# DigitalGlobe™

# Web Map Service Developer Guide

Cloud Services | August 2013

# Table of Contents

# List of Figures

# List of Tables

# 1   Introduction

## 1.1  About This Document

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

## 1.2  Targeted Audience

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

## 1.3  What is WMS?

WMS is a standard protocol for serving georeferenced map images over the Internet. These images are generated by a map server using data from a GIS database. A map is a visual representation of geodata, not the data itself. These maps are generally rendered in a pictorial format in the following formats: PNG, GIF or JPEG or occasionally as vector-based graphical elements in Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM). The specification was developed and first published by the OGC in 1999.

OGC released WMS version 1.0.0 in April 2000, followed by version 1.1.0 in June 2001 and version 1.1.1 in January 2002. The latest version of WMS is 1.3.0 which is released by OGC in January 2004.

WMS is a widely supported format for maps and GIS data accessed via the Internet and loaded into client-side GIS software. Major commercial GIS and mapping software that support WMS include ESRI® ArcGIS™ products, MapInfo® Professional, Google Earth™ mapping service. Open source software that support WMS include Quantum GIS, uDig, Autodesk MapGuide® Open Source and OpenLayers.

## 1.4  References

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

WEB MAP SERVICE – DEVELOPER GUIDE

# 2 Open Geospatial Consortium (OGC)

## 2.1 About OGC

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

A predecessor organization, OGF (Open GRASS Foundation) was established 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 Web Services by following the process of identifying and addressing existing problems in the GIS world. The OGC follows this process:

1. Identify Problem
2. Craft Solution
3. Evaluate 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 Internet.
- Delivering data to different systems easily.
- Common language regarding geospatial data or services.
- Finding and pulling together data from our automated sensors.

## 2.3 Details about Interoperability

Interoperability, at a technical level, refers to the ability for a system or components of a system to provide information portability and inter-application as well as cooperative process control. Interoperability comprises intercommunication at communication level protocol, hardware, software, and data compatibility layers.

Interoperability, in the context of the OpenGIS Specification Program, is software components operating reciprocally (working with each other) to overcome tedious batch conversion tasks, import/export obstacles, and distributed resource access barriers imposed by heterogeneous processing environments and heterogeneous data. Interoperability, with respect to geoprocessing, refers to the ability of digital systems to 1) freely exchange all kinds of spatial information and 2) cooperatively, over networks, run software capable of manipulating such information.

Organizations and companies have been providing online mapping services for years. These Web mapping systems have been implemented as a set of proprietary systems. As a result of this isolated development, online mapping services from different vendors are not interoperable. Thus, many technology islands are created and preserved, and many users are locked into single-vendor solutions. This situation (lack of interoperability) is slowly improving but, unfortunately, most Web mapping applications today are still inseparably tied to a specific server implementation. In other words, the Web client is hard-coded to interact with a particular vendor's proprietary map server implementation.

Figure 2.1 shows a scenario where the user must run three different Web applications in order to access the data and functionality provided by three different server implementations. In this situation, there is very little interoperability or reuse of the Web client and server implementations. Only Web client 1 enables access to more than one database. Unfortunately, Web client 1 may not provide all the functionality that Web client 2 and Web client 3 offer. Even with Web client 1's ability to access data of interest from multiple databases, the user must still run three different applications from the different Web clients to perform a given task.
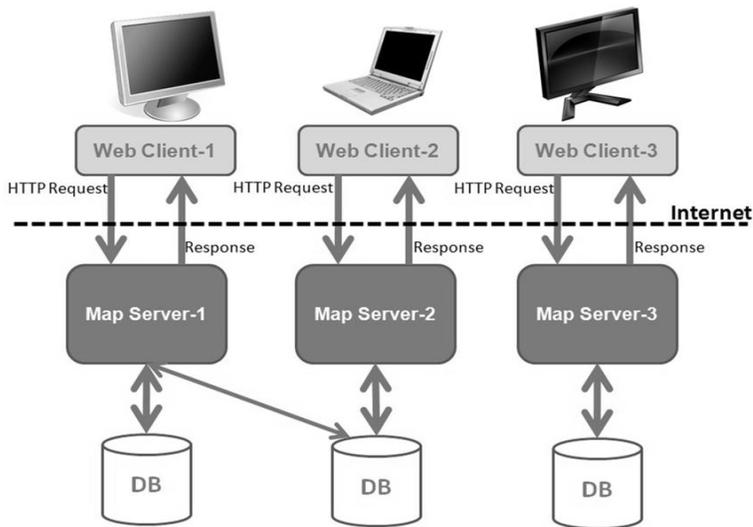
**FIGURE 2.1 INTEROPERABILITY**

To address this problem, the OGC developed a non-proprietary Web mapping approach based on open interfaces, encodings and schemas. The OGC Specification Program and Interoperability Program provide an industry consensus process to plan, develop, review and officially adopt OpenGIS Specifications for interfaces, encodings and schemas that enable interoperable geoprocessing services, data, and applications.

With standards-based interoperable Web mapping, each map server implements a common interface, a messaging protocol such as the WMS interface for accepting requests and returning responses. Now, the same client has Web access to potentially all available map servers and multiple data sources, where each map server is accessed by a client through the common interface. This concept of interoperable, distributed mapping systems is portrayed in Figure 2.2. This approach allows, among other things, the user to run a single client that accesses all the capabilities of each server. This enables a more open application environment where the best features of available Web services can be flexibly combined in innovative and previously unimagined ways to solve novel and increasingly complex problems.
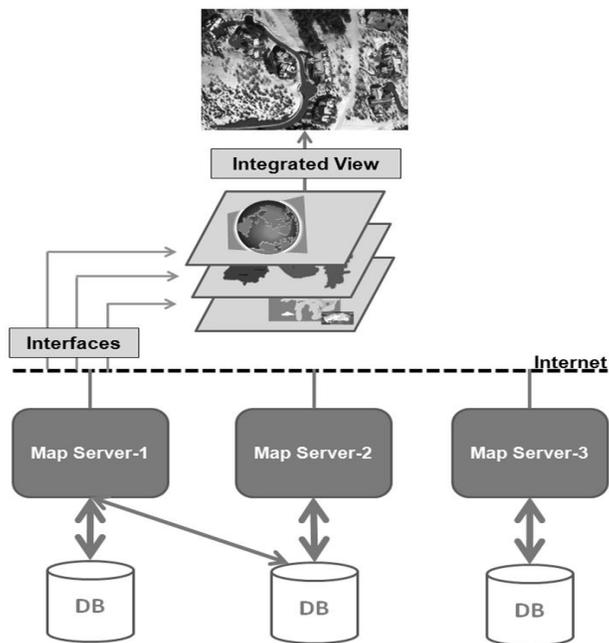


**FIGURE 2.2 INTEGRATED VIEW**

WEB MAP SERVICE – DEVELOPER GUIDE

## 2.4  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 OGC and have been developed by the membership to address specific interoperability challenges. The OGC documents are available to everyone at no cost. Table 2.1 lists the 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. |

## 2.5  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 when interoperability has reached the following stage. When implemented by two software engineers in ignorance of each other, the resulting components plug and play with each other at that interface.

## 2.6  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.

## 2.7  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 purpose 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

➔ Visit the following link for detailed information on OGC standards and specifications:
http://www.opengeospatial.org/standards

# 3  DGCS – Web Map Service

## 3.1  Introduction

The DigitalGlobe Web Map Service is an interoperable, distributed web mapping system which defines a set of functions that clients may use to achieve WMS capabilities. Any client making requests that conform to the OGC WMS specification can interact with DGCS WMS server. A simple and typical example of the structure of a web mapping application is a web-based client-server architecture, as illustrated in Figure 3.1.



**FIGURE 3.1  A TYPICAL STRUCTURE OF A DGCS-WMS APPLICATION**

In a DGCS web mapping scenario, the client application requests desired information from the web map server. The map server retrieves from the database the appropriate layers of geo-feature data for the specified spatial domain. From that, it generates a map, which is a simple graphic image (i.e. GIF or PNG) that can be viewed directly in a graphical web browser or other pictorial software. The client and web map server interact using Hypertext Transfer Protocol Secure (HTTPS).

## 3.2  WMS Client-Server Architecture

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

WMS server accepts requests from WMS client and viewer client in the form of HTTP 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 WMS server to generate GML documents or draw maps.
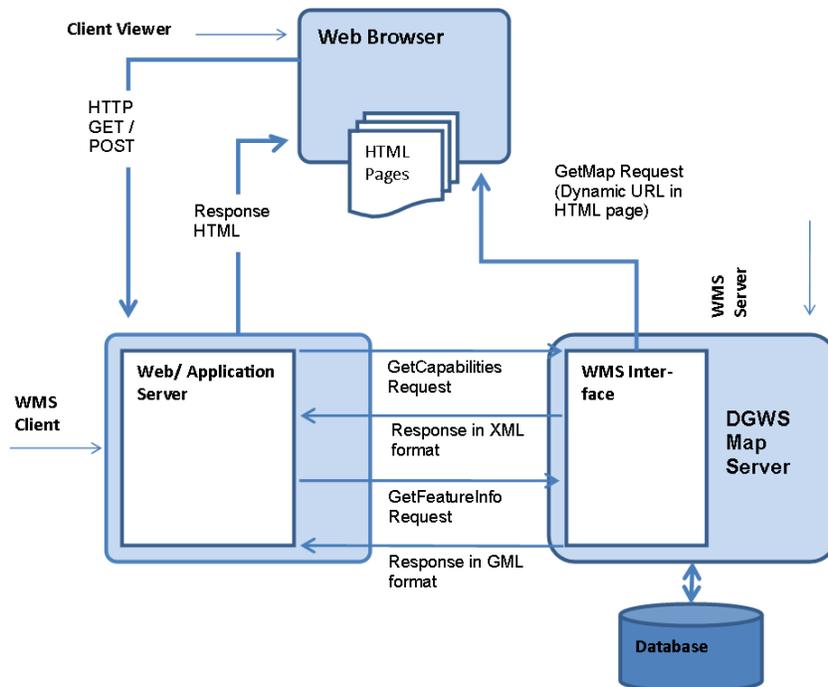


**FIGURE 3.2  SAMPLE WMS CLIENT SERVER APPLICATION**

WEB MAP SERVICE – DEVELOPER GUIDE

## 3.3  WMS Service Details

The DigitalGlobe WMS supports OGC WMS specification version 1.1.1. The DigitalGlobe WMS supports KVP request encoding only; no Simple Object Access Protocol (SOAP) or other protocols are supported.

The DigitalGlobe WMS provides raster imagery data at multiple resolutions in various formats for use in GIS applications that support the WMS standard. The DigitalGlobe WMS supports the following operations:

### GetCapabilities

The GetCapabilities request is used to obtain information about the supported map layers, including various imagery layers and metadata layers.

### GetMap

The GetMap request is used to retrieve map images of the layers contained in the Online Catalogs.

### GetFeatureInfo

The GetFeatureInfo request is used to obtain metadata (information) about the features displayed in map images retrieved via GetMap requests.

### GetLegendGraphic

The GetLegendGraphic request is used to obtain information about the display styling of the map layers, specifically the metadata layers.

The DigitalGlobe WMS does NOT support the following optional capabilities:
- DescribeLayer operation
- GetStyles operation
- PutStyles operation

As stated above, the DigitalGlobe WMS supports map layers of both imagery and imagery metadata. The data for these layers is supplied in the following way:

When metadata layers are requested at smaller map scales (smaller than approximately 1:230,000) an imagery bounding box is returned. When a map scale larger than this is requested, only an image date is returned.

When imagery layers are requested at smaller map scales (smaller than approximately 1:200,000) no imagery is returned. At map scales larger than this, map images are returned.

## 3.4  Basic Service Elements

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

### 3.4.1  HTTP REQUEST

In the client-server computing model, HTTP functions as a request-response protocol. Specifically, a web browser 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 (i.e. HTML files and images) 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

WEB MAP SERVICE – DEVELOPER GUIDE

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 3.1 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-WMS specification in order to construct a valid request URL. Refer to Table 3.2 for reserved characters, per HTTP rules.

**TABLE 3.1  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 mandated for each operation is described in Table 3.8, Table 3.9, and Table 3.10. |

**TABLE 3.2  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. A DGCS-WMS does not require additional parameters to be appended to the URL in order to construct a valid target for the Operation request.

## HTTPS

In addition to or instead of offering web map services using the HTTP protocol, DigitalGlobe offers web map 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.

## 3.4.2  HTTP RESPONSE

Upon receiving a valid HTTP request, the service sends a response corresponding to the request exactly 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 issues a Service Exception.

Response objects are accompanied by the appropriate Multipurpose Internet Mail Extensions (MIME) type for that object. Different types for operation responses and service exceptions are depicted in

Table 3.3.

**TABLE 3.3  VALUES FOR OUTPUT FORMAT ATTRIBUTE**

| MIME TYPE | DOCUMENT CONTENT |
|---|---|
| application/vnd.ogc.wms_xml | WMS Capabilities XML |
| application/vnd.ogc.gml | Geography Markup Language XML [1] |
| application/vnd.ogc.se_xml | Service Exception XML |
| application/vnd.ogc.se_inimage | Image overwritten with Exception message. |
| application/vnd.ogc.se_blank | Blank image because Exception occurred. |
| Image/jpeg | Image Object Response to GetMap operation |
| Image/png | Image Object Response to GetMap operation |

## 3.5  Request Parameters

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

### Base URL

For every request to the DigitalGlobe WMS server, the client needs to append parameters to the base URL. DigitalGlobe provides different services like WMS, WMTS, WFS and WCS, which use the common base URL as described below.

**Base URL:** https://services.digitalglobe.com/mapservice/wmsaccess

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

### ConnectID

CONNECTID is a parameter name that needs to be appended to the base URL mentioned above. This parameter is a unique 32-digit alphanumeric value. It is a mandatory parameter which should be part of every request the client makes with the server. Please contact DigitalGlobe to get your unique ConnectID.

ConnectID format: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx where **x** → alphanumeric value

### Service

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

**Example:** service=WMS

### 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

WEB MAP SERVICE – DEVELOPER GUIDE

form "**x.y.z**". The numbers "y" and "z" will never exceed 99. Each Web Service provided by DigitalGlobe is numbered independently, per respective OGC specification standards. The latest version of DigitalGlobe WMS implemented for the OGC specification is 1.1.1.

The version number appears in two places:
- In response XML of the GetCapabilities request describing WMS service
- In the parameter list of client requests to the WMS service

In response to a GetCapabilities request containing a version number, a WMS 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 labels the response accordingly. Please refer to *OpenGIS Web Map Service (WMS) Implementation Specification* by the OGC for negotiation rules.

**Example:** version=1.1.1 (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 Map Service. Refer to WMS Service Details on page 11 for different operations supported by DigitalGlobe WMS, along with the descriptions.

**Example:** request=GetCapabilities

## FORMAT

The FORMAT parameter specifies the output format of the response to a request operation. Formats are expressed in both Capabilities XML and in operation requests using MIME types. Each Operation has a distinct list of supported formats. Some formats may be offered by several operations, and are then duplicated as needed in each list. If a request contains a format not offered by WMS server, the server throws a Service Exception (with code "InvalidFormat"). Refer to WMS Service Details on page 11 for different types of formats supported by DigitalGlobe WMS for different response types.

**Example:** format=image/jpeg

## EXCEPTIONS

The EXCEPTIONS parameter indicates the format in which the client wishes to be notified of Service Exceptions. The only value of the EXCEPTIONS parameter for WMS 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.

Table 3.7 shows the exception codes defined by OGC and implemented by DigitalGlobe for different WMS operations.

**Example**: exceptions= application/vnd.ogc.se_xml

## SRS (Spatial Reference System)

The SRS is a text parameter that names a horizontal coordinate reference system code. The name includes a namespace prefix, a colon, a numeric identifier, and possibly a comma followed by additional parameters. The DigitalGlobe WMS implementation is defined with EPSG and AUTO namespaces. If a request contains an SRS not offered by a DigitalGlobe WMS server, the server throws a Service Exception (code = "InvalidSRS").

The EPSG namespace makes use of the European Petroleum Survey Group tables, which define numeric identifiers (the EPSG "CRS code," corresponding to the field "COORD_REF_SYS_CODE" in the EPSG database) for many common projections and which associate projection or coordinate metadata (such as measurement units or central meridian) for each identifier. An SRS name in the EPSG namespace includes the prefix and the identifier. This format is used both as the value of the SRS parameter in a service request and as the value of an <SRS> element in the Capabilities XML.

When the SRS parameter specifies a Geographic Coordinate Reference System, e.g., "EPSG:4326", the returned image is implicitly projected using a pseudo-Plate Carrée projection that plots longitude along the X-axis and latitude along the Y-axis. The BBOX request parameter values for such a coordinate reference system shall be specified in the order minimum longitude, minimum latitude, maximum longitude, and maximum latitude.

WEB MAP SERVICE – DEVELOPER GUIDE

**Example:** SRS=EPSG: 4326

➔ NOTE: In the absence of a specified SRS value, the default value EPSG:4326 would be taken.

Possible values for SRS depend on the profile of the account. Applicable values can be seen in the response of WMS GetCapabilities request.

### Bounding Box (BBOX)

The Bounding Box is a set of four comma-separated decimal, scientific notation or integer values representing the georeferenced 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 (as in Figure 3.3): minimum X is the left edge, maximum X the right, minimum Y the bottom, and maximum Y the top. The relation of the Bounding Box to the image pixel matrix is shown in the figure: 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.
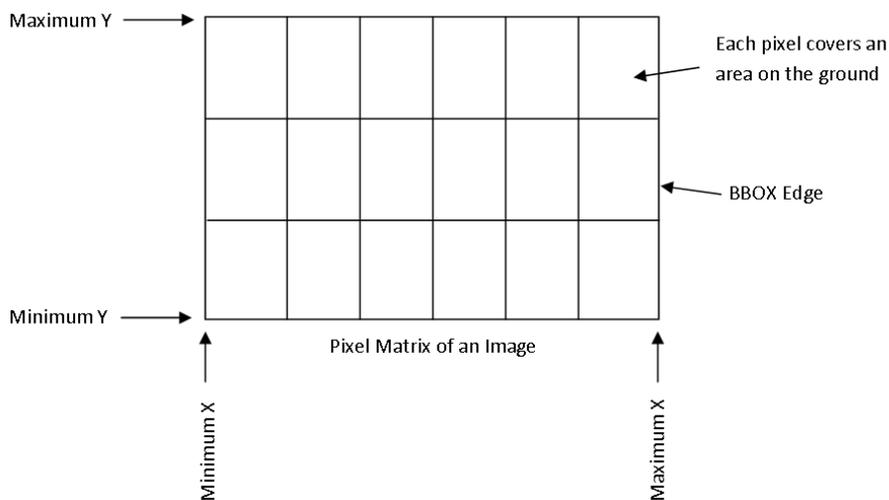


**FIGURE 3.3 REPRESENTATION OF BOUNDING BOX**

**Example: BBOX=**-88.1035780267704,40.4568762655891,-88.0928025063267,40.4638383078358

## 3.6 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 WMS GetMap) 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.

WEB MAP SERVICE – DEVELOPER GUIDE

## 3.7 Integration Procedure

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

### STEP-1

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

> Name: GetCapabilities
> Example URL:
> https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECTID>&service=WMS&request=GetCapabilities&version=1.1.1&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 are not required to be in the same order as shown above.

### Understanding URL

The URL shown above contains Base URL and vital parameters as explained in Section 3.5 on page 13. The key parameter for this request is "**request=GetCapabilities**", which fetches the capabilities of Web Map Service and responds in the form of XML data.

**TABLE 3.4  UNDERSTANDING URL PARAMETERS FOR WMS GETCAPABILITIES REQUEST**

| PARAMETER NAME | PARAMETER VALUE | DESCRIPTION |
|---|---|---|
| CONNECTID* | **<CONNECTID>** provided by DigitalGlobe | The value for this parameter is a unique 32-digit alphanumeric value assigned by DigitalGlobe (Explained under CONNECTID in Section 3.5 on page 13). A valid CONNECTID is mandatory for every request. |
| SERVICE* | WMS | Refer to SERVICE in Section 3.5 on page 13 for details. |
| REQUEST* | GetCapabilities | The value for this parameter should always be "**GetCapabilities**" for step-1. |
| VERSION | 1.1.1 | Refer to VERSION in Section 3.5 on page 13 for details. |

\* mandatory parameter

### Response

In response to a GetCapabilities request, the DGCS WMS server produces an XML document. This document contains the WMS server's service metadata; describes all the operations it supports; and provides information about available maps. The client application has to parse the XML capabilities document to retrieve the necessary information used to request a map. The Document Object Model (DOM) is a widely-used and efficient XML parser, which represents the XML document as a tree of nodes that can easily be edited with its standard interfaces. The response XML contains:

- WMS Service details (e.g. Name, Title, and URL)
- Contact Information (e.g. Person, Organization, Address, Telephone, Fax and Email)
- WMS Capabilities (e.g. GetCapabilities, GetMap, GetFeatureInfo and GetLegendGraphic) along with respective formats and URLs.
- Layers like Imagery and ImageryFootprint and Bounding Box details.

```xml
  <<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
  !!DOCTYPE WMT_MS_Capabilities (View Source for full doctype...)>
- <WMT_MS_Capabilities updateSequence="102" version="1.1.1">
- <<Service>
        <Name>OGC:WMS</Name>
        <Title>DigitalGlobe Web Map Service</Title>
        <Abstract />
-       <KeywordList>
            <Keyword>WFS</Keyword>
  \         <Keyword>WMS</Keyword>
    </KeywordList>
  <OnlineResource
xlink:href="https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECT
ID>&"xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink" />
-     <ContactInformation>
-         <ContactPersonPrimary>
              <ContactPerson>Customer Service Department</ContactPerson>
              <ContactOrganization>DigitalGlobe Inc</ContactOrganization>
      </ContactPersonPrimary>
          <ContactPosition>Customer Service Department</ContactPosition>
    <ContactAddress>
            <AddressType>Work</AddressType>
            <Address />
            <City />
            <StateOrProvince />
            <PostCode />
            <Country />
      </ContactAddress>
          <ContactVoiceTelephone>800.496.1225</ContactVoiceTelephone>
          <ContactFacsimileTelephone>303.684.4562</ContactFacsimileTelephone>
  <ContactElectronicMailAddress>
 info@digitalglobe.com
    </ContactElectronicMailAddress>
    </ContactInformation>
        <Fees>NONE</Fees>
      <AccessConstraints>NONE</AccessConstraints>
  </Service>
<Capability>
  <Request>
    <GetCapabilities>
        <Format>application/vnd.ogc.wms_xml</Format>
  <DCPType>
    <HTTP>
      <Get>
        <OnlineResource
xlink:href=https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECTI
D>&SERVICE=WMS& xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink" />
                </Get>
      </HTTP>
</DCPType>
      </GetCapabilities>
  <GetMap>
    <Format>image/png</Format>
<LatLonBoundingBox maxy="90.0" maxx="180.0"
miny="-90.0" minx="-180.0" />
    <Layer cascaded="0" queryable="0" opaque="0" noSubsets="0">
      <Name>DigitalGlobe:Imagery</Name>
      <Title>DigitalGlobe:Imagery</Title>
       <Abstract>

Continued…
```

```
…Continued
      Dynamic Raster Layer with content based on Membership
        </Abstract>
        <KeywordList />
        <SRS>EPSG:4326</SRS>
        <LatLonBoundingBox maxy="90.0" maxx="180.0"
          miny="-90.0" minx="-180.0" />
        <ScaleHint max="1000000" min="1" />
    </Layer>
     <Layer cascaded="0" queryable="0" opaque="0" noSubsets="0">
        <Name>DigitalGlobe:ImageryFootprint</Name>
        <Title>DigitalGlobe:ImageryFootprint</Title>
        <Abstract>Vector Layer based on generic</Abstract>
        <KeywordList />
        <SRS>EPSG:4326</SRS>
        <LatLonBoundingBox maxy="90.0" maxx="180.0"
          miny="-90.0" minx="-180.0" />
        <ScaleHint max="1000000" min="1" />
    </Layer>
  </Layer>
  </Capability>
  </WMT_MS_Capabilities>
      <Format>image/jpeg</Format>
     <DCPType>
 <HTTP>
   <Get>
<OnlineResource
xlink:href="https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNE
CTID>&SERVICE=WMS&" xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink"
/>
<OnlineResource
xlink:href="https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNE
CTID>&SERVICE=WMS&" xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink"
/>
          </Get>
        </HTTP>
      </DCPType>
   </GetMap>
   <GetFeatureInfo>
      <Format>text/plain</Format>
      <Format>text/html</Format>
      <Format>application/vnd.ogc.gml</Format>
<DCPType>
  <HTTP>
        <Get>
         </Get>a
       </HTTP>
      </DCPType>
  </GetFeatureInfo>
  <DescribeLayer>
      <Format>application/vnd.ogc.wms_xml</Format>
<DCPType>
       <HTTP>
   <Get>
<OnlineResource
xlink:href="https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNE
CTID>&SERVICE=WMS&" xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink"
/>

Continued…
```

```
…Continued
            </Get>
</HTTP>
      </DCPType>
  </DescribeLayer>
  <GetLegendGraphic>
        <Format>image/png</Format>
        <Format>image/jpeg</Format>
<DCPType>
  <HTTP>
    <Get>
<OnlineResource
xlink:href="https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNEC
TID>&SERVICE=WMS&" xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink" />
        </Get>
        </HTTP>
      </DCPType>
  </GetLegendGraphic>
  </Request>
<Exception />
  <UserDefinedSymbolization RemoteWFS="1" UserStyle="1" UserLayer="1"
SupportSLD="1" />
  <Layer noSubsets="0" opaque="0" queryable="0">
    <Title>DigitalGlobe Web Map Service</Title>
  <Abstract />
```

## STEP-2

The client can request GetMap with the map server's capabilities information in order to get a map image. The DigitalGlobe WMS supports maps in these formats:

- PNG
- JPEG

Name: GetMap
Example URL:
https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECTID>&SERVICE=WMS&REQUEST=GetMap&version=1.1.1&SRS=EPSG:4326&BBOX=-73.8854994266476,40.6341176876158,-73.8606463081034,40.6501752378911&WIDTH=1102&HEIGHT=712&LAYERS=DigitalGlobe:Imagery&format=image/jpeg

Replace <ConnectID> with the ConnectID provided by DigitalGlobe. Parameters are not required to be in the same order as shown above.

## Understanding URL

The URL shown above contains a Base URL and vital parameters as explained in Section 3.5 on page 13. The key parameter for this request is "**request=GetMap**" which fetches the map image for the specified **BBOX** and responss with PNG, JPEG, or GeoTIFF data based on the **format** parameter value.

**TABLE 3.5  URL PARAMETERS FOR WMS GETMAP REQUEST**

| PARAMETER NAME | PARAMETER VALUE | DESCRIPTION |
|---|---|---|
| CONNECTID* | **<CONNECTID>,** provided by Digital Globe | The value for this parameter is a unique 32-digit alphanumeric value assigned by DigitalGlobe (Explained under CONNECTID in Section 3.5 on page 13). A valid CONNECTID is mandatory for every request. |
| SERVICE* | WMS | Refer to SERVICE under Section 3.5 on page 13 for details. |
| REQUEST* | GetMap | The value for this parameter should always be "**GetMap**" for step-2 |
| VERSION | 1.1.1 | Refer to VERSION under Section 3.5 on page 13 for details. |
| SRS* | EPSG:4326<br>EPSG:3395<br>EPSG:3857<br>AUTO:42004 | This parameter is optional. Default projection will be taken if the parameter is not used in URL. Refer to SRS under Section 3.5 on page 13 for details.<br>A complete list of supported values is found in the GetCapabilities response. |
| BBOX* | 73.8854994266476, 40.6341176876158, - 73.8606463081034, 40.6501752378911<br><br>(Above BBOX value represents an area from New York) | This defines the user's area of interest. This parameter should be provided in the GetMap request URL to define the geospatial region and get the corresponding map. Refer to BBOX under Section 3.5 on page 13 for details. |
| WIDTH and HEIGHT* | WIDTH=1102<br>HEIGHT=712 | Width and height specify the requested map image size. Width and Height values should be proportional to the BBOX value. |
| LAYERS* | DigitalGlobe:Imagery<br>(or)<br>DigitalGlobe:ImageryFootprint | Imagery outputs the raster data and ImageryFootprint gives vector data for the given BBOX. The user can always use both of these layers as list parameter values (comma-separated). The last value in the list will be stacked over the previous and therefore visible on the screen.<br>Refer to Section 3.9 on page 2513 for more information on various WMS layers. |
| FORMAT* | Image/jpeg<br>Image/png<br>(or)<br>Image/geotiff | Defines the output image's format. |

\* mandatory parameter

## Response

In response to a GetMap request, the DigitalGlobe WMS server returns a map in the requested format (i.e. PNG or JPEG). Based on the requested layer, the response produces the corresponding image. WMS supports two different layers, namely Imagery and ImageryFootprint. Imagery produces a raster image while ImageryFootprint outputs vector data. Figure 3.4 shows the Response Image (raster data) for the GetMap request URL shown above.

WEB MAP SERVICE – DEVELOPER GUIDE

**FIGURE 3.4 GETMAP RESPONSE MAP (AN AREA FROM NEW YORK)**

## STEP-3

The client can request **GetFeatureInfo** by specifying a point on the map to receive more geographic feature information. This is an optional operation supported for those layers that have the attribute queryable="1". The canonical use case for **GetFeatureInfo** is that a user sees the response of a Map request and chooses a point on that map for which to obtain more information.

The basic operation provides the ability for a client to specify which pixel is being asked about, which layer(s) should be investigated, and in what format the information should be returned. Because the WMS protocol is stateless, the **GetFeatureInfo** request indicates to the WMS what map the user is viewing by including most of the original GetMap request parameters (all but VERSION and REQUEST). From the spatial context information (BBOX, SRS, WIDTH, HEIGHT) in that **GetMap** request, along with the X,Y position the user chose, the WMS can (possibly) return additional information about that position.

---

Name: GetFeatureInfo
Example URL:
https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECTID>&SERVICE=WMS&REQUEST=GetFeatureInfo&version=1.1.1&SRS=EPSG:4326&BBOX=-73.8854994266476,40.6341176876158,-73.8606463081034,40.6501752378911&WIDTH=1102&HEIGHT=712&QUERY_LAYERS=DigitalGlobe:ImageryFootprint&X=200&Y=150&LAYERS=DigitalGlobe:ImageryFootprint

Replace <ConnectID> with the ConnectID provided by DigitalGlobe. Parameters are not required to be in the same order as shown above.

---

### Understanding URL

The URL shown above contains the Base URL and vital parameters as explained in Table 3.6. The key parameter for this request is "**request=GetFeatureInfo**", which fetches the feature information or metadata at a particular pixel location (identified with X and Y values) on the map for the specified **BBOX** and within valid **QUERY_LAYERS.** WMS server responds back in the form of a .txt file which contains the requested feature information like feature id, % of cloud cover, age of image in days, image acquisition date, source of image, and so on.

**TABLE 3.6  URL PARAMETERS FOR WMS GETFEATUREINFO REQUEST**

| PARAMETER NAME | PARAMETER VALUE | DESCRIPTION |
|---|---|---|
| CONNECTID* | <CONNECTID> provided by Digital Globe | The value for this parameter is a unique 32-digit alphanumeric value assigned by DigitalGlobe (explained under CONNECTID in Section 3.5 on page 13). A valid CONNECTID is mandatory for every request. |
| SERVICE* | WMS | Refer to SERVICE under Section 3.5 on page 13 for details. |
| REQUEST* | GetFeatureInfo | The value for this parameter should always be "**GetFeatureInfo**" for step-3. |
| VERSION | 1.1.1 | Refer to VERSION under Section 3.5 on page 13 for details. |
| SRS* | EPSG:4326 EPSG:3395 EPSG:3857 AUTO:42004 | This parameter is optional. Default projection (EPSG:4326) will be taken if the parameter is not used in URL. Refer to SRS in Section 3.5 on page 13 for details. A complete list of supported values is found in the GetCapabilities response. |
| BBOX* | 73.8854994266476, 40.6341176876158, - 73.8606463081034, 40.6501752378911  (Above BBOX value represents an area from New York) | This defines the user's area of interest. This parameter should be provided in the GetFeatureInfo request URL to define the geospatial region. Refer to BBOX in Section 3.5 on page 13 for details. |
| WIDTH and HEIGHT* | WIDTH=1102 HEIGHT=712 | Width and height specify the requested map image size. Width and Height values should be proportional to the BBOX value. |
| LAYERS* | DigitalGlobe:ImageryFootprint | ImageryFootprint gives vector data for the given BBOX. The user should always use this layer only to get the feature information or metadata. |
| QUERY_LAYERS* | DigitalGlobe:ImageryFootprint | ImageryFootprint gives vector data for the given BBOX. The user should always use this layer only to get the feature information or metadata.  Refer to Section 25 on page 25 for more information on various WMS layers. |
| X and Y* | X=200 and Y=150 | X and Y represent the coordinates of a particular pixel on that particular region (depends on BBOX and WIDTH and HEIGHT). |

* mandatory parameter

## Response

In response to a **GetFeatureInfo** request, the DigitalGlobe WMS server produces a text file with the details of a requested feature. The response for the GetFeatureInfo request URL shown above is:

```
Results for FeatureType 'ImageryFootprint':
------------------------------------------------------------------------------
featureId = b101db8114077461dc308cc8045834ba
geometry = [GEOMETRY (Polygon) with 5 points]
offNadirAngle = 4.1947756
sunElevation = 67.1707
cloudCover = 0.0
```

WEB MAP SERVICE – DEVELOPER GUIDE

```
sunAzimuth = 150.24382
source = WV02
sourceUnit = Strip
productType = Pan Sharpened Natural Color
groundSampleDistance = 0.5
groundSampleDistanceUnit = Meter
dataLayer = country_coverage
legacyDescription =
outputMosaic = false
colorBandOrder = RGB
assetName = FINISHED
assetType = PRODUCT_GEOMETRY
legacyId = 1030010005C7FE00
factoryOrderNumber = 052411716-20
acquisitionDate = 2010-05-20 16:04:36.352
perPixelX = 4.499999992E-6
perPixelY = -4.499999999E-6
crsFromPixels = EPSG:4326
url =
ageDays = 193
formattedDate = 2010-05-20
CE90Accuracy = 10.2 meters
RMSEAccuracy = 4.75 meters
ingestDate = 2010-10-15 03:36:00.349574
spatialAccuracy = 1:12,000
earliestAcquisitionDate =
latestAcquisitionDate =
```

## 3.8 Service Exceptions

The WMS server throws an exception when a client request is invalid or the requested data is not available. WMS server sends exception as an XML document with content that varies in different scenarios and operation failures. 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 WMS Web Service is "application/vnd.ogc.se_xml", which means "Service Exception XML". The error messages appear as <**ServiceException**> elements within the <**ServiceExceptionReport**> in Service Exception XML. Table 3.7 shows the WMS exception codes as defined by OGC and implemented by DigitalGlobe.

**TABLE 3.7  WMS EXCEPTION CODES AND THEIR MEANINGS**

| EXCEPTION CODE | MEANING |
|---|---|
| InvalidFormat | The request contains a format not offered by the service instance. |
| InvalidSRS | The request contains an SRS not offered by the service instance for one or more of the layers in the request. |
| LayerNotDefined | The request is for a layer not offered by the service instance. |
| StyleNotDefined | The request is for a layer in a style not offered by the service instance. |
| LayerNotQueriable | GetFeatureInfo request is applied to a layer which is not declared queryable. |
| CurrentUpdateSequence | Value of (optional) UpdateSequence parameter in GetCapabilities request is equal to current value of Capabilities XML update sequence number. |
| InvalidUpdateSequence | Value of (optional) UpdateSequence parameter in GetCapabilities request is greater than current values of Capabilities XML update sequence number. |

The following sample requests generate different service exceptions:

Example Request 1:
https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECTID>&SERVICE=WMS&REQUEST=GetMap&version=1.1.1&SRS=EPSG:4326&BBOX=-

WEB MAP SERVICE – DEVELOPER GUIDE

The above request contains the parameter "**format=image/bmp**", which is not supported by WMS for **GetMap** operation. This request throws the "**InvalidFormat**" exception shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE ServiceExceptionReport (View Source for full doctype...)>
<ServiceExceptionReport version="1.1.1">
  <ServiceException code="InvalidFormat">
There is no support for creating maps in image/bmp format
    </ServiceException>
</ServiceExceptionReport>
```

Example Request 2:
https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECTID>&SERVICE=WMS&REQUEST=GetMap&version=1.1.1&SRS=EPSG:40026&BBOX=-73.8854994266476,40.6341176876158,-73.8606463081034,40.6501752378911&WIDTH=1102&HEIGHT=712&LAYERS=DigitalGlobe:Imagery&format=image/jpeg

The above request contains the parameter "**SRS=EPSG:40026**", which is not supported by WMS. This request throws the "**InvalidSRS**" exception shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE ServiceExceptionReport (View Source for full doctype...)>
<ServiceExceptionReport version="1.1.1">
  <ServiceException code="InvalidSRS">
Error occurred decoding the espg code EPSG:40026 No code "EPSG:40026" from
authority "European Petroleum Survey Group" found for object of type
"IdentifiedObject".
    </ServiceException>
</ServiceExceptionReport>
```

Example Request 3:
https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECTID>&SERVICE=WMS&REQUEST=GetMap&version=1.1.1&SRS=EPSG:4326&BBOX=-73.8854994266476,40.6341176876158,-73.8606463081034,40.6501752378911&WIDTH=1102&HEIGHT=712&LAYERS=DigitalGlobe:CitySphereTileService&format=image/jpeg

The parameter "**LAYERS=DigitalGlobe:CitySphereTileService**" in the above request trying to get image from CitySphereTileService layer, which is not supported by WMS. This request throws the "**LayerNotDefined**" exception shown below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE ServiceExceptionReport (View Source for full doctype...)>
<ServiceExceptionReport version="1.1.1">
  <ServiceException code=" LayerNotDefined">
Could not find layer DigitalGlobe:CitySphereTileService
    </ServiceException>
</ServiceExceptionReport>
```

Example Request 4:
https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECTID>&SERVICE=WMS&REQUEST=GetFeatureInfo&version=1.1.1&SRS=EPSG:4326&BBOX=-

WEB MAP SERVICE – DEVELOPER GUIDE

73.8854994266476,40.6341176876158,-
73.8606463081034,40.6501752378911&WIDTH=1102&HEIGHT=712&X=200&Y=150&LAYERS=Di
gitalGlobe:ImageryFootprint

The above "GetFeatureInfo" request does not contain "QUERY_LAYERS" parameter which is mandatory in this operation. As a result, the request throws the "**No QUERY_LAYERS**" exception shown below:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE ServiceExceptionReport (View Source for full doctype...)>
<ServiceExceptionReport version="1.1.1">
   <ServiceException
code="org.vfny.geoserver.wms.requests.GetFeatureInfoKvpReader">
No QUERY_LAYERS has been requested, or no queriable layer in the request anyways
   </ServiceException>
</ServiceExceptionReport>
```

Example Request 5
https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECTID>&SERVICE=W
MS&REQUEST=GetFeatureInfo&version=1.1.1&SRS=EPSG:4326&BBOX=-
73.8854994266476,40.6341176876158,-
73.8606463081034,40.6501752378911&WIDTH=1102&HEIGHT=712&QUERY_LAYERS=DigitalGlo
be:Imagery&X=200&Y=150&LAYERS=DigitalGlobe:ImageryFootprint

The above "**GetFeatureInfo**" request contains incorrect value for "**QUERY_LAYERS**" parameter which must be "**ImageryFootprint**". As a result, the request throws the "**Internal error occurred**" exception shown below:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE ServiceExceptionReport (View Source for full doctype...)>
<ServiceExceptionReport version="1.1.1">
   <ServiceException code="Internal error occurred" />
</ServiceExceptionReport>
```

Example Request 6
https://services.digitalglobe.com/mapservice/wmsaccess?connectid=<CONNECTID>&SERVICE=W
MS&REQUEST=GetFeatureInfo&version=1.1.1&SRS=EPSG:4326&BBOX=-
73.8854994266476,40.6341176876158,-
73.8606463081034,40.6501752378911&WIDTH=1102&HEIGHT=712&QUERY_LAYERS=DigitalGlo
be:ImageryFootprint&LAYERS=DigitalGlobe:ImageryFootprint

The above "**GetFeatureInfo**" request does not contain the mandatory parameters X and Y. As a result, it throws the "**X and Y incorrectly specified**" exception shown below:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE ServiceExceptionReport (View Source for full doctype...)>
<ServiceExceptionReport version="1.1.1">
   <ServiceException>
X and Y incorrectly specified
   </ServiceException>
</ServiceExceptionReport>
```

## 3.9  WMS Layers

**TABLE 3.8  WMS LAYERS AND DESCRIPTION**

| OGC LAYER(S) | DESCRIPTION |
| --- | --- |
| Imagery | Imagery for all data layers available to the account, with the default |

WEB MAP SERVICE – DEVELOPER GUIDE

| OGC LAYER(S) | DESCRIPTION |
|---|---|
| | display for a location determined by stacking profile. |
| ImageryFootprint | Vector representation of the raster coverage available in the Imagery Layer. |
| CrisisEvent | Imagery from the FirstLook Product, with the most-recent image displayed by default. |
| CrisisEventFootprint | Vector representation of the raster coverage available in the CrisisEvent Layer. |
| LegacyCountryCoverage | Imagery from the DGWS 1.0 system, formerly known as Layerstack 49. Stacking profiles do not apply. |
| LegacyCountryCoverageFootprint | Vector representation of the raster coverage available in the LegacyCountryCoverageLayer. |
| LegacyGlobalBaseMap | Imagery from the DGWS 1.0 system, formerly known as Layerstack 58. Stacking profiles do not apply. |
| LegacyGlobalBasemapFootprint | Vector representation of the raster coverage available in the LegacyGlobalBaseMap Layer. |
| LegacyOilAndGasCoverage | Imagery from the DGWS 1.0 system, formerly known as Layerstack 48. Stacking profiles do not apply. |
| LegacyOilAndGasCoverageFootprint | Vector representation of the raster coverage available in the LegacyOilAndGasCoverage Layer. |

# 3.10 API Reference

The API reference provides a list of all possible request parameters for every WMS operation as well as detailed information about corresponding response.

The client should provide the respective information in a Key-Value Pair (KVP) format for every WMS 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=WMS".

## 3.10.1 GET CAPABILITIES

Please refer to Section 3.4 Basic Service Elements on page 11 for request parameters and response XML.

## 3.10.2 GET MAP

The following table shows all possible request parameters for GetMap operation of WMS server.

TABLE 3.9  PARAMETERS OF A WMS GETMAP REQUEST

| NAME | VALUE | DESCRIPTION |
|---|---|---|
| SERVICE* | WMS | Web Map Service |
| VERSION | 1.1.1 | Request version |
| REQUEST* | GetMap | Request name |
| LAYERS* | One or more of the available data layers; each is prefixed with "DigitalGlobe:" Supported Layers by WMS service are: **DigitalGlobe: Imagery** **DigitalGlobe: ImageryFootprint** **crisis_event** (FirstLook - Natural and | The layers available from the Online Catalogs. If more than one layer is requested they should be in a comma-separated list. |

| NAME | VALUE | DESCRIPTION |
|---|---|---|
| | 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) | |
| STYLES* | A style definition for each requested layer; valid styles are defined in the GetCapabilities response: Example value: dg_ras | If multiple layers are requested, the styles parameter is a comma-separated list. If the default style is desired, the "STYLES parameter must still be included, but the value(s) may be left empty. |
| SRS* | One of: EPSG:4326 EPSG:3395 EPSG:3857 AUTO:42004 | Only one SRS is supported. If SRS is not specified, EPSG:4326 is the default value in which the map will be returned. 4326 provides tiles where latitude and longitude are treated as X/Y values. 3857 provides tiles in the spherical mercator projection. A complete list of supported values is found in the GetCapabilities response. |
| BBOX* | Bounding box defined as: minx,miny,maxx,maxy | Bounding box of the requested map area, in coordinates of the requested SRS. |
| WIDTH* | Integer number of pixels to be returned in the X coordinate | NOTE: If the width/height aspect is different than the BBOX ratio, a distorted map may result. |
| HEIGHT* | Integer number of pixels to be returned in the Y coordinate | NOTE: If the width/height aspect is different than the BBOX ratio, a distorted map may result. |
| FORMAT* | MIME format in which the data is to be returned: image/png image/jpeg image/geotiff | The format in which to return the map image. |
| TRANSPARENT | Binary field indicating whether the returned image should have a transparent background: FALSE: Not transparent TRUE: Transparent | Default =TRUE if not supplied |
| BGCOLOR | Background color in hexadecimal format. | Default =0Xffffff if not supplied |

| NAME | VALUE | DESCRIPTION |
|------|-------|-------------|
| | Example: 0xFFFFFF | |
| CONNECTID* | Character String | User's unique identifier supplied by DigitalGlobe; required to access the DG Web Services. |
| USERNAME | Username in the form of a valid email address | Username for commercial users; This parameter may be mandatory based on account type. |
| PASSWORD | Password assigned by DigitalGlobe for this username | Password for commercial users; This parameter may be mandatory based on account type. |
| COVERAGE_CQL_FILTER | Filters the images included in the returned map | See Section 4 Common Query Language on page 31 of this document for description of CQL filtering |
| EXCEPTIONS | MIME type for exceptions:<br><br>application/vnd.ogc.se_xml | Format in which exceptions will be reported. If not specified, default is XML. |
| FEATUREPROFILE | The current profiles include:<br>Accuracy_Profile<br>Aerial_CIR_Profile<br>Cloud_Cover_Profile<br>Consumer_Profile<br>Currency_Profile<br>Currency_RGB_Profile<br>Default_Profile<br>True_Currency_Profile | Profiles describe the stacking rules on an account. The user can choose a profile in a request by specifying the featureProfile. The most current available profiles can be found in the GetCapabilities response. |

* mandatory parameter

### 3.10.3 GETFEATUREINFO

The following table shows all possible request parameters for GetFeatureInfo operation of WMS server.

**TABLE 3.10  PARAMETERS OF A WMS GETFEATUREINFO REQUEST**

| NAME | VALUE | DESCRIPTION |
|------|-------|-------------|
| SERVICE* | WMS | Web Map Service |
| VERSION* | 1.1.1 | Request version |
| REQUEST* | GetFeatureInfo | Request name |
| BBOX* | Bounding box defined as:<br>minx,miny,maxx,maxy<br>This parameter value from the GetMap request should be repeated here. | Bounding box of the requested map area, in coordinates of the requested SRS. |
| WIDTH* | Integer number of pixels to be returned in the X coordinate<br>This parameter value from the GetMap request should be repeated here. | NOTE: If the width/height aspect is different than the BBOX ratio, a distorted map may result |

| NAME | VALUE | DESCRIPTION |
|---|---|---|
| HEIGHT* | Integer number of pixels to be returned in the Y coordinate.<br>This parameter value from the GetMap request should be repeated here. | NOTE: If the width/height aspect is different than the BBOX ratio, a distorted map may result |
| QUERY_LAYERS* | One or more of the available WMS layers, for example:<br>DigitalGlobe: Imagery | The layer(s) for which feature info is being requested; if multiple layers they are in a comma-separated list.<br>Refer to Section 3.7 Integration Procedure on page 16 for more information on various WMS layers. |
| INFO_FORMAT | MIME type of returned feature info:<br>application/vnd.ogc.gml | Mime format in which to return the feature info; XML is the supported value. |
| FEATURE_COUNT | Its value is a positive integer greater than zero. | This states the maximum number of features for which feature information should be returned.<br>If this parameter isn't specified, the default value is 1. |
| X* | Integer; value from 0 to specified "WIDTH" value | X coordinate (column) in pixels of feature (measured from upper left corner=0) |
| Y* | Integer; value from 0 to specified "HEIGHT" value | Y coordinate (row) in pixels of feature (measured from upper left corner=0) |
| EXCEPTIONS | MIME type for exception reporting: application/vnd.ogc.se_xml | Mime format in which any exception should be returned; if not specified, the default is XML. |
| CONNECTID* | Character String | User's unique identifier supplied by DigitalGlobe; required to access the DG Web Services |
| USERNAME | Username in the form of a valid email address. | Username for commercial users. This parameter may be mandatory based on account type. |
| PASSWORD | The password assigned by DigitalGlobe for this username. | Password for commercial users. This parameter may be mandatory based on account type. |

* mandatory parameter

**TABLE 3.11 GETFEATUREINFO RESPONSE**

| ELEMENT | DATATYPE | VALUE | DESCRIPTION |
|---|---|---|---|
| featureId | String | Example:<br>b101db8114077461dc308cc8045834ba | The Feature ID for the map feature at the given (X,Y) coordinate |
| acquisitionDate | DateTime | Example:<br>2009-05-27 01:20:43.63599 | The GMT of acquisition of the image used to make the product described by the feature. |
| formattedDate | Date | Example:<br>2010-05-20<br>(YYYY-MM-DD) | The date of production or the date the image was reproduced. |
| cloudCover | Float | 0.0-1.0 | The percentage of cloud cover described by the Feature.<br>Example: To represent 20% cloud cover, specify |

| ELEMENT | DATATYPE | VALUE | DESCRIPTION |
|---------|----------|-------|-------------|
| | | | 0.2. |
| offNadirAngle | Float | 0.0-90.0 | The off-nadir angle of the sensor used to create the product described by the feature. |
| sunElevation | Float | 0.0-90.0 | Specifies the desired sun elevation angle. |
| sunAzimuth | Float | +/- 180.00 | The azimuth angle of the sun for the image used to make the product described by the Feature |
| ageDays | Integer | Example: 29 | The number of days between the acquisitionDate and today. |
| data_layer | String | Country_coverage | Represents the layer to which this particular image belongs.<br><br>Refer to Section 3.7 Integration Procedure on page 16 for more information on various WMS layers. |
| source | String | WV02 | Represents the source of the image. This value would be one of the following:<br>QB-QuickBird<br>WV01-WorldView1<br>WV02-WorldView2<br>Aerial- Aerial Imagery |
| outputMosaic | Boolean | TRUE/ FALSE | Value will be TRUE if the output is a mosaic of more than one source image. |

# 4  Common Query Language

Common Query Language (CQL) is a formal language for representing queries to Information Retrieval systems such as web indexes, bibliographic catalogues and museum information collections.

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

## 4.1  CQL Filter Principle

The following examples explain CQL Filter principles and their usage.
***Example 1:***

To request a 1000 x 1000 pixel JPG image within the specified bounding box, created from a specific featureid from a layer, the request would be as follows. Note the single quotes around the featureid value.

Example Request 1:
https://services.digitalglobe.com/mapservice/wmsaccess?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&LAYERS=DigitalGlobe:Imagery&STYLES=&FORMAT=image/jpeg&BGCOLOR=0xFFFFFF&TRANSPARENT=TRUE&SRS=EPSG:4326&BBOX=-115.154891318604,36.0873836159735,-115.149610131305,36.0928787247827&WIDTH=790&HEIGHT=822&connectid=<CONNECTID>&featureProfile=Default_Profile&COVERAGE_CQL_FILTER=featureId=%27f54fc0c06fa7e94c6c17b8ab30f333ce%27

Username and Password parameters may not be applicable depending on your account type. Replace <ConnectID> with the ConnectID provided by DigitalGlobe. Parameters are not required to be in the same order as shown above.

***Example 2:***

To request all layers that have existed for fewer than 14 days, the request would be as follows:

Example Request 2:
https://services.digitalglobe.com/mapservice/wmsaccess?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&LAYERS=DigitalGlobe:Imagery&STYLES=&FORMAT=image/jpeg&BGCOLOR=0xFFFFFF&TRANSPARENT=TRUE&SRS=EPSG:4326&BBOX=-115.154891318604,36.0873836159735,-115.149610131305,36.0928787247827&WIDTH=790&HEIGHT=822&connectid=<CONNECTID>&featureProfile=Default_Profile&Coverage_CQL_Filter=age_days<14

Replace <ConnectID> with the ConnectID provided by DigitalGlobe. Parameters are not required to be in the same order as shown above.

Since the parameter Coverage_CQL_Filter is set to age_days<14, the response will include all layers that have existed for fewer than 14 days.
***Example 3:***

To request all layers that have a cloud cover of less than 5%, the request would be as follows:

Example Request 3:
https://services.digitalglobe.com/mapservice/wmsaccess?REQUEST=GetMap&SERVICE=WMS&VERSION=1.1.1&LAYERS=DigitalGlobe:Imagery&STYLES=&FORMAT=image/jpeg&BGCOLOR=0xFFFFFF&TRANSPARENT=TRUE&SRS=EPSG:4326&BBOX=-

WEB MAP SERVICE – DEVELOPER GUIDE

115.154891318604,36.0873836159735,-
115.149610131305,36.0928787247827&WIDTH=790&HEIGHT=822&connectid=<CONNECTID>
&featureProfile=Default_Profile&Coverage_CQL_Filter=cloud_cover<.05

Replace <ConnectID> with the ConnectID provided by DigitalGlobe. Parameters are not required to be in the same order as shown above.

Since the parameter Coverage_CQL_Filter is set to cloud_cover<.05, the response will include all layers that have less than 5% cloud cover.

***Example 4:***

To request all layers that have the acquisition_date greater than '5/1/2009', the request would look like this:

Example Request 4:
https://services.digitalglobe.com/mapservice/wmsaccess?REQUEST=GetMap&SERVICE=WMS&VE
RSION=1.1.1&LAYERS=DigitalGlobe:Imagery&STYLES=&FORMAT=image/jpeg&BGCOLOR=0xFF
FFFF&TRANSPARENT=TRUE&SRS=EPSG:4326&BBOX=-
115.154891318604,36.0873836159735,-
115.149610131305,36.0928787247827&WIDTH=790&HEIGHT=822&connectid=<CONNECTID>&fe
atureProfile=Default_Profile&COVERAGE_CQL_FILTER=acquisition_date>%275/1/2009%27

Replace <ConnectID> with the ConnectID provided by DigitalGlobe. Parameters are not required to be in the same order as shown above.

Since the parameter Coverage_CQL_Filter is set to acquisition_date>%275/1/2009%27, the response will show all layers that have the acquisition_date greater than '5/1/2009'.

## 4.2  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.

The following table shows the possible CQL Filter parameters.

**TABLE 4.1  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 |

| ELEMENT | DATATYPE | EXAMPLE VALUE | DESCRIPTION |
|---|---|---|---|
| | | | 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. |
| 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. |

| ELEMENT | DATATYPE | EXAMPLE VALUE | DESCRIPTION |
|---|---|---|---|
| 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 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. An 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 |

WEB MAP SERVICE – DEVELOPER GUIDE

| ELEMENT | DATATYPE | EXAMPLE VALUE | DESCRIPTION |
|---------|----------|---------------|-------------|
| | | | 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 time (GMT) 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. |

# Glossary

### AOI

Area of Interest. Describes 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.

### 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

Overlapping raster datasets can be blended along the seamline by a specified width. Seamlines are created with the Seamline extension tools, and are stored as shapefiles within the image service definition.

### 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.

WEB MAP SERVICE – DEVELOPER GUIDE

**WCS**

Web Coverage Service.

**WebCGM**

Web Computer Graphics Metafile.

**WFS**

Web Feature Service.

**WMS**

Web Map Service.

**WMTS**

Web Map Tile Service.

# Index