION” in Section 4.4.3.

* mandatory parameter

GetFeature

TABLE 4.6 GETFEATURE REQUEST PARAMETERS

NAME	VALUE	DESCRIPTION
Service*	WFS	Web Feature Service
Version*	1.1.0	Request version
Request*	GetFeature	Request name
resultType	Possible values: result (default) hits	A value of **results** generates full results set response; A value of **hits** returns only a count of the number of features that match the request. **results** is the default
maxFeatures	An integer	the maximum number of features that will be returned; if not specified the default is to return all results
srsName	EPSG:4326 EPSG::3395 EPSG::3857	This is the SRS in which the Feature data will be returned. Only one SRS is supported, and EPSG:4326 the default if no SRS is specified,. Refer SRS in *Section 4.4.3* Complete list of supported values are found in the GetCapabilities response.
typeName*	FinishedFeature TileMatrixFeature ImageInMosaicFeature	The feature type for which results will be returned.
featureId	List of FeatureIds: Feature Ids are the ID Property of the Feature. The format of a	Specific Feature instances for

NAME	VALUE	DESCRIPTION
(Mutually exclusive with FILTER and BBOX)	Feature ID is: 17027d713c0c3d9eaef07e80d959e347	which results will be returned. Note that if any FEATUREIDs are supplied, BBOX and FILTER may NOT be used in the same request
FILTER (Prerequisite: TYPENAME) (Mutually exclusive with FEATUREID and BBOX)	Filter specification including properties and values to be filtered on; Any of the properties returned in the DescribeFeatureType response can be used as Filter parameters Note that if a FILTER is supplied, FEATUREID and BBOX may NOT be used in the same request; however, a BBOX may be supplied within the filter in order to accomplish the same capability	For details of the filter format see the OGC WFS specification
WIDTH and HEIGHT*	WIDTH=1102 HEIGHT=712 Integer number of pixels to be returned in the X (Width) and Y (Height) coordinates	Width and height represents the map image size being requested. **Note**: The width/height values should be proportional to BBOX value, as a different aspect may result in a distorted map.
BBOX (Prerequisite: TYPENAME) (Mutually exclusive with FEATUREID and FILTER.)	Bounding box in GML coordinates; used to specify the geographic extent of the results to be returned; see examples below. Note that if a BBOX is supplied, Filter and FEATUREID may NOT be used in the same request.	Refer BBOX in Section 4.4.3 for more information.
SORTBY	List of **propertyName** [A,D], where: A=Ascending (default) D=Descending	Specifies a sort order for returned results
connectId*	Unique 32-digit alphanumeric value provided by DigitalGlobe.	A ConnectID is required to access DigitalGlobe Cloud Services
CQL_FILTER	Filter parameters appropriate for the requested feature type.	Refer to Section 5 *Common Query Language* on page 38 for a description of CQL filtering.
showTheRasterReturned	TRUE or FALSE.	FALSE is default. If TRUE, then only those features that are used to fulfill the requested BBOX will be returned.

* mandatory parameter

DescribeFeatureType

TABLE 4.7 DESCRIBEFEATURETYPE REQUEST PARAMETERS

NAME	VALUE	DESCRIPTION
Service*	WFS	Web Feature Service.
Version*	1.1.0	Request version.
Request*	DescribeFeatureType	Request name.
typeName	FinishedFeature ImageInMosaicFeature	FinishedFeature and ImageInMosaicFeature are applicable to finished products and not for imagery features.
outputFormat	text/xml; subtype=gml/3.1.1	Format to return; if not supplied the default will

		be returned. gml/3.1.1 is the only format supported at this time and therefore the default.
connectId*	Unique 32-digit alphanumeric value provided by DigitalGlobe	A ConnectID is required to access DigitalGlobe Cloud Services.

* mandatory parameter

The table below displays the response parameters for the DescribeFeatureType request with typename as FinishedFeature.

TABLE 4.8 ELEMENTS OF FEATURE TYPE "FINISHEDFEATURE"

ELEMENT	DATATYPE	EXAMPLE VALUE	DESCRIPTION
featureId	String	17027d713c0c3d9e aef07e80d959e347	A string that uniquely identifies the particular feature; this string is basically a CONNECTID.
geometry	gml:Polygon PropertyType	35.596, 69.519 37.017,69.516 37.024, 69.735 39.958, 69.732 35.596, 69.519	A GML polygon representing the geometry of the Feature instance; the GML includes the Coordinate System of the polygon.
source	String	WV01	The sensor used to collect the imagery used to make the product described by the Feature.
sourceUnit	String	Strip	The type of product described by the Feature.
productType	String	Panchromatic	The type of product described by the feature. Pan Sharpened Natural Color Panchromatic
groundSampleDistance	String	0.5	The GSD of the product described by the Feature.
groundSampleDistanceUnit	String	Meter	The units of the field groundSampleDistance.
dataLayer	String	Metro	Defines the layer in which the product described by the Feature exists. **crisis_event** (FirstLook - Natural and human caused disasters) **country_coverage** (Global Basemap Countries) **metro** (Global Basemap Cities) **aerial_cells** (Global Basemap Aerial Cells. U.S. and Western Europe Coverage, only) **aerial_markets** (GBM Aerial Markets/urban areas. US and Western EU) **aerial_cir_cells** (Color infrared version of aerial cells) **aerial_cir_markets** (Color infrared version of aerial markets) **monitoring** (AssuredLook – monitoring of selected sites)
legacyDescription	String	Maracaibo	A text description of the product described by the Feature.

ELEMENT	DATATYPE	EXAMPLE VALUE	DESCRIPTION
outputMosaic	Boolean	True	“True” if the product described by the Feature consists of multiple images, as in the case of a Region Tile; "False" if ProductType is Strip.
colorBandOrder	String	RGB	Null for panchromatic data; otherwise the order in which the spectral bands occur in the product described by the feature.
assetName	String	FINISHED	The Name of the catalog that contains the Product described by the feature.
assetType	String	PRODUCT_ GEOMETRY	The Type of the catalog that contains the Product described by the feature.
legacyId	String	10200100077D4A00	If source_unit is “Strip”, this is the DigitalGlobe internal identifier of the image used to make the product described by the feature; for other source_units, this field is null.
factoryOrderNumber	String	052187237-10	The order number that identifies the product described by this feature within the Digitalglobe internal systems.
acquisitionDate	String	2009-05-27 01:20:43.63599	For source_unit=”Strip”, the GMT of acquisition of the image used to make the product described by the Feature; for other source_unit values, the GMT of acquisition of the **oldest** image contained in the product described by the Feature.
perPixelX	Float	4.5E-6	The pixel size in the x direction; the unit is determined by the "crsFromPixels" value.
perPixelY	Float	4.5E-6	The pixel size in the y direction; the unit is determined by the “crsFromPixels” value.
crsFromPixels	String	EPSG:4326	The Coordinate Reference System of the Product described by the Feature.
url	String	See below	A URL that points to a browse image of the product described by the Feature.
cloudCover	String	0.5	The cloud cover percentage of the image used to make the product described by the Feature; this will be 0 for products made from more than one image.
offNadirAngle	String	37.6	The off-nadir angle of the image used to create the product described by the Feature; this will be null for products made from more than one image.
sunElevation	String	83.5	The elevation angle of the sun for the image used to make the product described by the Feature; this will be null for products made from more than one image.
sunAzimuth	String	-117.3	The azimuth angle of the sun for the image used to make the product described by the Feature; this will be null for products made from more than one image.

ELEMENT	DATATYPE	EXAMPLE VALUE	DESCRIPTION
CE90Accuracy	String	4.21 meters	The geographic accuracy of the product described by the feature, in Meters, as a CE90 value. CE90, Circular Error of 90%, is commonly used for quoting and validating geodetic image registration accuracy. A CE90 value is the minimum diameter of the horizontal circle that can be centered on all photo-identifiable Ground Control Points (GCPs) and also contain 90% of their respective twin counterparts acquired in an independent geodetic survey.
RMSEAccuracy	String	7.63	The geographic accuracy of the product described by the feature, in Meters, as a 2d RMS Error value. RMSE, Root Mean Squared Error, is commonly used for quoting and validating geodetic image registration accuracy. A RMSE value is a single summary statistic that describes the square-root of the mean horizontal distance between all photo-identifiable GCPs and their respective twin counterparts acquired in an independent geodetic survey.
spatialAccuracy	String	1:50,000	The accuracy of the product described by the feature, using the NMAS accuracy scales. National map accuracy standards are specifications of accuracy standards for well-defined map points on published maps that are specified by the U.S. Geological Survey and revised by the U.S. Bureau of the Budget.
area	String	385	The area, in square kilometers, of the product described by the Feature. This area is computed using a Cylindrical Equal Area projection (EPSG:9835).
ingestDate	String	2010-06-21 13:20:43.63599	The GMT time at which the feature was made available in the DGCS platform.
Age in Days	String	ageDays<14 (Date Format: MM/DD/YYY)	The age (in days) of the image from today's date.

The table below shows the response parameter for the DescribeFeatureType request with typename as ImageInMosaicFeature.

TABLE 4.9 ELEMENTS OF FEATURETYPE IMAGEINMOSAICE FEATURE

ELEMENT	DATATYPE	EXAMPLE VALUE	DESCRIPTION
featureId	String	17027d713c0c3d9e aef07e80d959e347	A string that uniquely identifies the particular Feature.
parentFeatureId	String	17027d713c0c3d9e aef07e80d959e586	The featureId of the mosaic image that this Feature is associated with.
geometry	gml:Polygon	Example: 35.596, 69.519 37.017,69.516 37.024,	A GML polygon describing the geometry of the Feature instance, in WGS84 Lat/Long (EPSG:4326)

ELEMENT	DATATYPE	EXAMPLE VALUE	DESCRIPTION
		69.735 39.958, 69.732 35.596, 69.519	coordinates.
legacyId	String	10200100077D4A00	The DigitalGlobe system identifier for the image strip described by the feature.
source	String	WV01	The sensor used to collect the image strip described by the Feature.
acquisitionDate	DateTime	Example: 2009-05-27 01:20:43.63599	The GMT of acquisition of the image described by the Feature.
cloudCover	Float	0.32	The cloud cover percentage of the image described by the Feature.
offNadirAngle	Float	20.0	The average off-nadir angle of the image strip described by the Feature.
sunElevation	Float	0.0-90.0 or null	The elevation angle of the sun for the image strip described by the Feature.
sunAzimuth	Float	+/- 180.00 or null	The azimuth angle of the sun for the image strip described by the Feature.
createdContentIdentifier	String	2000037383009	The DigitalGlobe catalog identifier of the mosaic product with which this feature is associated.

5 Common Query Language

CQL (Common Query Language) is a formal language for representing queries to Information Retrieval systems such as web indexes, bibliographic catalogues and museum collection information. The design objective is that queries be human readable and writable, and that the language be intuitive while maintaining the expressiveness of more complex languages.

Traditionally, query languages have fallen into two camps: Powerful, expressive languages, not easily readable or writable by non-experts (e.g. SQL, PQF, and XQuery); or simple and intuitive languages not powerful enough to express complex concepts (e.g. CCL and Google).

CQL tries to combine simplicity and intuitiveness of expression for simple, every day queries, with the richness of more expressive languages to accommodate complex concepts when necessary.

DGCS – WFS uses CQL in the request to filter the returned response. This capability provides enhanced flexibility and better user control over response results for sophisticated users of the services.

5.1 CQL Filter Principle

The following examples explain CQL Filter principles and their usage.

Example 1:

To request all FinishedFeatures that have existed for less than 14 days, the request would look like this:

> URL:
> https://services.digitalglobe.com/catalogservice/wfsaccess?SERVICE=WFS&REQUEST=GetFeature%20&typeName=DigitalGlobe:FinishedFeature&VERSION=1.1.0&CONNECTID=<CONNECTID>&srsName=EPSG:4326&CQL_Filter=ageDays<14
>
> Username and Password parameters may not be applicable depending on your account type.
> Replace <CONNECTID> with the ConnectID provided by DigitalGlobe. Parameters do not have to be entered in the order shown.
> The response will have all FinishedFeatures that have existed for less than 14 days as the parameter CQL_Filter is set to ageDays<14.

Example 2:

To request all FinishedFeatures that have a cloud cover of less than 5%, the request would look like this:

> URL:
> https://services.digitalglobe.com/catalogservice/wfsaccess?SERVICE=WFS&REQUEST=GetFeature%20&typeName=DigitalGlobe:FinishedFeature&VERSION=1.1.0&CONNECTID=<CONNECTID>&srsName=EPSG:4326&CQL_Filter=cloudCover<.05
>
> Replace <CONNECTID> with the ConnectID provided by DigitalGlobe. Parameters do not have to be entered in the order shown.
> The response will have all FinishedFeatures that have less than 5% as the parameter CQL_Filter is set to cloudCover<.05.

Example 3:

To request all FinishedFeatures that have the formatted date greater than '2011-02-06', the request would look like this:

> URL:
> https://services.digitalglobe.com/catalogservice/wfsaccess?SERVICE=WFS&REQUEST=GetFeature%20&typeName=DigitalGlobe:FinishedFeature&VERSION=1.1.0&CONNECTID=<CONNECTID>&srsName=EPSG:4326&CQL_Filter=formattedDate>'2011-02-06'
>
> Replace <CONNECTID> with the ConnectID provided by DigitalGlobe. Parameters do not have to be entered in the order shown.
> The response will have all FinishedFeatures that have the formatted date greater than 2011-02-06 as the parameter CQL_Filter is set to formattedDate>'2011-02-06'.

5.2 How to Combine Two Attributes in a CQL Filter

There are two approaches to combining two attributes in a CQL Filter.

- Post syntax (xml); get syntax (CQL-based). The filter can be an arbitrarily complex expression, such as:

- property = value

  or
- (property = value) AND (otherProperty > otherValue)

sub-expressions can be put together with AND and OR to form more complex expressions.

Table 5.1 lists the possible CQL Filter parameters.

TABLE 5.1 CQL FILTER PARAMETERS

ELEMENT	DATATYPE	EXAMPLE VALUE	DESCRIPTION
featureId	String	17027d713c0c3d9e aef07e80d959e347	A string that uniquely identifies the particular feature to be included or excluded from returned results.
geometry	gml:Polygon PropertyType	35.596, 69.519 37.017,69.516 37.024, 69.735 39.958, 69.732 35.596, 69.519	A GML polygon representing the geometry of the Feature instance; the GML includes the Coordinate System of the polygon.
source	String	WV01	The sensor used to collect the imagery used to make the product described by the Feature.
sourceUnit	String	Strip	The type of product described by the Feature.
productType	String	Panchromatic	The type of product described by the feature. • Pan Sharpened • Natural Color • Panchromatic
groundSampleDistance	String	0.5	The GSD of the product described by the Feature.
groundSampleDistanceUnit	String	Meter	The units of the field groundSampleDistance.
dataLayer	String	Metro	Defines the layer in which the product described by the Feature exists. **crisis_event** (FirstLook - Natural and human caused disasters) **country_coverage** (Global Basemap Countries) **metro** (Global Basemap Cities) **aerial_cells** (Global Basemap Aerial Cells. U.S. and Western Europe Coverage, only) **aerial_markets** (GBM Aerial Markets/urban areas. US and Western EU) **aerial_cir_cells** (Color infrared version of aerial cells) **aerial_cir_markets** (Color infrared version of aerial markets) **monitoring** (AssuredLook – monitoring of selected sites)
legacyDescription	String	Maracaibo	A text description of the product described by the Feature.
outputMosaic	Boolean	True	“True” if the product described by the Feature consists of multiple images, as in the case of a Region Tile; “False” if ProductType is Strip.

ELEMENT	DATATYPE	EXAMPLE VALUE	DESCRIPTION
colorBandOrder	String	RGB	Null for panchromatic data; otherwise the order in which the spectral bands occur in the product described by the feature.
assetName	String	FINISHED	The Name of the catalog that contains the Product described by the feature.
assetType	String	PRODUCT_ GEOMETRY	The Type of the catalog that contains the Product described by the feature.
legacyId	String	10200100077D4A00	If source_unit is “Strip”, this is the DigitalGlobe internal identifier of the image used to make the product described by the feature; for other source_units, this field is null.
factoryOrderNumber	String	052187237-10	The order number that identifies the product described by this feature within the Digitalglobe internal systems.
acquisitionDate	String	2009-05-27 01:20:43.63599	For source_unit=”Strip”, the GMT of acquisition of the image used to make the product described by the Feature; for other source_unit values, the GMT of acquisition of the **oldest** image contained in the product described by the Feature.
perPixelX	Float	4.5E-6	The pixel size in the x direction; the unit is determined by the “crsFromPixels” value.
perPixelY	Float	4.5E-6	The pixel size in the y direction; the unit is determined by the “crsFromPixels” value.
crsFromPixels	String	EPSG:4326	The Coordinate Reference System of the Product described by the Feature.
url	String		A URL that points to a browse image of the product described by the Feature.
cloudCover	String	0.5	The cloud cover percentage of the image used to make the product described by the Feature; this will be 0 for products made from more than one image.
offNadirAngle	String	37.6	The off-nadir angle of the image used to create the product described by the Feature; this will be null for products made from more than one image.
sunElevation	String	83.5	The elevation angle of the sun for the image used to make the product described by the Feature; this will be null for products made from more than one image.
sunAzimuth	String	-117.3	The azimuth angle of the sun for the image used to make the product described by the Feature; this will be null for products made from more than one image.
CE90Accuracy	String	4.21 meters	The geographic accuracy of the product described by the feature, in Meters, as a CE90 value. CE90, Circular Error of 90%, is

ELEMENT	DATATYPE	EXAMPLE VALUE	DESCRIPTION
			commonly used for quoting and validating geodetic image registration accuracy. A CE90 value is the minimum diameter of the horizontal circle that can be centered on all photo-identifiable Ground Control Points (GCPs) and also contain 90% of their respective twin counterparts acquired in an independent geodetic survey.
RMSEAccuracy	String	7.63	The geographic accuracy of the product described by the feature, in Meters, as a 2d RMS Error value. RMSE, Root Mean Squared Error, is commonly used for quoting and validating geodetic image registration accuracy. A RMSE value is a single summary statistic that describes the square-root of the mean horizontal distance between all photo-identifiable GCPs and their respective twin counterparts acquired in an independent geodetic survey.
spatialAccuracy	String	1:50,000	The accuracy of the product described by the feature, using the NMAS accuracy scales. National map accuracy standards are specifications of accuracy standards for well-defined map points on published maps that are specified by the U.S. Geological Survey and revised by the U.S. Bureau of the Budget.
area	String	385	The area, in square kilometers, of the product described by the Feature. This area is computed using a Cylindrical Equal Area projection (EPSG:9835).
ingestDate	String	2010-06-21 13:20:43.63599	The GMT time at which the feature was made available in the DGCS platform.
Age in Days	String	ageDays<14 (Date Format: MM/DD/YYY)	The age (in days) of the image from today's date.
Catalog ID	String	catalogIdentifier=%27102001 0007808500%27	The catalog ID for the image strip.

Glossary

AOI

Area of Interest. The area on the Earth that you want to view.

Bilinear Interpolation

Bilinear interpolation uses the value of the four nearest cell centers to determine the value on the output raster. The new value is a weighted average of these four values, adjusted to account for their distance from the center of the output cell. The result is a smoother-looking surface than provided by “nearest neighbor”.

Bicubic Interpolation

Bicubic interpolation combines data points on a two-dimensional grid. This method outputs the smoothest surface of all interpolation methods.

Geographic Projection

Maps longitudes as straight vertical lines and latitudes as straight horizontal lines all spaced out consistently for constant intervals.

GeoTIFF format

A GeoTIFF file is a TIFF file that is embedded with geographic data tags.

GML

Geography Markup Language. GML is XML code used to express geographical features.

JPEG2000 format

The JPEG2000 format is a JPEG format that was introduced in the year 2000. It has considerable advantages over basic JPEG format including error resilience and progressive transmission.

MrSid format

Multi-Resolution Seamless Image Database. This format compresses large raster images while maintaining the image quality.

National Imagery Transmission Format

See NITF format.

Nearest Neighbor Interpolation

Uses the value of the closest point and disregards all other values, yielding a piecewise-constant interpolant.

NITF format

National Imagery Transmission Format. A United States Department of Defense standard for transmitting and storing digital imagery.

OGC

Open GIS Consortium. An international standards organization comprised of commercial, governmental, nonprofit and research organizations. They support geospatial content development as well as data processing and sharing.

OWS

OGC Web Service Common.

Seamlines

Seamlines are the lines at which two separate images overlap. These overlapping images can be blended along the seamline to show a more uniform image.

Universal Transverse Mercator Geographic Coordinate System

See UTM.

UTM

Universal Transverse Mercator Geographic Coordinate System. UTM utilizes a two-dimensional Cartesian system to specify locations on the Earth’s surface.

WCS

Web Coverage Service.

WebCGM

Web Computer Graphics Metafile.

WFS

Web Feature Service.

WMS

Web Map Service.

WMTS

Web Map Tile Service.

Index

AbstractFeature, 9
AbstractFeatureType, 9
API reference, 32
area of interest, defined, 42
Base URL, 20
BBOX, 21
bicubic interpolation, defined, 42
bilinear interpolation, defined, 42
bounding box. *See* BBOX
combining attributes in CQL filter, 38
common query language, 38
CONNECTID, 20
CQL filter
  combining attributes, 38
  principle, 38
CQL, overview, 38
DescribeFeatureType, request parameters, 33
dictionary schema
  details, 10
  document location, 10
EXCEPTIONS, 21
FinishedFeature
  elements, 34
  examples, 38
  request, 28
geographic projection, defined, 42
geography markup language. *See* GML
GeoTIFF, defined, 42
GET query
  reserved characters, 18
GET request
  components, 18
GetCapabilities
  request, 22
  request parameters, 32
  response, 23
GetFeature
  request, 26
  request parameters, 32
  using to retrieve metadata, 27
getting additional metadata, 27
GML
  defined, 42
  overview, 8
GML schema
  features, 9
  overview, 8
GML-based request, 13
HTTP
  request, 16
  response, 19
HTTP GET, 16
HTTP POST, 18
HTTPS, 19
ImageInMosaic feature, elements, 36
integration procedure, 22
JPEG2000, defined, 42
layers, 31
MrSid format, defined, 42
nearest neighbor, defined, 42
NITF format, defined, 42
OGC
  abstract specification, 7
  defined, 42
  overview, 6
  process, 6
  standards, 6
OGC reference model. *See* ORM
OGC standards, 5
ORM, 7
OWS, defined, 42
REQUEST, 21
request parameters
  overview, 20
  rules, 22
requesting GetCapabilities, 22
requesting GetFeature, 26
seam lines, defined, 42
SERVICE, 20
UTM, defined, 42
VERSION, 20
WCS, defined, 43
Web Feature Service. *See* WFS
WebCGM, defined, 43
WFS
  client-server architecture, 15
  defined, 43
  introduction, 15
  layers, 31
  overview, 5
  service details, 15
WMS, defined, 43
WMTS, defined, 43