# Table of Contents

List of Figures ................................................................................................................................. 2  
List of Tables ................................................................................................................................. 3  
1 Introduction .................................................................................................................................. 4  
  1.1 About this Document .............................................................................................................. 4  
  1.2 Targeted Audience ................................................................................................................. 4  
  1.3 What is WMTS? ...................................................................................................................... 4  
  1.4 References ............................................................................................................................. 4  
2 Open Geospatial Consortium (OGC) ............................................................................................... 5  
  2.1 About OGC .............................................................................................................................. 5  
  2.2 The OGC Process ................................................................................................................... 5  
  2.3 OGC Standards and Specification ......................................................................................... 5  
  2.4 OGC Standards ..................................................................................................................... 6  
  2.5 Abstract Specification ............................................................................................................ 6  
  2.6 OGC Reference Model (ORM) ............................................................................................... 6  
3 Geography Markup Language (GML) ............................................................................................. 7  
  3.1 Introduction to GML ................................................................................................................ 7  
  3.2 Overview of GML Schema ..................................................................................................... 7  
  3.3 GML Schema Features .......................................................................................................... 8  
      3.3.1 AbstractFeatureType ....................................................................................................... 8  
      3.3.2 AbstractFeature ............................................................................................................. 8  
  3.4 Overview ............................................................................................................................... 8  
      3.4.1 Dictionary Schema ......................................................................................................... 9  
4 Web Map Tile Service ....................................................................................................................... 14  
  4.1 Introduction ............................................................................................................................ 14  
  4.2 Advantages of WMTS ............................................................................................................ 14  
  4.3 WMTS Client-Server Architecture ........................................................................................ 14  
  4.4 WMTS Service Details .......................................................................................................... 15  
5 Basic Service Elements .................................................................................................................. 16  
  5.1.1 HTTP Request .................................................................................................................. 16  
  5.1.2 HTTP response ................................................................................................................. 17  
  5.1.3 Request Parameters ......................................................................................................... 18  
  5.1.4 Request Parameter Rules ................................................................................................. 20  
  5.2 Integration Procedure ............................................................................................................ 20  
  5.3 Service Exceptions ................................................................................................................. 23  
  5.4 WMTS Layers ....................................................................................................................... 24  
  5.5 API Reference ....................................................................................................................... 24  
6 Tile Cache .................................................................................................................................... 26  
  6.1 Introduction ........................................................................................................................... 26  
  6.2 Building a Tile Cache from DigitalGlobe WMTS .................................................................... 26  
  6.3 Using Tile Cache .................................................................................................................... 27  
  6.4 Updating Tile Cache ............................................................................................................... 27  
  6.5 World-File Generation .......................................................................................................... 28  

Glossary ........................................................................................................................................... 29

Index ............................................................................................................................................... 31
List of Figures

Figure 4.1 A Typical Structure of a DGCS-WMTS Application .......................................................... 14
Figure 4.2 Sample WMTS Client-Server Application ...................................................................... 14
Figure 5.1 Sample HTTP Post Request/Response ........................................................................ 17
Figure 5.2 Representation of Bounding Box (BBOX) ................................................................... 20
Figure 6.1 Tile Cache Response Image .......................................................................................... 27
List of Tables

Table 2.1 OGC Document Types ........................................................................................................ 5
Table 5.1 A General Get Request ..................................................................................................... 16
Table 5.2 Reserved Characters in HTTP GetQuery ........................................................................... 16
Table 5.3 Values for outputFormat Attribute ................................................................................... 18
Table 5.4 WMTS Layers .................................................................................................................... 24
Table 5.5 Understanding URL Parameters for WMTS GetCapabilities Request ......................... 24
Table 5.6 GetTile Request Parameters .............................................................................................. 25
1 Introduction

1.1 About this Document

This document covers the concepts of Web Map Tile Service (WMTS), Open Geospatial Consortium (OGC) standards for WMTS, capabilities of WMTS and ways to integrate DigitalGlobe Cloud Services (DGCS)-WMTS 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 WMTS can read about the DGCS-WMTS framework, capabilities, integration procedures and development best-practices to design methods for creating innovative world-class GIS applications.

1.3 What is WMTS?

The OGC WMTS Implementation Standard provides an interface to server digital maps using predefined image tiles. The WMTS standard complements the existing Web Map Service standard of the OGC. Instead of overlaying an arbitrary number of the map layers offered by the server, over an arbitrary geographic bound, with an arbitrary background color at an arbitrary scale, in any supported coordinate reference system the client can perform image overlays themselves. This limits clients to request map images that are not at exactly the right position thereby forcing the clients to mosaic the tiles obtained from the server and clip the set of tiles into a final image.

Tile

The tile resource is generally a rectangular image containing cartographic data. Alternatively, this resource might be a non-image representation of the tile such as a description of the tile or a link to the actual image. For example, the tile resource could be a KML document used in a super-overlay, or a tile metadata document. When returning an image tile, a full single tile SHALL always be returned. Also, the background pixels of a tile SHOULD be transparent when possible so that the client can overlay the tiles on top of other map data (possibly other tiles).

1.4 References

- http://www.opengeospatial.org/standards
- http://www.wikipedia.org/
2 Open Geospatial Consortium (OGC)

2.1 About OGC

The OGC is an international voluntary consensus standards organization, 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, OGF, the Open GRASS Foundation, 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 GIS world. The steps described below are the process followed by OGC:

- Identifying Problem
- Crafting Solution
- Evaluating Proposed Solution, and
- Implementing 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 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 OGC Document Types lists the documents currently available on the OGC website.

<table>
<thead>
<tr>
<th>OGC DOCUMENT TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract Specification</td>
<td>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.</td>
</tr>
<tr>
<td>Best Practices</td>
<td>A document containing discussion related 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.</td>
</tr>
<tr>
<td>Discussion Papers</td>
<td>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.</td>
</tr>
<tr>
<td>White Papers</td>
<td>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.</td>
</tr>
</tbody>
</table>
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 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.

2.5 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 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 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 very general and includes not only conventional “vector” or discrete objects, but also coverage’s 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/>
  </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 Web Map Tile Service (WMTS) 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 WMTS you use GML for this purpose. When you wish 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 WMTS.

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 feature 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, BoundingBoxType, 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:

```xml
<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, declared as follows:

```xml
<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>
    </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 that are members of a dictionary. The element `gml:dictionaryEntry` is declared as follows:

```xml
<element name="dictionaryEntry" type="gml:DictionaryEntryType"/>
<complexType name="DictionaryEntryType">
  <complexContent>
    <extension base="gml:AbstractMemberType">
      <sequence minOccurs="0">
        <element ref="gml:Definition"/>
      </sequence>
    </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` may be substituted 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. Thus, it 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 and 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:
A simple dictionary of rock types using components from gmlBase

Rock Types

Granite
A igneous rock normally composed of quartz, two feldspars and optional mica

Sandstone
A detrital sedimentary rock normally composed of siliceous grains

GML Dictionary

conceptual schema for data required by one or more applications

application schema

application schema written in XML Schema in accordance with the rules specified in ISO 19136

semantic relationship between two or more classifiers that specifies connections among their instances

Continued...
Continued

<gmldictionaryEntry>
  <gml:Definition gml:id="term4.4">
    <gml:description>name-value pair contained in an element</gml:description>
  </gml:Definition>
</gmldictionaryEntry>

<!--... -->
</gmldictionary>
4 Web Map Tile Service

4.1 Introduction

The DigitalGlobe Web Map Tile 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 WMTS specification can interact with the DGCS WMTS server. Web-based client-server architecture is a typical example of the structure of a Web Map Tile Service application, as illustrated in Figure 4.1.

FIGURE 4.1 A TYPICAL STRUCTURE OF A DGCS-WMTS APPLICATION

In the DGCS - WMTS scenario, the client application requests desired information from the web map tile service server. The WMTS server retrieves from the database the appropriate information including capabilities, Tile and FeatureInfo, and responds to the request with relevant information.

The goal of providing a WMTS enabled service is to be performance-oriented and scalable. Therefore, servers must be able to return tiles quickly. A good way to achieve this is to use locally-stored pre-rendered tiles that don’t require any image manipulation or geo-processing.

The purpose of a WMTS service is to serve maps divided in individual tiles.

FIGURE 4.2 SAMPLE WMTS CLIENT-SERVER APPLICATION

4.2 Advantages of WMTS

The DigitalGlobe WMTS provides raster imagery data at multiple resolutions in predefined imagery tiles in PNG or JPEG formats, which are more scalable than WMS. The Web Map Tile Service is similar to Web Map Service, but it enables better server performance in applications that involve several simultaneous requests. To improve performance; WMTS returns small pre-generated images or reuses identical previous requests that follow a discrete set of tile matrices, rather than creating a new image for each request. This service is optimized by sending rapid image tile delivery information from the cache for display on portals or applications where server response time is of primary concern.

4.3 WMTS Client-Server Architecture

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

The WMTS server accepts requests from the WMTS client and viewer client in the form of HTTP URL strings, and returns results encoded as XML, GIF, GML, and so on.
4.4 WMTS Service Details

The DigitalGlobe Web Map Tile Service (WMTS) provides raster imagery data at multiple resolutions in various formats via predefined imagery tiles. The DigitalGlobe WMTS supports the following operations:

**GetCapabilities**

The GetCapabilities request is used to obtain information about the available map tile types and supported operations.

**GetTile**

The GetTile request is used to obtain an actual imagery tile. Tiles are available for a subset of the layers provided by the online catalogs.
5  Basic Service Elements

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

5.1.1  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 Cloud 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 5.1 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-WMTS specification in order to construct a valid request URL. Please refer to Table 5.2 for reserved characters as per HTTP rules.

<table>
<thead>
<tr>
<th>URL COMPONENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://host%5B:port%5D/path?%7Bname=value%7D&amp;">http://host[:port]/path?{name=value}&amp;</a></td>
<td>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.</td>
</tr>
<tr>
<td>name=value&amp;</td>
<td>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 in Table 5.2.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHARACTER</th>
<th>RESERVED USAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Separator indicating start of query string.</td>
</tr>
<tr>
<td>&amp;</td>
<td>Separator between parameters in query string.</td>
</tr>
<tr>
<td>=</td>
<td>Separator between name and value of parameter</td>
</tr>
<tr>
<td>/</td>
<td>Separator between MIME type and subtype in format parameter value.</td>
</tr>
</tbody>
</table>
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-WMTS do not require additional parameters to be appended to the URL in order to construct a valid target for the Operation request. Figure 5.1 shows a sample of a HTTP Post request.

FIGURE 5.1 SAMPLE HTTP POST REQUEST/RESPONSE

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 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 the secure communication.

5.1.2 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 4.7.
Output formats

The optional `outputFormat` attribute specifies the format of the response to a Cloud 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. Table 5.3 summarizes the possible values for the `outputFormat` attribute.

<table>
<thead>
<tr>
<th>OUTPUTFORMAT VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>GML2</td>
<td>This value is kept for backward compatibility and indicates that an XML instance document must be generated that validates against a GML2 application schema.</td>
</tr>
<tr>
<td>text/xml; subtype=gml/2.1.2</td>
<td>Same as GML2</td>
</tr>
<tr>
<td>text/xml; subtype=gml/3.1.1</td>
<td>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.</td>
</tr>
</tbody>
</table>

5.1.3 REQUEST PARAMETERS

As per the specification standards of WMTS, 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 WMTS URL.

Base URL

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

Base URL: https://services.digitalglobe.com/earthservice/wmtsaccess

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

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: x x x x x x - x x x x - x x x x - x x x x x x x x x x

x → alpha numeric number

SERVICE

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

Example: service=WMTS

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 WMTS implemented for the OGC specification is 1.0.0.

The version number appears in following two places:
- In response XML of GetCapabilities request describing WMTS service and
- In the parameter list of client requests to the WMTS service.
In response to a GetCapabilities request containing a version number, an WMTS server responds with output that conforms to that version of the specification, or negotiate 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 OGC specification for WMTS on negotiation rules.

Example: version=1.0.0 (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 Section 4.1 - Introduction of this document on different operations supported by DigitalGlobe WMTS 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 WMTS server, the server throws a Service Exception (with code "InvalidFormat"). Please refer Table 4.3. Values for output Format attribute for different types of formats supported by DigitalGlobe WMTS for different response types.

Example: format=image/jpeg

EXCEPTIONS

The EXCEPTIONS parameter in a request indicates the format in which the Client wishes to be notified of Service Exceptions. Individual error messages appear as <ServiceException> elements within the <ServiceExceptionReport> in Service Exception XML. Please refer to Section 4.7 for more details on Service Exceptions.

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 WMTS implementation is defined with EPSG and AUTO namespaces. If a request contains an SRS not offered by a DG WMTS 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, maximum latitude.

Example: SRS=EPSG: 4326

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, as in following figure: 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.

![Diagram: Representation of Bounding Box (BBOX)](image)

Example: BBOX=88.1035780267704,40.4568762655891, -8.0928025063267,40.4638383078358

5.1.4 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 WMTS 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.

5.2 Integration Procedure

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

STEP-1

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

Example Request:

https://services.digitalglobe.com/earthserv/wmtsaccess?SERVICE=WMTS&REQUEST=GetCapabilities&VERSION=1.0.0&connectid=<ConnectID>

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

The above given URL contains Base URL, and few parameters as explained in Section 5 on page 16. The key parameter for this request is "request=GetCapabilities" which fetches the capabilities of Web Map Tile Service and response back in form of XML data.

Refer to Request ParametersRequest Parameters on page 18 for details on the URL parameters for WMTS GetCapabilities request.

Response

In response to a GetCapabilities request, the DGCS WMTS server produces an eXtensible Markup Language (XML) document containing the WMTS server’s service metadata, describing all the operations it supports, and providing information about the available map types. 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:
- WMTS Service details like Name, Title, URL
- Contact Information Person, Organization, Address, Telephone, Fax and Email

WMTS Capabilities like GetCapabilities and GetTile, with respective formats and URLs.
STEP 2

The client can request a GetTile operation from the WMTS server in order to get the actual imagery tile of a particular tile matrix set in a predefined format. This operation has some parameters in common with WMS GetMap but it has been deliberately simplified.

Example Request:

https://services.digitalglobe.com/earthservice/wmtsaccess?SERVICE=WMTS&amp;VERSION=1.0.0&amp;REQUEST=GetTile&amp;CONNECTID=&lt;CONNECTID&gt;&amp;LAYER=DigitalGlobe:ImageryTileService&amp;FORMAT=image/jpeg&amp;TileRow=18200&amp;TileCol=27207&amp;TileMatrixSet=EPSG:4326&amp;TileMatrix=EPSG:4326:16&amp;featureProfile=Consumer_Profile&amp;inclusionOfSimpleMetadata=false

- Replace &lt;ConnectID&gt; with your ConnectID provided by DigitalGlobe.
- Parameters are not required to be in the same order as shown above.

Understanding URL

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=WMTS”. Refer to Table 5.6 GetTile Request Parameters for details on the URL parameters for WMTS GetTile request.

Response

The response to a GetTile Request is a map tile, in the requested format and projection. All DGCS-WMTS tiles are 256x256 pixels; no other tile sizes are currently supported.

5.3 Service Exceptions

In the event that a Web Map Tile 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 &lt;ExceptionReport&gt; element will contain one or more processing exceptions specified using the &lt;Exception&gt; element.

Individual &lt;Exception&gt; element contains exceptionCode whose values specifies the actual exception that occurred.

The following is an example of an exception report. This exception indicates that a parameter value is missing and a parameter value in invalid in the request URL.
5.4 WMTS Layers

<table>
<thead>
<tr>
<th>OGC LAYER(S)</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageryTileService</td>
<td>Tiled imagery for all data layers available to the account, with the</td>
</tr>
<tr>
<td></td>
<td>default display for a location determined by stacking profile.</td>
</tr>
<tr>
<td>CitySphereTileService</td>
<td>Tiled imagery from the Global Basemap City Program. Stacking profiles do not</td>
</tr>
<tr>
<td></td>
<td>apply.</td>
</tr>
<tr>
<td>CountryCoverageTileService</td>
<td>Tiled imagery from the Global Basemap Country Program. Stacking profiles do</td>
</tr>
<tr>
<td></td>
<td>not apply.</td>
</tr>
<tr>
<td>LegacyCountryCoverageTileService</td>
<td>Tiled imagery from the DGWS 1.0 system, formerly known as Layerstack 49.</td>
</tr>
<tr>
<td></td>
<td>Stacking profiles do not apply.</td>
</tr>
<tr>
<td>LegacyGlobalBaseMapTileService</td>
<td>Tiled imagery from the DGWS 1.0 system, formerly known as Layerstack 58.</td>
</tr>
<tr>
<td></td>
<td>Stacking profiles do not apply.</td>
</tr>
<tr>
<td>LegacyOilAndGasCoverageTileService</td>
<td>Tiled imagery from the DGWS 1.0 system, formerly known as Layerstack 48.</td>
</tr>
<tr>
<td></td>
<td>Stacking profiles do not apply.</td>
</tr>
</tbody>
</table>

5.5 API Reference

API reference gives the list of all possible request parameters for every WMTS operation and 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 “ey=value”; for example, “service=WMTS”.

GetCapabilities

Following table shows all possible request parameters for GetCapabilities operation of WMTS server.

<table>
<thead>
<tr>
<th>PARAMETER NAME</th>
<th>PARAMETER VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECTID*</td>
<td>&lt;ConnectID&gt;</td>
<td>Value for this parameter is an unique 32 digit alphanumeric value given by</td>
</tr>
<tr>
<td></td>
<td>provided by Digital Globe</td>
<td>DigitalGlobe (Explained in CONNECTID of Section 4.5.3 - Request Parameters).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid CONNECTID is mandatory for every request.</td>
</tr>
<tr>
<td>SERVICE*</td>
<td>WMTS</td>
<td>Explained in SERVICE of Section 4.5.3 - Request Parameters</td>
</tr>
<tr>
<td>REQUEST*</td>
<td>GetCapabilities</td>
<td>The value for this parameter should always be “GetCapabilities” for step-1.</td>
</tr>
<tr>
<td>VERSION</td>
<td>1.1.0</td>
<td>Refer VERSION in Section 4.5.3 - Request Parameters</td>
</tr>
</tbody>
</table>
### GetTile

#### TABLE 5.6 GETTILE REQUEST PARAMETERS

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>service*</td>
<td>WMTS</td>
<td>Web Map Tile Service</td>
</tr>
<tr>
<td>version*</td>
<td>1.0.0</td>
<td>Request version</td>
</tr>
<tr>
<td>request*</td>
<td>GetTile</td>
<td>Request name</td>
</tr>
<tr>
<td>layer*</td>
<td>One or more of the available layers, such as: DigitalGlobe:ImageryTileService</td>
<td>The layers available from the Online Catalogs; if more than one layer is requested they are in a comma-separated list. Refer to Section 5.4 on page 24 for more information on various WMTS layers.</td>
</tr>
<tr>
<td>tileMatrixSet*</td>
<td>String; for example: EPSG:4326 EPSG:3857 EPSG:3395</td>
<td>The tileMatrix set to be used to generate the response. Complete list of supported values are found in the GetCapabilities response.</td>
</tr>
<tr>
<td>tileMatrix*</td>
<td>String; supported values are: EPSG:4326:0 through EPSG:4326:30 EPSG:3857:0 through EPSG:3857:30</td>
<td>The Tile Matrix identifier of the tileMatrix in the tileMatrixSet requested that has the desired scale denominator that you want to request.</td>
</tr>
<tr>
<td>tileRow*</td>
<td>Integer</td>
<td>The Row location of the tile in the defined tileMatrixSet.</td>
</tr>
<tr>
<td>tileCol*</td>
<td>Integer</td>
<td>The Column location of the tile in the defined tileMatrixSet.</td>
</tr>
<tr>
<td>format*</td>
<td>Mime type of the tile to be returned: image/png image/jpeg</td>
<td>The tile format to return.</td>
</tr>
<tr>
<td>connectId*</td>
<td>Character String</td>
<td>User’s unique identifier supplied by DigitalGlobe; required to access the DG Cloud Services.</td>
</tr>
<tr>
<td>exceptions</td>
<td>application/vnd.ogc.se_xml</td>
<td>Format in which exceptions will be reported; if not specified, the default is XML.</td>
</tr>
</tbody>
</table>

* mandatory parameter
6 Tile Cache

6.1 Introduction

This section describes the requirements and procedure to build a Tile Cache into the client repository online/real-time. Tile Cache can save download time thus improving performance of client applications.

6.2 Building a Tile Cache from DigitalGlobe WMTS

A client program can communicate with the DGCS WMTS server using GetTile request for a set of valid Tile Rows and Tile Columns at a given zoom level to build the Tile Cache. The client program has to request GetTile from the WMTS server in order to determine what the WMTS server can do and what features the WMTS server can provide.

More specifically, in a typical WMTS Tile-Cache client/server interaction, the following steps can be followed:

**Build Tile Cache**

Identify the Tile Row, Tile Column range for the required AOI and request for GetTile at different zoom levels and required file formats to receive map tile response.

**GetTile Request**

An example request for a Tile is:

```
```

- Replace `<ConnectID>` with your ConnectID provided by DigitalGlobe.
- Replace `<TILE_ROW>`, `<TILE_COLUMN>`, and `<ZOOM_LEVEL>` with the required values.
- Parameters are not required to be in the same order as shown above.

The above given URL contains Base URL and few parameters as explained in Section 4.5 - Basic Service Elements. The key parameter for this request is "request=GetTile" which fetches the Tiles of WMTS and responds in the form of an image.

The Parameters required for the TileCache request are shown in Table 4.6.

**Caching GetTile Response**

The response to a Tile Cache Request is a map tile, in the requested format and projection. All DGCS-WMTS tiles are 256x256 pixels; no other tile sizes are currently supported.

In the following Tile-Cache output response image, you may note the values of Tile Rows and Tile Columns at zoom level 16.
On successful building, the tile bytes are received from a series of requests, which can be read through respective file operations API of any programming language and can be saved to local disk/required location. The suggested folder structure to save map tiles locally is given below.

Requirements

- Tile Cache building must be executed by the user.
- Tile Cache QA/Error processing must be detected and raised by the user.
- Certain number of threads (governance) to access the system, which is reportable, is provided to the user.
- The customer must know what metadata and format requirements are necessary.
- The tiles X,Y,Z necessary to cover landmass AOI down to zoom 18, should be known by the user.
- User would know how to determine that tiles are all water. It is recommended only going to zoom level 12.
- User would know how to transform the filenames into any caching strategy.
- User would build their ESRI World-file, as needed.

DigitalGlobe delivers file or URLs that can be sourced by the user’s program to go through 80 million+ requests and receive tile bytes. One set of metadata per feature would be delivered.

6.3 Using Tile Cache

While navigating through AOI, the client program can search for tiles at respective locations in the local disk without requesting DGCS-WMTS every time. If the tiles are not available in the local disk for the specified region, the request can be sent to DGCS-WMTS GetTile request and save the tiles accordingly.

6.4 Updating Tile Cache

Client can re-request DGCS-WMTS GetTile to receive recent map tiles and overwrite the corresponding map tiles in local disk. Follow the steps given in section 5.2 to build the tile cache and write to the local disk.
6.5 World-File Generation

If needed by the application, customers must generate their own world-file for each tile. The following link describes the contents of a world file:

http://en.wikipedia.org/wiki/World_file

The z, x, and y values are used to translate from indexes to valid Lat/Lon values. The lat/lon values are used for lines 5 and 6 of the world file.
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

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

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.

WCS
Web Coverage Service.

WebCGM

WFS
Web Feature Service.

WMS
Web Map Service.

WMTS
Web Map Tile Service.
## Index

- area of interest, defined, 29
- bicubic interpolation, defined, 29
- bilinear interpolation, defined, 29
- geographic projection, defined, 29
- GeoTIFF, defined, 29
- GML, defined, 29
- JPEG2000, defined, 29
- MrSid format, defined, 29
- nearest neighbor, defined, 29
- NITF format, defined, 29
- OGC, defined, 29
- OWS, defined, 29
- seam lines defined, 29
- UTM, defined, 29
- WCS, defined, 30
- WebCGM, defined, 30
- WFS, defined, 30
- WMS, defined, 30
- WMTS, defined, 30