
Open GIS Consortium Inc.

Date: 2001-06-21

Reference number of this OpenGIS® project document: OGC 01-047r2

Version: 1.1.0

Category: OpenGIS® Implementation Specification

Status: Adopted Specification

Editor: Jeff de La Beaujardière

Web Map Service Implementation Specification

Document type: OpenGIS® Publicly Available Standard
Document stage: Adopted Specification
Document language: English

***WARNING:** The Open GIS Consortium (OGC) releases this specification to the public without warranty. It is subject to change without notice. This specification is currently under active revision by the OGC Technical Committee*

Requests for clarification and/or revision can be made by contacting the OGC at revisions@opengis.org.

Copyright 1999, 2000, 2001 BBN Technologies
Copyright 1999, 2000, 2001 Cadcorp Ltd.
Copyright 1999, 2000, 2001 CubeWerx Inc.
Copyright 1999, 2000, 2001 IONIC Software s.a.
Copyright 1999, 2000, 2001 Laser-Scan Limited
Copyright 1999, 2000, 2001 SICAD Geomatics GmbH & Co. oHG
Copyright 1999, 2000, 2001 Social Change Online Pty Ltd
Copyright 1999, 2000, 2001 US Army Engineer Research and Development Center

The companies listed above have granted the Open GIS Consortium, Inc. (OGC) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version.

This document does not represent a commitment to implement any portion of this specification in any company's products.

OGC's Legal, IPR and Copyright Statements are found at <http://www.opengis.org/legal/ipr.htm>

NOTICE

Permission to use, copy, and distribute this document in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the above list of copyright holders and the entire text of this NOTICE.

We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of OGC documents is granted pursuant to this license. However, if additional requirements (as documented in the Copyright FAQ at http://www.opengis.org/legal/ipr_faq.htm) are satisfied, the right to create modifications or derivatives is sometimes granted by the OGC to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by government is subject to restrictions as set forth in subdivision (c)(1)(ii) of the Right in Technical Data and Computer Software Clause at DFARS 252.227.7013

OpenGIS® is a trademark or registered trademark of Open GIS Consortium, Inc. in the United States and in other countries.

Contents

i.	Preface	iv
ii.	Submitting Organizations	iv
iii.	Submission Contact Points	iv
iv.	Revision History	v
v.	Changes to the OpenGIS Abstract Specification	v
	Foreword	vi
	Introduction	viii
1	Scope	1
2	Conformance	1
3	Normative references	1
4	Terms and definitions	2
5	Conventions	3
5.1	Normative verbs	3
5.2	Abbreviated Terms	3
6	Basic Service Elements	3
6.1	Version Numbering and Negotiation	4
6.1.1	Version Number Form	4
6.1.2	Version Changes	4
6.1.3	Appearance in Requests and in Service Metadata	4
6.1.4	Version Number Negotiation	4
6.2	General HTTP Request Rules	5
6.2.1	HTTP GET	6
6.2.2	HTTP POST	6
6.3	General HTTP Response Rules	7
6.4	Request Parameter Rules	7
6.4.1	Parameter Ordering and Case	7
6.4.2	Parameter Lists	7
6.5	Common Request Parameters	8
6.5.1	VERSION=version	8
6.5.2	REQUEST=operation_name	8
6.5.3	FORMAT=output_format	8
6.5.4	EXCEPTIONS=exception_format	9
6.5.5	Spatial Reference System	9
6.5.6	Bounding Box	10
6.5.7	TIME=time	12

6.5.8	ELEVATION=elevation	12
6.5.9	Other Sample Dimensions	12
6.5.10	Additional PARAMETER=value pairs.....	13
6.5.11	Vendor-Specific Parameters.....	13
6.6	Service Result.....	13
6.7	Service Exceptions.....	13
7	Web Map Service Operations	14
7.1	GetCapabilities (required).....	15
7.1.1	General	15
7.1.2	GetCapabilities Request Overview	15
7.1.3	Request Parameters	15
7.1.4	UPDATESEQUENCE=number.....	16
7.1.5	GetCapabilities Response	16
7.1.6	Output Formats	22
7.2	GetMap (required)	23
7.2.1	General	23
7.2.2	GetMap Request Overview	23
7.2.3	Request Parameters	24
7.2.4	Vendor-Specific Parameters.....	28
7.2.5	GetMap Response.....	29
7.3	GetFeatureInfo (optional).....	29
7.3.1	General	29
7.3.2	GetFeatureInfo Request Overview	29
7.3.3	Request Parameters	30
7.3.4	GetFeatureInfo Response	32
7.4	DescribeLayer (optional).....	32
Annex A	(normative) XML Document Type Definitions.....	33
A.1	WMS Capabilities DTD (Normative).....	33
A.2	Sample WMS Capabilities XML (Informative)	37
A.3	Service Exception DTD (Normative).....	41
A.4	Sample Service Exception XML (Informative)	41
Annex B	(normative) Formatting Dates and Times	43
B.1	Overview	43
B.2	Time Format Details	43
B.2.1	Basic Syntax.....	43
B.2.2	Extension for years B.C.E.	44
B.2.3	Extension for geologic datasets	44
B.3	Period Format.....	44
B.4	Time Lists and Ranges	44
B.5	Date Fragments.....	45
B.4.1	Truncated representation	45
B.4.2	Days of the week	45
B.5	Examples	45
B.5.1	Complete dates and times.....	45
B.5.2	Date and time fragments.....	45

Annex C (normative) Handling Multi-Dimensional Data	47
C.1 Overview	47
C.2 Declaring Dimensions	47
C.3 Specifying Dimensional Extents.....	48
C.4 Including Dimensional Values in a Request	50
C.4.1 Elevation and Time Values in Requests	50
C.4.2 Sample Dimension Values in Requests.....	51
C.4.3 Single- and Multiple-Valued Requests.....	51
C.4.4 Applicability to Multiple Data Objects	52
C.4.5 Example requests.....	52
C.5 Server Responses	53
C.5.1 Incorrect Values	53
C.5.1 Default Values.....	53
C.5.2 Nearest Values	53
Annex D (normative) Conformance Tests.....	55
Bibliography	56

i. Preface

This document is a partial description of the accomplishments of the Open GIS Consortium (OGC) Web Mapping Testbed, phase 2 (WMT2).

Web Mapping within the OGC was first described in "WWW Mapping Framework" [1]. The first OGC consensus position of the WWW Mapping Special Interest Group, a core task force of the OGC, is described in "User Interaction with Geospatial Data" [2]. From these documents, as well as from "A Web Mapping Scenario" [3], an OGC-sponsored initiative was begun. That initiative, known as the Web Mapping Testbed (WMT), was first described in a Request For Technology (RFT) [4] and then in a Request for Quotation (RFQ) [5].

The WMT Phase I process culminated in the OpenGIS Web Map Service Interface Implementation Specification version 1.0.0 [17], hereinafter "WMS 1.0.0." That first version supported basic interoperability of simple map servers and clients, but did not fully address access to Simple Features, Coverages, data with temporal or other dimensions, and other types of geoprocessing services. Many of these elements were addressed in the follow-on Web Mapping Testbed phase 2 and the Geospatial Fusion Services Testbed. One of the results of those testbeds is this document, which describes a WMS and specifies its behavior.

This version is an extension of, rather than a dramatic departure from, WMS 1.0.0. Substantive differences between the present specification and its predecessor are summarized in the Foreword and are called out in the text where appropriate.

ii. Submitting Organizations

The OGC WWW Mapping Special Interest Group submits this Implementation Specification to the OGC Technical Committee as a Revision to Web Map Service Interface Implementation Specification version 1.0.0.

iii. Submission Contact Points

All questions regarding this submission should be directed to the Editor or to the WWW Mapping SIG chair:

Jeff de La Beaujardière (Editor)
 NASA Goddard Space Flight Center, Code 933
 Greenbelt MD 20771 USA
 +1 301 286 1569
 delabeau@iniki.gsfc.nasa.gov

Allan Doyle (WWW Mapping SIG Chair)
 International Interfaces, Inc.
 948 Great Plain Ave. PMB-182
 Needham, MA 02492 USA
 +1 781 433 2695
 adoyle@intl-interfaces.com

iv. Revision History

Date	Release	Editor	Description
2000-04-19	1.0.0	Allan Doyle	Previous OGC Implementation Specification, to be replaced by this document
2000-10-13	1.0.4	Jeff de La Beaujardière	Web Mapping Testbed 2 development version
2001-01-29	1.0.6	"	OGC Discussion paper #01-021
2001-03-02	1.0.7	"	OGC Discussion Paper #01-021r1
2001-05-14	1.0.8	"	WMS Revision Working Group submittal to OGC TC (document #01-047)

v. Changes to the OpenGIS Abstract Specification

No further revisions to the OGC Abstract Specification are required. The revisions previously approved for Topic 12, "Service Architecture," including definitions of the terms "operation", "interface" and "service" are relevant to and sufficient for this specification.

Foreword

Attention is drawn to the possibility that some of the elements of this part of OGC 01-047r2 may be the subject of patent rights. Open GIS Consortium Inc. shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

This second edition cancels and replaces the first edition (OGC 99-028), which has been technically revised.

Additional Contributors

Craig Bruce
CubeWerx
csbruce@cubewerx.com

Adrian Cuthbert
adrian.cuthbert@bigfoot.com

Allan Doyle
International Interfaces
adoyle@intl-interfaces.com

John Evans,
GST, Inc./NASA GSFC
john.evans@gsfc.nasa.gov

George Percivall
GST, Inc./NASA GSFC
percivall@gsfc.nasa.gov

Summary of Changes from Version 1.0.0

1. Multi-dimensional properties of the source data, including time, elevation and other sample dimensions, may now be declared in service metadata and included in map request operations.
2. Service metadata may now include pointers to external URLs providing dataset-level metadata, map legends, data sources, stylesheets, feature lists, and identification labels defined by other collection authorities.
3. Optional extensions for "Styled Layer Descriptor" [10] map servers have been added.

4. In WMS 1.0, the map request was capable of returning Feature data. That is now the role of a Web Feature Service [7].
5. Several request parameters have been modified or added. In particular:
 - a) The "VERSION" parameter was previously "WMTVER".
 - b) The "SERVICE" parameter did not exist in version 1.
 - c) The "GetCapabilities" request type was previously "capabilities".
 - d) The "GetMap" request type was previously "map".
 - e) The "GetFeatureInfo" request type was previously "feature_info".
6. The allowed responses to a Service Exception have been clarified and restricted.
7. Attributes have been added to describe layers that have been cascaded by another server or that are opaque, not subsettable, or of fixed size.
8. More precise definitions of "operation," "service" and "interface" have been added.
9. Normative comments that were previously in the Capabilities XML DTD have been moved into the text of this specification.

Normative Annexes

All Annexes are normative, except that Subsections A.2 and A.4 are informative.

Introduction

A Web Map Service produces maps of georeferenced data. We define a "map" as a visual representation of geodata; a map is not the data itself. These maps are generally rendered in a pictorial format such as PNG, GIF or JPEG, or occasionally as vector-based graphical elements in Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) formats. This specification standardizes the way in which maps are requested by clients and the way that servers describe their data holdings. This document defines three operations, the first two of which are required of every WMS.

GetCapabilities (required): Obtain service-level metadata, which is a machine-readable (and human-readable) description of the WMS's information content and acceptable request parameters.

GetMap (required): Obtain a map image whose geospatial and dimensional parameters are well-defined.

GetFeatureInfo (optional): Ask for information about particular features shown on a map.

A standard web browser can ask a Web Map Service to perform these operations simply by submitting requests in the form of Uniform Resource Locators (URLs) [12]. The content of such URLs depends on which of the tasks is requested. All URLs include a specification version number and a request type parameter. In addition, when invoking GetMap a WMS Client can specify the information to be shown on the map (one or more "Layers"), possibly the "Styles" of those Layers, what portion of the Earth is to be mapped (a "Bounding Box"), the projected coordinate system to be used (the "Spatial Reference System," or SRS), the desired output format, the output size (Width and Height), and background transparency and color. When invoking GetFeatureInfo the Client indicates what map is being queried and which location on the map is of interest.

When two or more maps are produced with the same Bounding Box, Spatial Reference System, and output size, the results can be accurately layered to produce a composite map. The use of image formats that support transparent backgrounds (e.g., GIF or PNG) allows the lower Layers to be visible. Furthermore, individual map Layers can be requested from different Servers. The WMS GetMap operation thus enables the creation of a network of distributed Map Servers from which Clients can build customized maps.

A particular WMS provider in a distributed WMS network need only be the steward of its own data collection. This stands in contrast to vertically-integrated web mapping sites that gather in one place all of the data to be made accessible by their own private interface.

Because each WMS is independent, a WMS must be able to provide a machine-readable description of its capabilities. This "Service Metadata" enables Clients to formulate valid requests and enables the construction of searchable catalogs that can direct clients to particular WMSes.

A WMS may optionally allow the GetFeatureInfo operation. If it does, its maps are said to be "queryable," and a Client can request information about features on a map by adding to the map URL additional parameters specifying a location (as an X, Y offset from the upper left corner) and the number of nearby features about which to return information.

Cascading Map Servers

A "Cascading Map Server" is a WMS that behaves like a client of other WMSes and behaves like a WMS to other clients. For example, a Cascading Map Server can aggregate the contents of several distinct map servers into one service. Furthermore, a Cascading Map Server can perform additional functions such as output format conversion or coordinate transformation on behalf of other servers.

Styled Layer Descriptors

This specification applies to a Web Map Service that publishes its ability to produce maps rather than its ability to access specific data holdings. A basic WMS classifies its georeferenced information holdings into "Layers" and offers a finite number of predefined "Styles" in which to display those layers.

The behavior of a Web Map Service can be extended to allow user-defined symbolization of feature data instead of named Layers and Styles. The Styled Layer Descriptor (SLD) specification[10] describes this extension. In brief, an SLD-enabled WMS retrieves features from a Web Feature Service [7] and applies explicit styling information provided by the user in order to render a map.

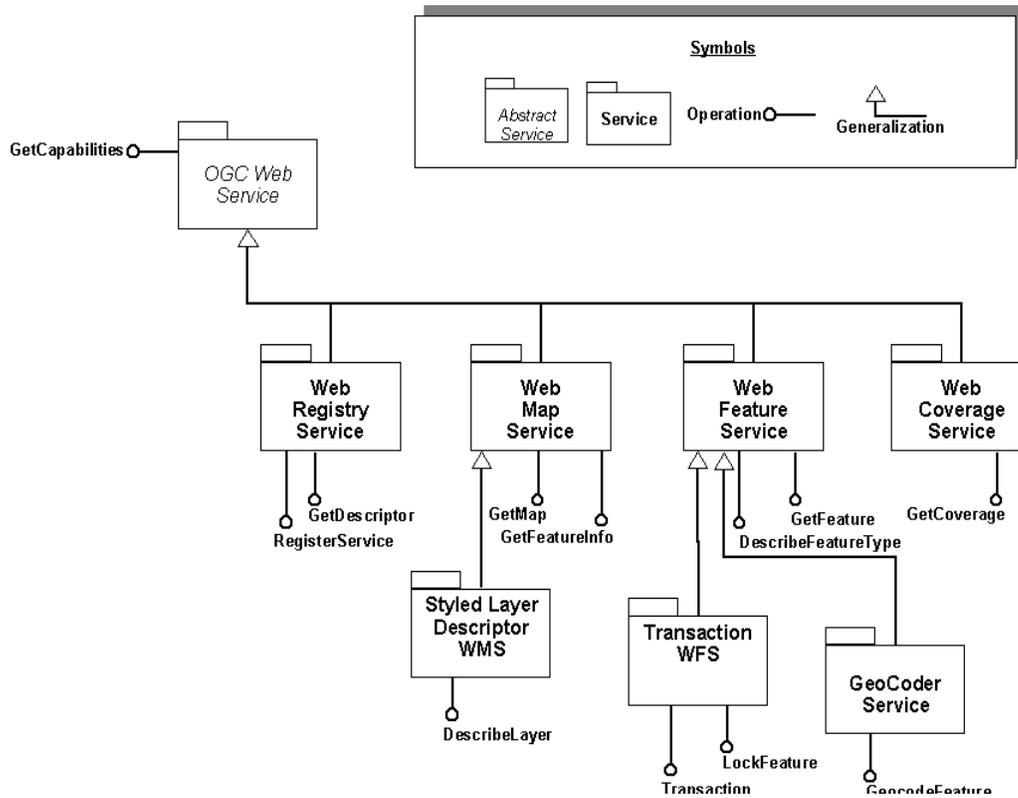
An SLD WMS adds the following additional operation that is not available on a basic WMS:

DescribeLayer (SLD WMS only): Ask for an XML description of a map layer. The result is the URL of the WFS containing the data and the feature type names included in the layer.

Relation to other OGC Web Services

The OGC Web Services (OWS) suite includes three principal types of georeferenced information access services: Web Map Server (WMS), Web Coverage Server (WCS), and Web Feature Server (WFS). In addition, there are services such as GeoParser and GeoCoder that return spatially referenced results. Figure 1 is an architecture diagram showing conceptually how some of the OGC Web Services are related, and naming some (not all) of the operations they define.

Figure 1 — OGC Web Services Architecture diagram



Web Mapping Examples

We conclude this Introduction with some illustrative URLs and their resulting maps.

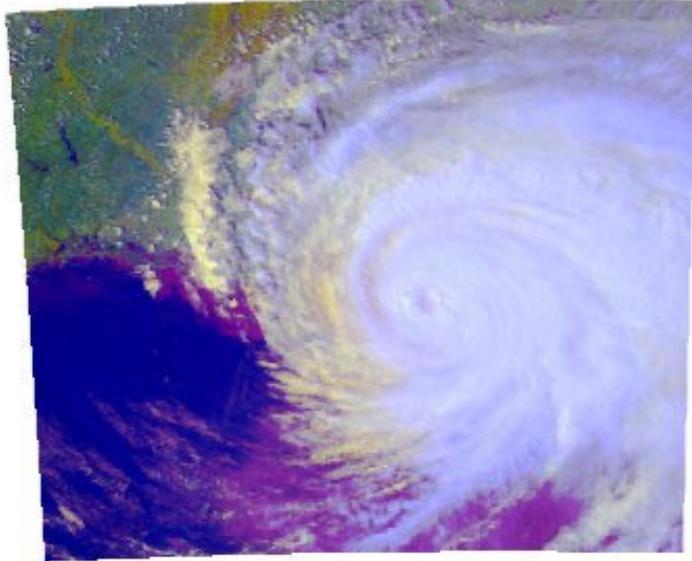
Example 1: One Server, One Layer

The following hypothetical URL requests the US National Oceanographic and Atmospheric Administration hurricane image shown in Figure 2:

```
http://a-map-co.com/mapserver.cgi?VERSION=1.1.0&REQUEST=GetMap&
SRS=EPSG%3A4326&BBOX=-97.105,24.913,78.794,36.358&
WIDTH=560&HEIGHT=350&LAYERS=AVHRR-09-27&STYLES=default&
FORMAT=image/png&BGCOLOR=0xFFFFFFFF&TRANSPARENT=TRUE&
EXCEPTIONS=application/vnd.ogc.se_inimage
```

NOTE: The string "%3A" in a URL represents the ":" character.

Figure 2 — NOAA Hurricane Image of the Gulf of Mexico

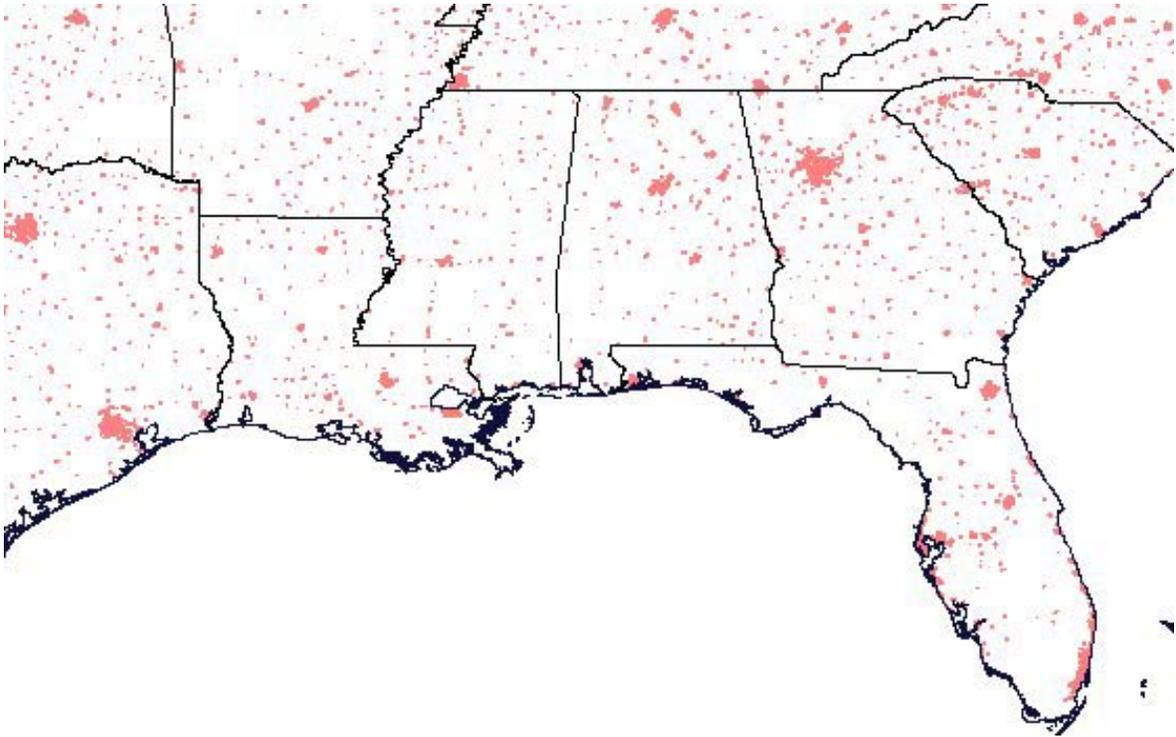


Example 2: One Server, Three Layers

The following hypothetical URL requests three layers--built-up areas, coastlines, and political boundaries --to produce the map shown in Figure 3:

```
http://b-maps.com/map.cgi?VERSION=1.1.0&REQUEST=GetMap&
SRS=EPSG%3A4326&BBOX=-97.105,24.913,78.794,36.358&
WIDTH=560&HEIGHT=350&LAYERS=BUILTUPA_1M,COASTL_1M,POLBNDL_1M&
STYLES=0XFF8080,0X101040,BLACK&FORMAT=image/png&BGCOLOR=0xFFFFFFFF&
TRANSPARENT=TRUE&EXCEPTIONS=application/vnd.ogc.se_inimage
```

Figure 3 — Political, Coastline, and Populated Areas, Southeastern United States



Notice that in both of these URLs the spatial information is identical:

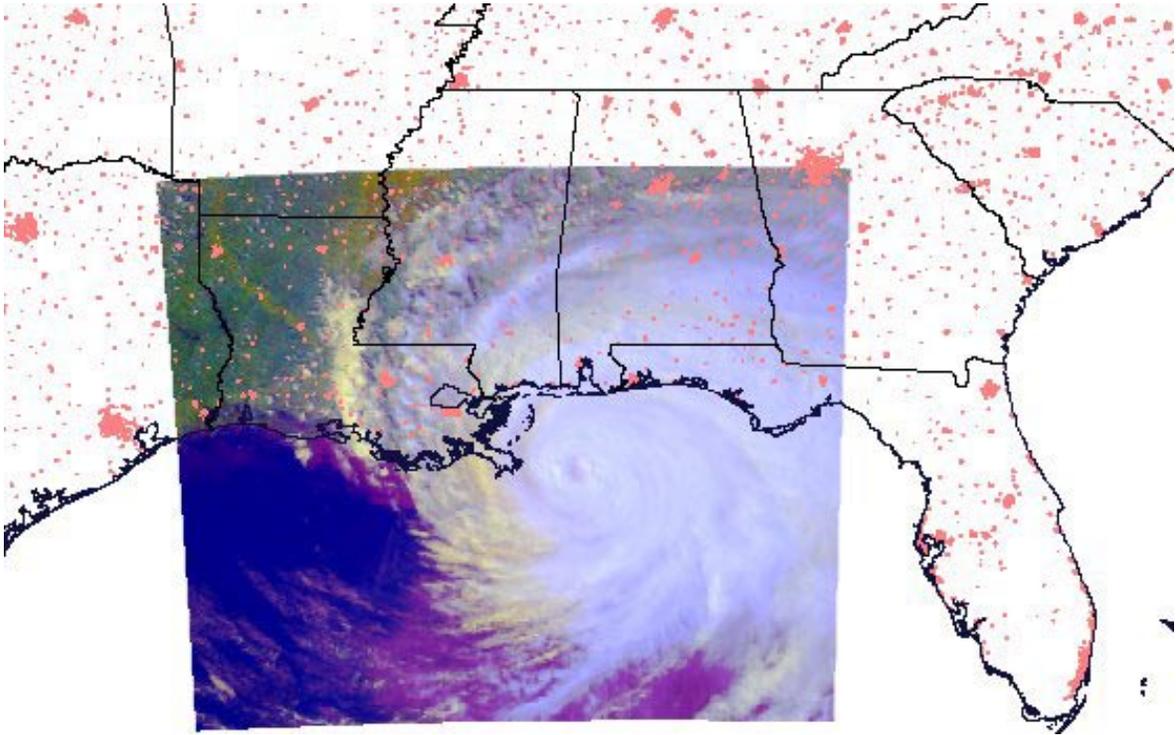
```
SRS=EPSG%3A4326&BBOX=-97.105,24.913,78.794,36.358  
&WIDTH=560&HEIGHT=350.
```

The second map, for which a transparent background was requested (TRANSPARENT=TRUE), can therefore be precisely overlaid on the first.

Example 3: Two Servers, Four Layers

Figure 4 shows the result of overlaying Figure 2 on Figure 3 to produce a composite map from two separate Map Servers.

Figure 4 — Combined Hurricane Image and Population Map



Web Map Service Implementation Specification

1 Scope

This OpenGIS[®] Standard specifies the behavior of a service that produces georeferenced maps. This standard specifies operations to retrieve a description of the maps offered by a service instance, to retrieve a map, and to query a server about features displayed on a map.

This OpenGIS[®] Standard is applicable to pictorial renderings of maps in a graphical format. This standard is *not* applicable to retrieval of actual feature data or coverage data values.

2 Conformance

Conformance with this specification shall be checked using all the relevant tests specified in Annex D (normative). The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in ISO 19105: Geographic information — Conformance and Testing.

3 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this specification are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

Berners-Lee, T., Fielding, N., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", IETF RFC 2396, August 1998, <http://www.ietf.org/rfc/rfc2396.txt>.

Bray, Paoli, Sperberg-McQueen, eds., "Extensible Markup Language (XML) 1.0", 2nd edition, W3C Recommendation, October 2000, <http://www.w3.org/TR/2000/REC-xml>.

European Petroleum Survey Group, "EPSG Geodesy Parameters", <http://www.epsg.org/>.

Fielding et. al., "Hypertext Transfer Protocol – HTTP/1.1," IETF RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>.

Freed, N. and Borenstein N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", IETF RFC 2045, November 1996, <http://www.ietf.org/rfc/rfc2045.txt>.

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1

Operation

Specification of an interaction that can be requested from an object to effect behavior (definition from ISO 19119).

4.2

Interface

An implementation of operations including the syntax of the interaction for a given distributed computing technology (definition from ISO 19119).

4.3

Service

A collection of operations, accessible through an interface, that allows a user to evoke a behavior of value to the user (definition from ISO 19119).

4.4

Service Instance

An actual implementation of a Service. Service Instance is synonymous with Server.

4.5

Client

A software component that can invoke an Operation from a Server.

4.6

Request

An invocation by Client of an Operation.

4.7

Response

The result of an Operation returned from a Server to a Client.

4.8

Capabilities XML

Service-level metadata describing the operations and content available at a Service Instance.

5 Conventions

5.1 Normative verbs

In the sections labeled as normative, the key words "**must**", "**must not**", "**required**", "**shall**", "**shall not**", "**should**", "**should not**", "**recommended**", "**may**", and "**optional**" in this document are to be interpreted as described in Internet RFC 2119 [15].

5.2 Abbreviated Terms

CGI	Common Gateway Interface
DCP	Distributed Computing Platform
DTD	Document Type Definition
EPSG	European Petroleum Survey Group
GIF	Graphics Interchange Format
GIS	Geographic Information System
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
JPEG	Joint Photographic Experts Group
MIME	Multipurpose Internet Mail Extensions
OGC	Open GIS Consortium
OWS	OGC Web Service
PNG	Portable Network Graphics
RFC	Request for Comments
SLD	Styled Layer Descriptor
SVG	Scalable Vector Graphics
URL	Uniform Resource Locator
WebCGM	Web Computer Graphics Metafile
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
XML	Extensible Markup Language

6 Basic Service Elements

This clause specifies aspects of Web Map Server behavior (more generally, of OGC Web Service behavior) that are independent of particular operations or are common to several operations.

6.1 Version Numbering and Negotiation

6.1.1 Version Number Form

The published specification 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 OWS specification is numbered independently.

6.1.2 Version Changes

A particular specification's version number **shall** be changed with each revision. The number **shall** increase monotonically and **shall** comprise no more than three integers separated by decimal points, with the first integer being the most significant. There may be gaps in the numerical sequence. Some numbers may denote experimental or interim versions. Service instances and their clients need not support all defined versions, but **must** obey the negotiation rules below.

6.1.3 Appearance in Requests and in Service Metadata

The version number appears in at least two places: in the Capabilities XML describing a service, and in the parameter list of client requests to that service. The version number used in a client's request of a particular service instance **must** be equal to a version number which that instance has declared it supports (except during negotiation as described below). A service instance may support several versions, whose values clients may discover according to the negotiation rules.

6.1.4 Version Number Negotiation

An OWS Client may negotiate with a Service Instance to determine a mutually agreeable specification version. Negotiation is performed using the GetCapabilities operation (described in Section 7.1) according to the following rules.

All Capabilities XML must include a protocol version number. In response to a GetCapabilities request containing a version number, an OGC Web Service **must** either respond 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 **must** respond with the highest version it understands and label the response accordingly.

Version number negotiation occurs as follows:

- 1) If the server implements the requested version number, the server **must** send that version.
- 2a) If a version unknown to the server is requested, the server **must** send the highest version less than the requested version.

2b) If the client request is for a version lower than any of those known to the server, then the server **must** send the lowest version it knows.

3a) If the client does not understand the new version number sent by the server, it **may** either cease communicating with the server **or** send a new request with a new version number that the client does understand but which is less than that sent by the server (if the server had responded with a lower version).

3b) If the server had responded with a higher version (because the request was for a version lower than any known to the server), and the client does not understand the proposed higher version, then the client **may** send a new request with a version number higher than that sent by the server.

The process is repeated until a mutually understood version is reached, or until the client determines that it will not or cannot communicate with that particular server.

Example 1: Server understands versions 1, 2, 4, 5 and 8. Client understands versions 1, 3, 4, 6, and 7. Client requests version 7. Server responds with version 5. Client requests version 4. Server responds with version 4, which the client understands, and the negotiation ends successfully.

Example 2: Server understands versions 4, 5 and 8. Client understands version 3. Client requests version 3. Server responds with version 4. Client does not understand that version or any higher version, so negotiation fails and client ceases communication with that server.

6.2 General HTTP Request Rules

At present, the only distributed computing platform (DCP) explicitly supported by OGC Web Services is the World Wide Web itself, or more specifically Internet hosts implementing the Hypertext Transfer Protocol (HTTP)[8]. Thus the Online Resource of each operation supported by a service instance is an HTTP Uniform Resource Locator (URL). The URL may be different for each operation, or the same, at the discretion of the service provider. Each URL **must** conform to the description in [8] (Section 3.2.2 "HTTP URL") but is otherwise implementation-dependent; only the parameters comprising the service request itself are mandated by the OGC Web Services specifications.

HTTP supports two request methods: GET and POST. One or both of these methods may be defined for a particular OGC Web Service type and offered by a service instance, and the use of the Online Resource URL differs in each case. The basic WMS specification only defines HTTP GET for invoking operations. (A Styled Layer Descriptor WMS[10] defines HTTP POST for some operations.)

6.2.1 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 appends 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[16], which mandates the presence of '?' before the sequence of query parameters and the '&' between each parameter. As with all CGI applications, the query URL is encoded[12] to protect special characters.

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 in this specification in order to construct a valid request URL.

Table 1 summarizes the components of an operation request URL.

Table 1 — A general OGC Web Service 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 an OGC Web Service. The actual list of required and optional parameters is mandated for each operation by the appropriate OWS specification.

6.2.2 HTTP POST

An Online Resource URL intended for HTTP POST requests is a complete and valid URL to which Client transmit request parameters in the body of the POST request. A WMS **must not** require additional parameters to be appended to the URL in order to construct a valid target for the Operation request.

Operation requests using HTTP POST have not yet been defined for the basic Web Map Service.

6.3 General HTTP Response Rules

Upon receiving a valid request, the service **must** send a response corresponding exactly to the request as detailed in the appropriate specification. Only in the case of Version Negotiation (described above) may the server offer a differing result.

Upon receiving an invalid request, the service **must** issue a Service Exception as described in Section 6.7.

NOTE: As a practical matter, in the WWW environment a client should be prepared to receive either a valid result, or nothing, or any other result. This is because the client may itself have formed a non-conforming request that inadvertently triggered a reply by something other than an OGC Web Service, because the Service itself may be non-conforming, etc.

Response objects **must** be accompanied by the appropriate Multipurpose Internet Mail Extensions (MIME) type[11] for that object. Allowable types for operation responses and service exceptions are discussed below.

Response objects **should** 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.

6.4 Request Parameter Rules

6.4.1 Parameter Ordering and Case

Parameter names **shall not** be case sensitive, but parameter values **shall** be case sensitive. In this document, parameter names are typically shown in uppercase for typographical clarity, not as a requirement.

Parameters in a request **may** be specified in any order.

An OGC Web Service **must** be prepared to encounter parameters that are not part of this specification. In terms of producing results per this specification, an OGC Web Service **shall** ignore such parameters.

6.4.2 Parameter Lists

Parameters consisting of lists (for example, the LAYERS and STYLES in WMS GetMap) **must** use the comma (",") as the separator between items in the list. Additional white space **must not** be used to delimit list items.

List items **may** be empty, as represented by two successive commas (",,").

6.5 Common Request Parameters

6.5.1 VERSION=version

VERSION specifies the protocol version number. The format of the version number, and version negotiation, are described in Section 6.1.

6.5.2 REQUEST=operation_name

REQUEST indicates which service operation is being invoked. The value operation_name **must** be one of those offered by the OGC Web Service Instance.

6.5.3 FORMAT=output_format

This parameter specifies the output format of the response to an operation.

An OGC Web Service **may** offer only a subset of the formats known for that type of operation, but the server **shall** advertise in its Capabilities XML those formats it does support and **must** accept requests for any format it advertises. A Service Instance **may** optionally offer a new format not previously offered by other instances, with the recognition that clients are not required to accept or process an unknown format. If a request contains a Format not offered by a particular server, the server **must** throw a Service Exception (with code "InvalidFormat").

A Client **may** accept only a subset of the formats known for that type of operation. If a Client and Service do not support any mutually agreeable formats, the Client may, at its discretion, cease communicating with that service, or search for an intermediary service provider that performs format conversion, or allow the user to choose other disposition methods (e.g., saving to local storage or passing to helper application).

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.

Generally, OGC Web Service MIME types are chosen from among those in common use on the Internet[18]. However, additional OGC-specific types have been adopted to distinguish among different types of XML-formatted content (the generic XML MIME types being text/xml and application/xml), as listed in Table 2.

Table 2 — OGC-Specific MIME Types

MIME Type	Document Content
application/vnd.ogc.wms_xml	WMS Capabilities XML
application/vnd.ogc.gml	Geography Markup Language XML [6]
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.

6.5.4 EXCEPTIONS=exception_format

The format in which to report errors. See Section 6.7 on Service Exceptions, below.

6.5.5 Spatial Reference System

The Spatial Reference System (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. This specification defines two namespaces, EPSG and AUTO, which are defined below.

NOTE: The use of the term SRS is in keeping with WMS 1.0.0. A more modern definition uses Coordinate Reference System (CRS) when referring to spatial referencing by coordinates, and SRS when referring to spatial referencing by addresses or indexes.

OGC Web Services are **not** required to support all possible SRSEs, but **shall** advertise in their Capabilities XML those projections which they do offer and **must** accept requests for all advertised projections. If a request contains an SRS not offered by a particular server, the server **must** throw a Service Exception (code = "InvalidSRS").

Clients are **not** required to support all possible SRSEs. If a Client and Service do not support any mutually agreeable SRS, the Client may, at its discretion, cease communicating with that service, or search for an intermediary service provider that performs coordinate transformations, or allow the user to choose other disposition methods.

6.5.5.1 EPSG Namespace for SRS

The EPSG namespace makes use of the European Petroleum Survey Group tables [13], which define numeric identifiers (the EPSG "horiz_cs" code) 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 only the prefix and the identifier, not any additional parameters. 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.

The SRS "EPSG:4326" refers to a Platte Carrée projection of the WGS84 longitude/latitude coordinates in decimal degrees with Greenwich as the central meridian.

6.5.5.2 AUTO Namespace for SRS

The AUTO namespace is used for "automatic" projections; that is, a class of projections that includes an arbitrary central meridian [14]. An SRS name in the AUTO namespace includes an identifier, a code indicating what units are to be used for bounding boxes in that SRS, and values for the central longitude and latitude:

AUTO:wm_id, epsg_units, lon, lat

wm_id is one of the projection IDs defined by this specification in [14]; epsg_units is one of the EPSG-defined [13] numbers for units, and lon,lat are longitude,latitude in decimal degrees.

Example: "AUTO:42003,9001,-100,45" is auto orthographic projection, bounding box in meters, center at 100 West, 45 North. In addition to the Auto Orthographic projection, Auto UTM (Universal Transverse Mercator), Auto Transverse Mercator (with the given central meridian), and Auto Equirectangular (which uses the given central latitude to scale the X axis) are defined. See reference [14] for additional details.

In a request for a georeferenced map or data, the complete AUTO SRS is specified including longitude, latitude, and units. In Capabilities XML, the longitude, latitude, and units variables are omitted because they are chosen by the client in a request for an AUTO SRS.

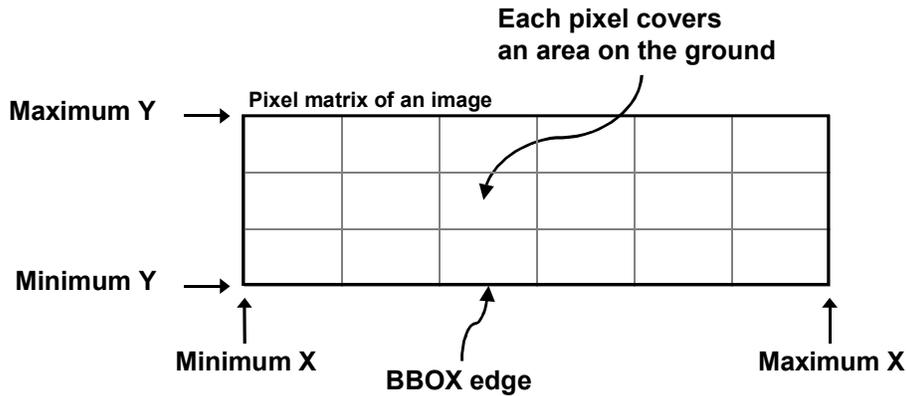
6.5.5.3 Undefined SRS

A Server may offer geographic information whose precise spatial reference is undefined. For example, a digitized collection of hand-drawn historical maps may represent an area of the Earth but not employ a modern coordinate system. In such case, the value "NONE" (case-insensitive) **must** be used when declaring the SRS of such a collection or object. Clients **should not** attempt to overlay information whose SRS=none with other information.

6.5.6 Bounding Box

The Bounding Box (BBOX) is a set of four comma-separated decimal, scientific notation, or integer values (if integers are provided where floating point is needed, the decimal point is assumed at the end of the number). These values specify the minimum X, minimum Y, maximum X, and maximum Y ranges, in that order, expressed in units of the SRS of the request, such that a rectangular area is defined in those units. When the SRS is a Platte Carrée projection of longitude and latitude coordinates, X refers to the longitudinal axis and Y to the latitudinal axis. The four bounding box values indicate the outside edges of a rectangle, as in Figure 5: 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.

Figure 5 — Bounding Box representation



A Bounding Box **should not** have zero area.

If a request contains an invalid Bounding Box (e.g., one whose minimum X is greater than or equal to the maximum X, or whose minimum Y is greater than or equal to the maximum Y) the server **must** throw an exception.

If a request contains a Bounding Box whose area does not overlap at all with the BoundingBox advertised in the Capabilities XML for the requested geodata object, the server **should** return empty content (e.g., a blank map, empty coverage file, null feature set) for that element. Any elements that are partly or entirely contained in the Bounding Box **should** be returned in the appropriate format.

If the Bounding Box values are not defined for the given SRS (e.g., latitudes greater than 90 degrees in EPSG:4326), the server **should** return empty content for areas outside the valid range of the SRS.

In the particular case of longitude, the following behavior applies regarding the anti-meridian at 180 degrees of longitude. There is a legitimate desire for maps that span the anti-meridian (for example, a map centered on the Pacific Ocean). If Xmin is the west-most longitude in degrees and Xmax is the east-most, then the following constraint applies:

$$-180 \leq X_{\min} < X_{\max} < 540$$

EXAMPLES: Xmin,Xmax values and the corresponding scope of the bounding box:

-180,180 = Earth centered at Greenwich
0,360 = Earth with Greenwich at left edge
120,250 = Pacific Ocean

6.5.7 TIME=time

Some geospatial information may be available at multiple times (for example, an hourly weather map). An OGC Web Service may announce available times in its Capabilities XML, and some operations include a parameter for requesting a particular time. The format of a time string is specified in Annex C. Depending on the context, time values may appear as a single value, a list of values, or an interval, as specified in Annex D. When providing temporal information, Servers **should** declare a default value in Capabilities XML unless there is compelling reason to behave otherwise, and Servers **must** respond with the default value if one has been declared and the Client request does not include a value.

6.5.8 ELEVATION=elevation

Some geospatial information may be available at multiple elevations (for example, ozone concentrations at different heights in the atmosphere). An OWS may announce available elevations in its Capabilities XML, and some operations include a parameter for requesting a particular elevation. A single elevation value is an integer or real number whose units are declared by naming an EPSG datum. Depending on the context, elevation values may appear as a single value, a list of values, or an interval, as specified in Annex D. When providing elevation information, Servers **should** declare a default value in Capabilities XML unless there is compelling reason to behave otherwise, and Servers **must** respond with the default value if one has been declared and the Client request does not include a value.

6.5.9 Other Sample Dimensions

Some geospatial information may be available at other dimensions (for example, satellite images in different wavelength bands). The dimensions other than the four space-time dimensions are referred to as "sample dimensions". An OWS may announce available sample dimensions in its Capabilities XML, and some operations include a mechanism for including dimensional parameters. Each sample dimension has a Name and one or more valid values. The declaration and use of sample dimensions are specified in Annex D.

6.5.10 Additional PARAMETER=value pairs

Most service requests require additional parameters (beyond REQUEST) to unambiguously state what result to construct. Each OGC Web Service specification defines the required and optional parameters for its operation(s).

6.5.11 Vendor-Specific Parameters

Finally, the requests allow for optional vendor-specific parameters (VSPs) that will enhance the results of a request. Typically, these are used for private testing of non-standard functionality prior to possible standardization. A generic client is **not** required or expected to make use of these VSPs.

An OGC Web Service **must** produce a valid result even if VSPs are missing or malformed (i.e., the Service **shall** supply a default value), or if VSPs are supplied that are not known to the Service (i.e., the Service **shall** ignore unknown request parameters).

An OGC Web Service **may** choose not to advertise some or all of its VSPs. If VSPs are included in the Capabilities XML, then they **must** be defined within an internal DTD section of that XML document. (An internal DTD comprises declarations enclosed in square brackets [] within the <DOCTYPE> element of the XML [see ref. 9].) In the absence of such parameters, the internal DTD is absent.

Clients **may** read the internal DTD and formulate requests using any VSPs advertised therein.

Vendors **should** choose vendor-specific parameter names with care to avoid clashes with standard parameters.

6.6 Service Result

The return value of a valid Service request **shall** correspond to the type requested in the FORMAT parameter. In an HTTP environment, the Content-type header of the response **shall** be exactly the MIME type given in the request.

Several OGC-specific MIME types have been defined in Table 2 for various XML document types (all of which would traditionally be labeled "text/xml"). To be compliant with this Specification a server **must** return the appropriate OGC MIME type if defined, and the client **must** be able to accept it, but it is **recommended** that the client also be prepared to accept the MIME type "text/xml" and deduce the specific content type through other means.

6.7 Service Exceptions

Upon receiving a request that is invalid according to the rules of the Distributed Computing Platform (DCP) in use, the service **may** issue an exception of a type valid in that DCP. For example: in the HTTP DCP, if the URL prefix is incorrect an HTTP 404 error code is sent.

Upon receiving a request that is invalid according to the relevant OGC Web Services specification, the service **must** issue a Service Exception as defined here. The Exception is meant to describe to the client application or its human user the reason(s) that the request is invalid.

NOTE: Change from version 1.0.0: A WMS was previously permitted to return nothing even if it could have constructed an exception report.

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 all OGC Web Services is "application/vnd.ogc.se_xml," which means "Service Exception XML." Particular services may define other formats; for example, a WMS may offer to write the text of the error message in an image via the "application/vnd.ogc.se_inimage" format.

NOTE: A client should also be prepared for other returned values and types since there is a possibility that the Service instance is poorly behaved or that a request was directed at a non-compliant OGC Web Service.

Service Exception XML **must** be valid according to the Service Exception DTD in Annex A.3. In an HTTP environment, the MIME type of the returned XML **must** be "application/vnd.ogc.se_xml". Specific error messages can be included either as chunks of plain text or as XML-like text containing angle brackets (" $<$ " and " $>$ ") if included in a character data (CDATA) section as shown in the example of Service Exception XML in Annex A.4.

Service Exceptions **may** include error codes as indicated in Annex A.3. Services **shall not** use these codes for meanings other than those specified. Specific service interface specifications will define the content and meaning of additional Exception Codes. Clients **may** use these codes to automate responses to Service Exceptions.

The specific codes and semantics of allowed exceptions may be extended by each specification.

NOTE: Changes from version 1.0.0: The MIME type application/vnd.ogc.se_xml was not previously used. The MIME type of the XML error report was previously "text/xml." The use of error codes was not previously defined.

7 Web Map Service Operations

The three operations defined for a Web Map Service are GetCapabilities, GetMap, and GetFeatureInfo. This section specifies the implementation and use of these WMS operations in the Hypertext Transfer Protocol (HTTP) Distributed Computing Platform (DCP). Future versions may apply to other DCPs.

NOTE: As discussed in the Introduction, an SLD-enabled WMS also offers a DescribeLayer operation, which is specified in reference [10].

7.1 GetCapabilities (required)

7.1.1 General

The purpose of the GetCapabilities operation is described in the Basic Service Elements section, above. In the particular case of a Web Map Service, the response of a GetCapabilities request is general information about the service itself and specific information about the available maps.

7.1.2 GetCapabilities Request Overview

The general form of a GetCapabilities request is defined in the Basic Service Elements section. When making this request of a WMS, which may offer other OGC Web Services as well, it is necessary to indicate that the client seeks information about the WMS in particular. Thus, the SERVICE parameter of the request **must** have the value "WMS" as shown in Table 3 below.

Table 3 — The parameters of a GetCapabilities request URL

Request Parameter	Required/ Optional	Description
VERSION=version	O	Request version
SERVICE=WMS	R	Service type
REQUEST=GetCapabilities	R	Request name
UPDATESEQUENCE=number	O	Sequence number for cache control

7.1.3 Request Parameters

7.1.3.1 VERSION=version

This parameter, and its use in version negotiation, is specified in the Basic Service Elements section.

In WMS version 1.0.0, the name of this parameter was "WMTVER". That name is now deprecated, but for backwards compatibility and version negotiation a post-1.0.0 server **must** accept either form without issuing a Service Exception. In the case that VERSION and WMTVER are both given, VERSION takes precedence.

NOTE: Change from version 1.0.0: The name of this parameter was previously "WMTVER".

7.1.3.2 SERVICE=service_name

SERVICE indicates which of the available service types at a particular service instance is being invoked. This parameter allows the same URL prefix to offer Capabilities XML for multiple OGC Web Services.

When invoking GetCapabilities on a WMS that implements this version of the specification or a later one, the service_name value "WMS" **must** be used.

When invoking GetCapabilities on a WMS that implements version 1.0.6 or earlier, the SERVICE parameter **should not** be used by Clients and **may** be ignored by Servers. An earlier server **must not** issue an Exception upon encountering this parameter (or any other unknown parameter), as specified in Section 6.2.5.1.4 of WMS 1.0.0[17] and Section 6.4.1 of the present specification.

NOTE: Change from version 1.0.0: This is a new parameter.

7.1.3.3 REQUEST=GetCapabilities

This nature of this parameter is specified in the Basic Service Elements section. To invoke the GetCapabilities operation, the value "GetCapabilities" is used. In WMS version 1.0.0, the value of this parameter was "capabilities". That value is now deprecated, but for backwards compatibility a post-1.0.0 server **must** accept either form without issuing a Service Exception. When a client is initially contacting a WMS whose version it does not know the Client **should** be prepared to recover if REQUEST=GetCapabilities fails and **may** send REQUEST=capabilities.

NOTE: Change from version 1.0.0: The value of this parameter was previously "capabilities".

7.1.4 UPDATESEQUENCE=number

UpdateSequence is an optional parameter for maintaining cache consistency. It is envisaged that this number will be a timestamp. The server **may** include an UpdateSequence number in its Capabilities XML. If present, this number **should** be increased when changes are made to the Capabilities (e.g., when new maps are added to the service). The client **may** include this parameter in its GetCapabilities request. The response of the server based on the presence and relative value of UpdateSequence in the client request and the server metadata **must** be according to Table 4:

Table 4 — Use of UpdateSequence Parameter

Client Request UpdateSequence Value	Server Metadata UpdateSequence Value	Server Response
none	any	most recent Capabilities XML
any	none	most recent Capabilities XML
equal	equal	Exception: code=CurrentUpdateSequence
lower	higher	most recent Capabilities XML
higher	lower	Exception: code=InvalidUpdateSequence

7.1.5 GetCapabilities Response

The Basic Service Elements section specifies general rules about the GetCapabilities response.

In the particular case of a Web Map Service complying with this version of the standard, the Extensible Markup Language (XML)[9] response must be valid according to the XML Document Type Definition (DTD) in Annex A.1 of this document. The DTD specifies the required and optional content of the response and how the content is formatted.

A server may comply with other published or experimental versions, in which case it must support Version Negotiation as described in the Basic Service Elements section. A DTD for version 1.0.0 was published as an annex to that version of the WMS specification. Other DTDs are archived at <http://www.digitalearth.gov/wmt/xml/>.

7.1.5.1 Names vs. Titles

A number of elements have both a <Name> and a <Title>. Typically, the Name is a single word used for machine-to-machine communication while the Title is for the benefit of humans. For example, a dataset might have the Title "Maximum Atmospheric Temperature" and be requested using the Name "ATMAX".

7.1.5.2 General Service Metadata

The first part of the Capabilities XML is a <Service> element providing general metadata for the service as a whole. It **must** include a Name, Title, and Online Resource URL. Optionally, Abstract, Keyword List, Contact Information, Fees, and Access Constraints **may** be provided.

The Service Name **must** be "OGC:WMS" in the case of a Web Map Service.

The Service Title is at the discretion of the provider, and **should** be brief yet descriptive enough to identify this server in a menu with other servers.

The Abstract element allows a descriptive narrative providing more information about the enclosing object.

The Service OnlineResource can be used, for example, to point to the web site of the service provider. There are other OnlineResource elements used for the URL prefix of each supported operation.

A list of keywords or keyword phrases **should** be included to help catalog searching. Currently, no controlled vocabulary has been defined.

Contact Information **should** be included.

The reserved word "none" (case-insensitive) **must** be used if there are no fees or access constraints, as follows: <Fees>none</Fees>, <AccessConstraints>none</AccessConstraints>. When constraints are imposed, no precise syntax has been defined for the place-holder elements.

7.1.5.3 Capability Metadata

The <Capability> element of the Capabilities XML names the actual operations that are supported by the service instance, the output formats offered for those operations, and the URL prefix for each operation. The XML DTD includes placeholders for Distributed Computing Platforms other than HTTP, and request methods other than HTTP GET, but currently only HTTP GET is defined for a basic WMS.

Ignorable vendor-specific elements **may** be included as discussed in Section 6.5.11. An SLD WMS would also include a <UserDefinedSymbolization> element and URLs for HTTP POST requests.

7.1.5.4 Layers and Styles

The most critical part of the WMS Capabilities XML is the Layers and Styles it defines.

Each available map is advertised by a <Layer> element in the Capabilities XML. A single parent Layer encloses any number of additional layers, which may be hierarchically nested as desired. A <Title> is required for all layers; it is a human-readable string for presentation in a menu. If, and only if, the layer has a <Name>, then it is a map layer that can be requested by using that Name in the LAYERS parameter of a GetMap request. If the layer has a Title but no Name, then that layer is only a category title for all the layers nested within. A Map Server that advertises a Layer containing a Name element **must** be able to accept that Name as the value of LAYERS argument in a GetMap request and return the corresponding map. A Client **must not** attempt to request a layer that has a Title but no Name. A server **must** throw an exception (code="LayerNotDefined") if an invalid layer is requested.

Layers are hierarchically nested in the Capabilities XML. Some properties defined in a parent layer are inherited by the children it encloses. These inherited properties may be either redefined or added to by the child. Section 7.1.5.6 defines whether or how each property is inherited.

A Map Server **must** include at least one <Layer> element for each map layer offered. If desired, layers **may** be repeated in different categories when relevant.

A containing category itself **may** include a Name by which all of the nested layers can be requested at once. For example, a parent layer "Roads" may have children "Interstates" and "State Highways" and allow the user to request either child individually or both together.

The <Abstract> and <KeywordList> parameters are optional but **recommended**.

Every Layer **must** have at least one <SRS> element that is either stated explicitly or inherited from a parent Layer (Section 7.1.5.6). The root Layer element **must** include a list of all SRSSes that are common to all subsidiary layers. This allows clients to quickly determine that a server cannot possibly satisfy a request for a particular SRS. Use an empty SRS list if there is no common SRS. Optionally, layers may add to the global SRS

list, or to the list inherited from a parent layer. Any duplication **shall** be ignored by clients.

Every Layer **must** have exactly one <LatLonBoundingBox> element that is either stated explicitly or inherited from a parent Layer (Section 7.1.5.6). The LatLonBoundingBox attributes indicate the edges of the enclosing rectangle in latitude/longitude decimal degrees (as in SRS EPSG:4326 [WGS1984 lat/lon]). LatLonBoundingBox **must** be supplied regardless of what SRS the map server may support, but it **may** be approximate if EPSG:4326 is not supported. Its purpose is to facilitate geographic searches without requiring coordinate transformations by the search engine.

Layers **may** have zero or more SRS-specific <BoundingBox> elements. The BoundingBox attributes indicate the edges of the bounding box in units of the specified spatial reference system as in Figure 5. **Optional** resx and resy attributes indicate the spatial resolution of the data in those same units. A BoundingBox for EPSG:4326 **may** be included in order, for example, to provide resolution information.

Layers **may** include a <ScaleHint> element that suggests minimum and maximum scales for which it is appropriate to display this layer. Because WMS output is destined for output devices of arbitrary size and resolution, the usual definition of scale as the ratio of map size to real-world size is not appropriate here. The following definition of Scale Hint is **recommended**. Consider a hypothetical map with a given Bounding Box, width and height. The central pixel of that map (or the pixel just to the northwest of center) will have some size, which can be expressed as the ground distance in meters of the southwest to northeast diagonal of that pixel. The two values in ScaleHint are the minimum and maximum recommended values of that diagonal. It is recognized that this definition is not geodetically precise, but at the same time the hope is that by including it conventions will develop that can be later specified more clearly.

Zero or more Styles may be advertised for a Layer or collection of layers using <Style> elements, which also have <Name>, <Title> and other elements. The style's Name is used in the Map request STYLES parameter. If only a single style is available, that style is known as the "default" style and need not be advertised by the server.

No controlled vocabulary has been defined, so at present Layer and Style Names, Titles and Keywords are arbitrary.

7.1.5.5 Layer Attributes

A <Layer> **may** have zero or more of the following XML attributes: queryable, cascaded, opaque, noSubsets, fixedWidth, fixedHeight. All of these attributes are **optional** and default to 0, in keeping with WMS 1.0.0. Each of these attributes can be inherited or replaced by subsidiary layers. The meaning of each attribute is summarized in Table 5 and detailed in the following subsections.

Table 5 — Layer Attributes

Attribute	Allowed Values	Meaning (0 is default value)
queryable	0, 1	0: not queryable. 1: queryable.
cascaded	0, positive integer	0: layer has not been retransmitted by a Cascading Map Server. n: layer has been retransmitted n times.
opaque	0, 1	0: map data represents vector features that probably do not completely fill space. 1: map data are mostly or completely opaque.
noSubsets	0, 1	0: WMS can map a subset of the full bounding box. 1: WMS can only map the entire bounding box.
fixedWidth	0, positive integer	0: WMS can resize map to arbitrary width. nonzero: map has a fixed width that cannot be changed by the WMS.
fixedHeight	0, positive integer	0: WMS can resize map to arbitrary height. nonzero: map has a fixed height that cannot be changed by the WMS.

7.1.5.5.1 Queryable layers

A Layer is said to be "queryable" if the server supports the GetFeatureInfo operation on that Layer. A server **may** support GetFeatureInfo on some of its layers but not on all. A server **must** issue a Service Exception (code="LayerNotQueryable") if GetFeatureInfo is requested on a Layer that is not queryable.

7.1.5.5.2 Cascaded layers

A Layer is said to have been "cascaded" if it was obtained from an originating server and then included in the Capabilities XML of a different server. The second server may simply offer an additional access point for the Layer, or may add value by offering additional output formats or spatial reference systems.

If a WMS cascades the content of another WMS then it **must** increment by 1 the value of the **cascaded** attribute for the affected layers. If that attribute is missing from the originating WMS's Capabilities XML, then the Cascading WMS **must** insert the attribute and set it to 1.

7.1.5.5.3 Opaque vs. transparent layers

If the optional **opaque** attribute is missing or has a value of "0," then maps made from that Layer will generally have significant "no-data" areas that a client may display as transparent. Vector features such as points and lines are considered not to be opaque in this context (even though at some scales and symbol sizes a collection of features might fill the map area). A value of "1" indicates that the Layer represents an area-filling coverage of space. For example, a map that represents topography and bathymetry as

regions of differing colors will have no transparent areas. The "opaque" declaration **should** be taken as a hint to the Client to place such a Layer at the bottom of a stack of maps.

This attribute describes only the Layer's data content, not the picture format of the map response. Whether or not a Layer is listed as opaque, a server **must** still comply with the GetMap operation specified below: that is, the server must send an image with a transparent background if and only if the client requests TRANSPARENT=true and a picture FORMAT that supports transparency.

NOTE: Change from version 1.0.0: The opaque attribute is new.

7.1.5.5.4 Subsettable and resizable layers

The Layer metadata may also include three optional attributes that indicate a map server that is less functional than a normal WMS because it is not able to extract a subset of a larger dataset and/or because it only serves maps of a fixed size and cannot resize them. For example, a WMS that houses a collection of digitized images of historical maps, or pre-computed browse images of satellite data, may not be able to subset or resize those images. However, it can still respond to GetMap requests for complete maps in the original size.

Static image collections may not have a well-defined coordinate system, in which case the server **must** declare SRS=NONE as described in the Basic Service Elements section.

When set to a value of 1, **noSubsets** indicates that the Server is not able to crop the data or map to a geographic area smaller than its enclosing bounding box.

When present and nonzero, **fixedWidth** and **fixedHeight** indicate that the Server is not able to resize, resample or interpolate the data to a different pixel resolution than its "native" resolution.

NOTE: Change from version 1.0.0: The noSubsets, fixedWidth and fixedHeight attributes are new.

7.1.5.6 Inheritance of Layer Properties

Table 6 summarizes how the properties of an enclosing parent layer are inherited by child layers. Properties may be not inherited at all, or inherited but replaced if the child redefines them, or inherited and added to if the child also defines them. In the latter case, any duplicated definition by the child is ignorable.

Table 6 — Inheritance of Layer Properties

Element	Number	Inherit from parent?	Comments
Layer	0+	no	optional; no default; if present, each is a "child" of the enclosing Layer and inherits some default values as described in this table

Name	1	no	no default; required for all mapable layers.
Title	1	no	required for each Layer; no default
Abstract	0/1	no	recommended; no default
KeywordList	0/1	no	recommended; no default
Style	0+	add	optional; default from parent
SRS	1	add	list may include multiple whitespace-separated values
LatLonBoundingBox	1	replace	exactly one is required; default from parent
BoundingBox	0+	replace	one is required per available SRS other than EPSG:4326; default from parent
Dimension	0+	add	optional; default from parent
Extent	0+	replace	optional; default from parent; no more than one per declared Dimension
Attribution	0/1	replace	optional; default from parent
AuthorityURL	0+	add	optional; default from parent; exactly one unique AuthorityURL name per Identifier authority used.
Identifier	0+	no	optional; no default
MetadataURL	0+	no	optional; no default
DataURL	0/1	no	optional; no default
FeatureListURL	0/1	no	optional; no default
ScaleHint	0/1	replace	optional; default from parent
attributes listed in Table 5	1	replace	optional; default is "0" or parent value if present

7.1.5.7 Test Layer

To facilitate visual confirmation of a Web Map Server's ability to properly geo-register its output, WMS providers **should** supply a layer named 'WMS_GRATICULE'. The layer should be provided as a TRANSPARENT layer. The map should consist of a grid of lines that follows the 10 degree Latitude and Longitude lines on the Earth. The grid should cross 0,0 (i.e. the Equator at the Greenwich Meridian). The lines should be centered in the pixels. The preferred means of drawing this layer would be from a data source containing these line; barring that, the lines should be drawn using the same mechanism used to render data into other layers.

7.1.6 Output Formats

Format specifiers appear in several places in Capabilities XML: as valid output formats for an operation, as supported Exception formats, and as the format of content at external URLs. Formats are specified as MIME types such as "image/gif" or "image/png".

Several OGC-specific types have been defined to distinguish various types of XML documents; these are listed in Table 2, above.

NOTE: Change from WMS 1.0.0: Previously, formats were represented differently in the Capabilities XML. Formats were named using abbreviations (e.g., "GIF"), individual XML elements were defined for each format (e.g., <GIF/>, and known formats were enumerated in a %KnownFormats entity in the XML DTD. %KnownFormats no longer exists.

7.2 GetMap (required)

7.2.1 General

The GetMap operation is designed to produce a map, which is defined to be either a pictorial image or a set of graphical elements.

NOTE: Change from version 1.0.0: Previously, the GetMap request was capable of returning Feature data. That is now the role of an OGC Web Feature Service[7].

Upon receiving a Map request, a Map Server **must** either satisfy the request or throw an exception in the format requested.

7.2.2 GetMap Request Overview

Table 7 describes the Map Request. The request is typically encoded as a URL that is invoked on the WMS using the HTTP GET operation. (The optional Styled Layer Descriptor [10] extensions describe an HTTP POST encoding for communicating with an SLD-enabled WMS.)

Table 7 — The Parameters of a GetMap Request

Request Parameter	Required/ Optional	Description
VERSION=version	R	Request version.
REQUEST=GetMap	R	Request name.
LAYERS=layer_list	R	Comma-separated list of one or more map layers. Optional if SLD parameter is present.
STYLES=style_list	R	Comma-separated list of one rendering style per requested layer. Optional if SLD parameter is present.
SRS=namespace:identifier	R	Spatial Reference System.
BBOX=xmin,ymin,xmax,ymax	R	Bounding box corners (lower left, upper right) in SRS units.
WIDTH=output_width	R	Width in pixels of map picture.
HEIGHT=output_height	R	Height in pixels of map picture.
FORMAT=output_format	R	Output format of map.
TRANSPARENT=TRUE FALSE	O	Background transparency of map (default=FALSE).
BGCOLOR=color_value	O	Hexadecimal red-green-blue color value for the background color (default=0xFFFFFF).

Request Parameter	Required/ Optional	Description
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default=SE_XML).
TIME=time	O	Time value of layer desired.
ELEVATION=elevation	O	Elevation of layer desired.
Other sample dimension(s)	O	Value of other dimensions as appropriate.
Vendor-specific parameters	O	Optional experimental parameters.
The following parameters are used only with Web Map Services that support the Styled Layer Descriptor specification[10].		
SLD=styled_layer_descriptor_URL	O	URL of Styled Layer Descriptor (as defined in SLD Specification).
WFS=web_feature_service_URL	O	URL of Web Feature Service providing features to be symbolized using SLD.

7.2.3 Request Parameters

7.2.3.1 VERSION=version

This parameter is specified in the Basic Service Elements section.

In WMS version 1.0.0, the name of this parameter was "WMTVER". That name is now deprecated, but for backwards compatibility and version negotiation a post-1.0.0 server **should** accept either form without issuing a Service Exception.

NOTE: Change from version 1.0.0: This name of this parameter was previously "WMTVER".

7.2.3.2 REQUEST=GetMap

This nature of this parameter is specified in the Basic Service Elements section. For GetMap, the value "GetMap" must be used.

In WMS version 1.0.0, the value of this parameter was "map". That value is now deprecated, but for backwards compatibility a post-1.0.0 server **should** accept either form without issuing a Service Exception.

NOTE: Change from version 1.0.0: The value of this parameter was previously "map".

7.2.3.3 LAYERS=layer_list

The value of the LAYERS parameter is a comma-separated list of one or more valid layer names. Allowed layer names are the character data content of any <Layer><Name> element in the Capabilities XML.

A WMS **must** render the requested layers by drawing the leftmost in the list bottommost, the next one over that, and so on.

7.2.3.4 STYLES=style_list

The value of the STYLES parameter is a comma-separated list of one or more valid layer names. There is a one-to-one correspondence between the values in the LAYERS parameter and the values in the STYLES parameter. Each map in the list of LAYERS is drawn using the corresponding style in the same position in the list of STYLES. Each style Name **must** be one that was defined in the <Name> element of a <Style> element that is either directly contained within, or inherited by, the associated <Layer> element in Capabilities XML. (In other words, the Client may not request a Layer in a Style that was only defined for a different Layer.) A server **must** throw an exception if an unadvertised Style is requested.

A client **may** request the default Style using a null value (as in "STYLES="). If several layers are requested with a mixture of named and default styles, the STYLES parameter includes null values between commas (as in "STYLES=style1,,style2,,"). If all layers are to be shown using the default style, either the form "STYLES=" or "STYLES=,,," is valid.

7.2.3.5 SRS=namespace:identifier

The SRS (Spatial Reference System) parameter is specified in the Basic Service Elements section. The values of this request parameter **must** be one of those defined in the character data section of an <SRS> element defined or inherited by the requested layer. The same SRS applies to all layers in a single request.

If the WMS server has declared SRS=NONE for a Layer, as discussed in the Basic Service Elements section, then the Layer does not have a well-defined spatial reference system and should not be shown in conjunction with other layers. The Client **must** specify SRS=NONE (case-insensitive) in the GetMap request and the Server may issue a Service Exception otherwise.

7.2.3.6 BBOX=xmin,ymin,xmax,ymax

The BBOX parameter allows a Client to request a particular Bounding Box. Bounding Boxes are defined in the Basic Service Elements section.

If the WMS server has declared that a Layer is not subtable, as described in Section 7.1.5.3.3, "Subtable and Resizable Layers," then the Client **must** specify exactly the declared Bounding Box values in the GetMap request and the Server may issue a Service Exception otherwise.

7.2.3.7 WIDTH, HEIGHT

The WIDTH and HEIGHT parameters are expressed as positive integers. Taken together, they specify a rectangle of pixels into which the resulting map should be drawn and returned. The returned picture, regardless of its return format, **shall** have exactly that width and height in pixels. In the case where the aspect ratio of the BBOX and that of the WIDTH/HEIGHT are different, the WMS **must** stretch the returned map so that the

resulting pixels could themselves be rendered in the aspect ratio of the BBOX. In other words, it should be possible using this definition to request a map for a device whose output pixels are themselves non-square, or to stretch a map into an image area of a different aspect ratio.

If a request is for a display element return format (e.g., SVG or WebCGM), the WIDTH and HEIGHT values **must** be present and **may** be used by the server as helpful information but this specification currently considers this use to be experimental.

If the WMS server has declared that a Layer has fixed width and height, as described in Section 7.1.5.5.4 under "Subsettable and Resizable Layers," then the Client **must** specify exactly those WIDTH and HEIGHT values in the GetMap request and the Server may issue a Service Exception otherwise.

7.2.3.8 FORMAT=output_format

The nature of the FORMAT parameter is specified in the Basic Service Elements section. Supported values for a GetMap request on a WMS instance are listed in one or more <Format> elements in the <Request><GetMap> element of its Capabilities XML. The entire MIME type string in <Format> is used as the value of the FORMAT parameter. In an HTTP environment, the MIME type **must** be set on the returned object using the Content-type entity header.

NOTE: Change from version 1.0.0: Abbreviated names were previously used for formats (e.g., GIF instead of image/gif).

NOTE: Change from version 1.0.0: No feature formats (notably GML) are included, as these are now handled by the Web Feature Service.

7.2.3.9 TRANSPARENT=TRUE|FALSE

The TRANSPARENT parameter can take on two values, TRUE or FALSE. The ability to return pictures drawn with transparent pixels allows results of different Map requests to be overlaid, producing a composite map. It is **strongly recommended** that every WMS offer a format that provides transparency for layers which could sensibly be overlaid above others.

NOTE: At the time of this writing, the image/gif format provides transparency and is properly displayed by common web clients. The image/png format provides a transparency option but support in viewing applications is very poor. The image/jpeg format does not provide transparency at all.

When TRANSPARENT is set to TRUE and the FORMAT parameter contains a Picture format (e.g., image/gif), then a WMS **must** return (when permitted by the requested format) a result where all of the pixels not representing features or data values in that Layer are set to a transparent value. For example, a "roads" layer would be transparent wherever no road is shown. When TRANSPARENT is set to FALSE, those pixels **shall** be set to the value of BGCOLOR.

When the Layer has been declared as "opaque" as in Section 7.1.5.5.3, then significant portions, or the entirety, of the map may not be able to be made transparent.

When the FORMAT parameter contains a Graphic Element format, the TRANSPARENT parameter **may** be included in the request but its value **shall** be ignored by the WMS.

7.2.3.10 BGCOLOR=color_value

The general format of BGCOLOR is a hexadecimal encoding of an RGB value where two hexadecimal characters are used for each of Red, Green, and Blue color values. The values can range between 00 and FF for each (0 and 255, base 10). The format is 0xRRGGBB; either upper or lower case characters are allowed for RR, GG, and BB values. The "0x" prefix **shall** have a lower case 'x'.

When FORMAT is a Picture format, a WMS shall render its output on a background whose pixels were initially uniformly of the color encoded in BGCOLOR (or the default value 0xFFFFFFFF ["white"]) if the BGCOLOR parameter is not present in the request).

When FORMAT is a Graphic Element format (which does not have an explicit background), a WMS should avoid use of the BGCOLOR value for foreground elements because they would not be visible against a background picture of the same color.

When the Layer has been declared as "opaque" as in Section 7.1.5.5.3, then significant portions, or the entirety, of the map may not show any background at all.

7.2.3.11 EXCEPTIONS=exception_format

A Web Map Service can offer one or more of the following error reporting formats by listing them in separate <Format> elements inside the <Exceptions> element of its Capabilities XML. The entire MIME type string in <Format> is used as the value of the EXCEPTIONS parameter.

application/vnd.ogc.se_xml

Errors are reported using Service Exception XML, as specified in Section 6.7 and Annex A.3. This is the default exception format if none is specified in the request. The MIME type of the XML document containing the error message(s) **must** be application/vnd.ogc.se_xml.

NOTE: Change from version 1.0.0: XML is now the default Exceptions format, not the "inimage" option defined below.

NOTE: Change from version 1.0.0: This Exception format was previously called WMS_XML.

application/vnd.ogc.se_inimage

In the case of a Picture format, error messages are graphically returned as part of the content. This would usually take the form of text containing the message being painted into the returned map.

In Graphic Element output formats, the response to this value is experimental and no specific use is required or suggested by this specification.

application/vnd.ogc.se_inimage is a pseudo MIME type; the format of the returned content with embedded error messages is actually that given in the FORMAT parameter.

NOTE: Change from version 1.0.0: this Exception format was previously called INIMAGE.

NOTE: Change from version 1.0.0: this is no longer the default Exceptions format.

application/vnd.ogc.se_blank

In the case of a Picture format, if the EXCEPTIONS parameter is set to application/vnd.ogc.se_blank, the WMS shall, upon detecting an error, return an object of the type specified in FORMAT whose content is uniformly "off". In the case of an image format such as GIF or JPEG, that would be an object containing only pixels of one color (the background color if BACKGROUND is specified). In the case of a picture format supporting transparency, if TRANSPARENT=TRUE is specified the pixels shall all be transparent.

In Graphic Element output formats, such as vector-based formats, this specification suggests that no visible graphic elements be returned.

application/vnd.ogc.se_blank is a pseudo MIME type; the format of the returned content with embedded error messages is actually that given in the FORMAT parameter.

NOTE: Change from version 1.0.0: this Exception format was previously called BLANK.

7.2.3.12 TIME=time

The use of Time values is specified in Annexes B and C.

NOTE: Change from version 1.0.0: This is a new request parameter.

7.2.3.13 ELEVATION=elevation

The use of Elevation values is specified in Annex C.

NOTE: Change from version 1.0.0: This is a new request parameter.

7.2.3.14 Other sample dimensions

The declaration of sample dimensions is specified in Annex C.

NOTE: Change from version 1.0.0: Sample dimension parameters are new.

7.2.4 Vendor-Specific Parameters

The use of Vendor-Specific Parameters (VSPs) is discussed in the Basic Service Elements section. We repeat here only that clients **may** ignore any VSP they encounter in Capabilities XML, and that servers **must not** require the presence of VSPs in a request.

7.2.5 GetMap Response

The response to a valid GetMap request **must** be a map of the georeferenced information layer requested, in the desired style, and having the specified spatial reference system, bounding box, size, format and transparency.

An invalid GetMap request **must** yield an error output in the requested Exceptions format (or a network protocol error response in extreme cases).

In an HTTP environment, the MIME type of the returned value's Content-type entity header **must** match the format of the return value.

7.3 GetFeatureInfo (optional)

7.3.1 General

GetFeatureInfo is an optional operation. It is only supported for those Layers for which the attribute queryable="1" (true) has been defined or inherited (Section 7.1.5.5.1). A Client **must not** issue a GetFeatureInfo request for other layers. A WMS **should** respond with a properly formatted Service Exception (application/vnd.ogc.se_xml) response if it encounters that request but does not support it.

The GetFeatureInfo operation is designed to provide clients of a WMS with more information about features in the pictures of maps that were returned by previous Map requests. 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 what format the information should be returned in. 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.

The behavior described above is geared toward the picture case. In the graphic element case, the semantics of GetFeatureInfo are less defined. The intent is to gain experience with this version of the request and perhaps provide additional rigor in future versions of the specification.

The actual semantics of how a WMS decides what to return more information about, or what exactly to return is left up to the WMS provider.

7.3.2 GetFeatureInfo Request Overview

The parameters of a GetFeatureInfo request are listed in Table 8 below.

Table 8 — The Parameters of a GetFeatureInfo Request

Request Parameter	Required/ Optional	Description
VERSION=version	R	Request version.
REQUEST=GetFeatureInfo	R	Request name.
<map_request_copy>	R	Partial copy of the Map request parameters that generated the map for which information is desired.
QUERY_LAYERS=layer_list	R	Comma-separated list of one or more layers to be queried.
INFO_FORMAT=output_format	O	Return format of feature information (MIME type).
FEATURE_COUNT=number	O	Number of features about which to return information (default=1).
X=pixel_column	R	X coordinate in pixels of feature (measured from upper left corner=0)
Y=pixel_row	R	Y coordinate in pixels of feature (measured from upper left corner=0)
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default=application/vnd.ogc.se_xml).
Vendor-specific parameters	O	Optional experimental parameters.

7.3.3 Request Parameters

7.3.3.1 Prefix

The role of the URL prefix is specified in the Basic Service Elements section. The prefixes for GetCapabilities, GetMap and GetFeature info may be different.

7.3.3.2 VERSION

This parameter is specified in the Basic Service Elements section.

NOTE: Change from version 1.0.0: This name of this parameter was previously "WMTVER".

7.3.3.3 REQUEST

This nature of this parameter is specified in the Basic Service Elements section. For GetFeatureInfo, the value "GetFeatureInfo" must be used.

NOTE: Change from version 1.0.0: The value of this parameter was previously "feature_info".

7.3.3.4 map_request_copy

<map request copy> is not a name/value pair like the other parameters. Instead, most of the GetMap request parameters that generated the original map are repeated. Two are omitted because GetFeatureInfo provides its own values: VERSION and REQUEST. The

remainder of the GetMap request **must** be embedded contiguously in the GetFeatureInfo request.

7.3.3.5 QUERY_LAYERS

QUERY_LAYERS is a comma-separated list of one or more map layers from which feature information is desired to be retrieved. This parameter **must** contain at least one layer name, but **may** contain fewer layers than the original GetMap request.

If any layer in this list is not contained in the Capabilities XML of the WMS, the results are undefined and the WMS **must** produce an exception response.

7.3.3.6 INFO_FORMAT

The INFO_FORMAT indicates what format to use when returning the feature information. Supported values for a GetFeatureInfo request on a WMS instance are listed as MIME types in one or more <Format> elements inside the <Request><FeatureInfo> element of its Capabilities XML. The entire MIME type string in <Format> is used as the value of the INFO_FORMAT parameter. In an HTTP environment, the MIME type **must** be set on the returned object using the Content-type entity header.

EXAMPLE: The parameter INFO_FORMAT=application/vnd.ogc.gml requests that the feature information be formatted in Geography Markup Language (GML) [6].

NOTE: Change from version 1.0.0: The supported formats were previously given as GML (instead of the full MIME type) and as "MIME" (which meant the server could respond with any format as long as it announced its MIME type).

7.3.3.7 FEATURE_COUNT

FEATURE_COUNT is the maximum number of features for which feature information should be returned. Inclusion of a value for this attribute is optional. When included, the value **must** be greater than zero. When absent, the amount of information to return is at the server's discretion.

7.3.3.8 X and Y

The X and Y parameters shall identify a single point within the borders of the WIDTH and HEIGHT parameters of the embedded Map request. The origin is set to (0,0) centered in the pixel at the upper left corner; X increases to the right and Y increases downward.

7.3.3.9 EXCEPTIONS

Upon encountering an exception, a WMS GetFeatureInfo request must return an error message formatted as Service Exception XML.

NOTE: Change from version 1.0.0: GetFeatureInfo was previously permitted to return a MIME-typed error message of an unspecified nature.

7.3.4 GetFeatureInfo Response

The WMS shall return a response according to the requested INFO_FORMAT if the request is valid, or issue an exception otherwise. The nature of the response is at the discretion of the WMS provider, but it must pertain to the feature(s) nearest to (X,Y).

7.4 DescribeLayer (optional)

The **optional** DescribeLayer operation applies only to a Styled Layer Descriptor WMS. It allows the user to request an XML description of one or more map layers. See the SLD specification[10] for this operation.

Annex A (normative)

XML Document Type Definitions

A.1 WMS Capabilities DTD (Normative)

This annex contains the WMS Capabilities Document Type Definition corresponding to this version of the specification. The DTD may also be found on-line at http://www.digitalearth.gov/wmt/xml/capabilities_1_1_0.dtd. Comments in the DTD are informative; in case of conflict with the main body of this specification the main body takes precedence.

```

<!ELEMENT WMT_MS_Capabilities (Service, Capability) >

<!ATTLIST WMT_MS_Capabilities
            version CDATA #FIXED "1.1.0"
            updateSequence CDATA #IMPLIED>

<!-- Elements used in multiple places. -->

<!-- The Name is typically for machine-to-machine communication. -->
<!ELEMENT Name (#PCDATA) >

<!-- The Title is for informative display to a human. -->
<!ELEMENT Title (#PCDATA) >

<!-- The abstract is a longer narrative description of an object. -->
<!ELEMENT Abstract (#PCDATA) >

<!-- An OnlineResource is typically an HTTP URL. The URL is placed in the
xlink:href attribute. The xmlns:xlink attribute is a required XML namespace
declaration. -->
<!ELEMENT OnlineResource EMPTY>
<!ATTLIST OnlineResource
            xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink"
            xlink:type CDATA #FIXED "simple"
            xlink:href CDATA #REQUIRED >

<!-- A container for listing an available format's MIME type. -->
<!ELEMENT Format (#PCDATA) >

<!-- General service metadata. -->

<!ELEMENT Service (Name, Title, Abstract?, KeywordList?, OnlineResource,
                  ContactInformation?, Fees?, AccessConstraints?) >

<!-- List of keywords or keyword phrases to help catalog searching. -->
<!ELEMENT KeywordList (Keyword*) >

<!-- A single keyword or phrase. -->
<!ELEMENT Keyword (#PCDATA) >

<!-- Information about a contact person for the service. -->
<!ELEMENT ContactInformation (ContactPersonPrimary?, ContactPosition?,
                             ContactAddress?, ContactVoiceTelephone?,
                             ContactFacsimileTelephone?,
                             ContactElectronicMailAddress?) >

<!--The primary contact person.-->

```

```

<!ELEMENT ContactPersonPrimary (ContactPerson, ContactOrganization) >

<!--The person to contact.-->
<!ELEMENT ContactPerson (#PCDATA) >

<!--The organization supplying the service.-->
<!ELEMENT ContactOrganization (#PCDATA) >

<!--The position title for the contact person.-->
<!ELEMENT ContactPosition (#PCDATA) >

<!--The address for the contact supplying the service.-->
<!ELEMENT ContactAddress (AddressType,Address,City,StateOrProvince,PostCode,
    Country) >

<!--The type of address.-->
<!ELEMENT AddressType (#PCDATA) >

<!--The street address.-->
<!ELEMENT Address (#PCDATA) >

<!--The address city.-->
<!ELEMENT City (#PCDATA) >

<!--The state or province.-->
<!ELEMENT StateOrProvince (#PCDATA) >

<!--The zip or postal code.-->
<!ELEMENT PostCode (#PCDATA) >

<!--The address country.-->
<!ELEMENT Country (#PCDATA) >

<!--Contact telephone number.-->
<!ELEMENT ContactVoiceTelephone (#PCDATA) >

<!--The contact fax number.-->
<!ELEMENT ContactFacsimileTelephone (#PCDATA) >

<!--The e-mail address for the contact.-->
<!ELEMENT ContactElectronicMailAddress (#PCDATA) >

<!-- Elements indicating what fees or access constraints are imposed. -->
<!ELEMENT Fees (#PCDATA)>
<!ELEMENT AccessConstraints (#PCDATA)>

<!-- A Capability lists available request types, how exceptions
may be reported, and whether any vendor-specific capabilities are defined. It
also includes an optional list of map layers available from this server. -->
<!ELEMENT Capability
    (Request, Exception, VendorSpecificCapabilities?,
    UserDefinedSymbolization?, Layer?) >

<!-- Available WMS Operations are listed in a Request element. -->
<!ELEMENT Request (GetCapabilities, GetMap,
    GetFeatureInfo?, DescribeLayer?) >

<!-- For each operation offered by the server, list the available output
formats and the online resource. -->

<!ELEMENT GetCapabilities (Format+, DCPTType+)>

<!ELEMENT GetMap (Format+, DCPTType+)>

<!ELEMENT GetFeatureInfo (Format+, DCPTType+)>

<!ELEMENT DescribeLayer (Format+, DCPTType+)>

<!-- Available Distributed Computing Platforms (DCPs) are
listed here. At present, only HTTP is defined. -->
<!ELEMENT DCPTType (HTTP) >

```

```

<!-- Available HTTP request methods. One or both may be supported. -->
<!ELEMENT HTTP (Get | Post)+ >

<!-- URL prefix for each HTTP request method. -->
<!ELEMENT Get (OnlineResource) >
<!ELEMENT Post (OnlineResource) >

<!-- An Exception element indicates which error-reporting formats are supported. -->
<!ELEMENT Exception (Format+)>

<!-- Optional user-defined symbolization (used only by SLD-enabled WMSes). -->
<!ELEMENT UserDefinedSymbolization EMPTY >
<!ATTLIST UserDefinedSymbolization
    SupportSLD (0 | 1) "0"
    UserLayer (0 | 1) "0"
    UserStyle (0 | 1) "0"
    RemoteWFS (0 | 1) "0" >

<!-- Nested list of zero or more map Layers offered by this server. -->
<!ELEMENT Layer ( Name?, Title, Abstract?, KeywordList?, SRS?,
    LatLonBoundingBox?, BoundingBox*, Dimension*, Extent*,
    Attribution?, AuthorityURL*, Identifier*, MetadataURL*, DataURL*,
    FeatureListURL*, Style*, ScaleHint?, Layer* ) >

<!-- Optional attributes-->
<!ATTLIST Layer queryable (0 | 1) "0"
    cascaded CDATA #IMPLIED
    opaque (0 | 1) "0"
    noSubsets (0 | 1) "0"
    fixedWidth CDATA #IMPLIED
    fixedHeight CDATA #IMPLIED >

<!-- Listing of available Spatial Reference Systems (SRS). -->
<!ELEMENT SRS (#PCDATA) >

<!-- The LatLonBoundingBox attributes indicate the edges of the enclosing
rectangle in latitude/longitude decimal degrees (as in SRS EPSG:4326 [WGS1984
lat/lon]). -->
<!ELEMENT LatLonBoundingBox EMPTY>
<!ATTLIST LatLonBoundingBox
    minx CDATA #REQUIRED
    miny CDATA #REQUIRED
    maxx CDATA #REQUIRED
    maxy CDATA #REQUIRED>

<!-- The BoundingBox attributes indicate the edges of the bounding box
in units of the specified spatial reference system. -->
<!ELEMENT BoundingBox EMPTY>
<!ATTLIST BoundingBox
    SRS CDATA #REQUIRED
    minx CDATA #REQUIRED
    miny CDATA #REQUIRED
    maxx CDATA #REQUIRED
    maxy CDATA #REQUIRED
    resx CDATA #IMPLIED
    resy CDATA #IMPLIED>

<!-- The Dimension element declares the _existence_ of a dimension. -->
<!ELEMENT Dimension EMPTY >
<!ATTLIST Dimension
    name CDATA #REQUIRED
    units CDATA #REQUIRED
    unitSymbol CDATA #IMPLIED>

<!-- The Extent element indicates what _values_ along a dimension are valid. -->
<!ELEMENT Extent (#PCDATA) >
<!ATTLIST Extent
    name CDATA #REQUIRED
    default CDATA #IMPLIED>

<!-- Attribution indicates the provider of a Layer or collection of Layers.
The provider's URL, descriptive title string, and/or logo image URL may be
supplied. Client applications may choose to display one or more of these
items. A format element indicates the MIME type of the logo image located at

```

```

LogoURL. The logo image's width and height assist client applications in
laying out space to display the logo. -->
<!ELEMENT Attribution ( Title?, OnlineResource?, LogoURL? )>
<!ELEMENT LogoURL (Format, OnlineResource) >
<!ATTLIST LogoURL
    width NMTOKEN #REQUIRED
    height NMTOKEN #REQUIRED>

<!-- A Map Server may use zero or more MetadataURL elements to offer detailed,
standardized metadata about the data underneath a particular layer. The type
attribute indicates the standard to which the metadata complies. Two types
are defined at present: 'TC211' = ISO TC211 19115; 'FGDC' = FGDC CSDGM. The
format element indicates how the metadata is structured. -->
<!ELEMENT MetadataURL (Format, OnlineResource) >
<!ATTLIST MetadataURL
    type ( TC211 | FGDC ) #REQUIRED>

<!-- A Map Server may use zero or more Identifier elements to list ID numbers
or labels defined by a particular Authority. For example, the Global Change
Master Directory (gcmd.gsfc.nasa.gov) defines a DIF_ID label for every
dataset. The authority name and explanatory URL are defined in a separate
AuthorityURL element, which may be defined once and inherited by subsidiary
layers. Identifiers themselves are not inherited. -->

<!ELEMENT AuthorityURL (OnlineResource) >
<!ATTLIST AuthorityURL
    name NMTOKEN #REQUIRED >
<!ELEMENT Identifier (#PCDATA) >
<!ATTLIST Identifier
    authority CDATA #REQUIRED >

<!-- A Map Server may use DataURL to offer more information about the data
underneath a particular layer. While the semantics are not well-defined, as
long as the results of an HTTP GET request against the DataURL are properly
MIME-typed, Viewer Clients and Cascading Map Servers can make use of this. -->
<!ELEMENT DataURL (Format, OnlineResource) >

<!-- A Map Server may use FeatureListURL to point to a list of the features
represented in a Layer. -->
<!ELEMENT FeatureListURL (Format, OnlineResource) >

<!-- A Style element lists the name by which a style is requested and a
human-readable title for pick lists, optionally (and ideally) provides a
human-readable description, and optionally gives a style URL. -->
<!ELEMENT Style ( Name, Title, Abstract?,
    LegendURL*, StyleSheetURL?, StyleURL? ) >

<!-- A Map Server may use zero or more LegendURL elements to provide an
image(s) of a legend relevant to each Style of a Layer. The Format element
indicates the MIME type of the legend. Width and height attributes are
provided to assist client applications in laying out space to display the
legend. -->
<!ELEMENT LegendURL (Format, OnlineResource) >
<!ATTLIST LegendURL
    width NMTOKEN #REQUIRED
    height NMTOKEN #REQUIRED>

<!-- StyleSheetURL provides symbology information foreach Style of a Layer. -->
<!ELEMENT StyleSheetURL (Format, OnlineResource) >

<!-- A Map Server may use StyleURL to offer more information about the data or
symbology underlying a particular Style. While the semantics are not
well-defined, as long as the results of an HTTP GET request against the
StyleURL are properly MIME-typed, Viewer Clients and Cascading Map Servers can
make use of this. A possible use could be to allow a Map Server to provide
legend information. -->
<!ELEMENT StyleURL (Format, OnlineResource) >

<!-- Minimum and maximum scale hints for which it is appropriate to
display this layer. -->
<!ELEMENT ScaleHint EMPTY>
<!ATTLIST ScaleHint
    min CDATA #REQUIRED
    max CDATA #REQUIRED>

```

A.2 Sample WMS Capabilities XML (Informative)

As an aid to understanding and a guide for implementation, this Annex contains **example XML** which is valid according to the DTD in Annex A.1. Implementors should consult the main body of the specification document and the DTD to ensure compliance rather than editing this XML without full understanding, as the examples herein apply only to a limited range of cases. This is a copy of http://www.digitalearth.gov/wmt/xml/capabilities_1_1_0.xml.

```
<?xml version='1.0' encoding="UTF-8" standalone="no" ?>
<!-- The DTD (Document Type Definition) given here must correspond to the version number
declared in the WMT_MS_Capabilities element below. -->
<!DOCTYPE WMT_MS_Capabilities SYSTEM
"http://www.digitalearth.gov/wmt/xml/capabilities_1_1_0.dtd"
[
<!-- Vendor-specific elements are defined here if needed. -->
<!-- If not needed, you can omit these [] and everything inside. -->
<!ELEMENT VendorSpecificCapabilities EMPTY>
]> <!-- end of DOCTYPE declaration -->

<!-- Note: this XML is just an EXAMPLE that attempts to show all
required and optional elements for illustration. Consult the Web Map
Service 1.1.0 specification and the DTD for guidance on what to actually
include and what to leave out. -->

<!-- The version number listed in the WMT_MS_Capabilities element here must
correspond to the DTD declared above. See the WMT specification document for
how to respond when a client requests a version number not implemented by the
server. -->
<WMT_MS_Capabilities version="1.1.0" updateSequence="0">
<!-- Service Metadata -->
<Service>
  <!-- The WMT-defined name for this type of service -->
  <Name>GetMap</Name>
  <!-- Human-readable title for pick lists -->
  <Title>Acme Corp. Map Server</Title>
  <!-- Narrative description providing additional information -->
  <Abstract>WMT Map Server maintained by Acme Corporation. Contact:
webmaster@wmt.acme.com. High-quality maps showing roadrunner nests and possible ambush
locations.</Abstract>
  <KeywordList>
    <Keyword>bird</Keyword>
    <Keyword>roadrunner</Keyword>
    <Keyword>ambush</Keyword>
  </KeywordList>
  <!-- Top-level web address of service or service provider. See also OnlineResource
elements under <DCPType>. -->
  <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="http://hostname/" />
  <!-- Contact information -->
  <ContactInformation>
    <ContactPersonPrimary>
      <ContactPerson>Jeff deLaBeaujardiere</ContactPerson>
      <ContactOrganization>NASA</ContactOrganization>
    </ContactPersonPrimary>
    <ContactPosition>Computer Scientist</ContactPosition>
    <ContactAddress>
      <AddressType>postal</AddressType>
      <Address>NASA Goddard Space Flight Center, Code 933</Address>
      <City>Greenbelt</City>
      <StateOrProvince>MD</StateOrProvince>
      <PostCode>20771</PostCode>
      <Country>USA</Country>
    </ContactAddress>
    <ContactVoiceTelephone>+1 301 286-1569</ContactVoiceTelephone>
    <ContactFacsimileTelephone>+1 301 286-1777</ContactFacsimileTelephone>

    <ContactElectronicMailAddress>delabeau@iniki.gsfc.nasa.gov</ContactElectronicMailAddress>
  </ContactInformation>
```

```

    <!-- Fees or access constraints imposed. -->
    <Fees>none</Fees>
    <AccessConstraints>none</AccessConstraints>
</Service>
<Capability>
  <Request>
    <GetCapabilities>
      <Format>application/vnd.ogc.wms_xml</Format>
      <DCPType>
        <HTTP>
          <Get>
            <!-- The URL here for invoking GetCapabilities using HTTP GET
            is only a prefix to which a query string is appended. -->
            <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
              xlink:type="simple"
              xlink:href="http://hostname:port/path" />
          </Get>
          <Post>
            <!-- The URL here for invoking GetCapabilities using HTTP POST
            includes the complete address to which a query would be sent in
            XML format. This is here for future expansion; no POST encoding
            has yet been defined. -->
            <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
              xlink:type="simple"
              xlink:href="http://hostname:port/path" />
          </Post>
        </HTTP>
      </DCPType>
    </GetCapabilities>
    <GetMap>
      <Format>image/gif</Format>
      <Format>image/png</Format>
      <Format>image/jpeg</Format>
      <DCPType>
        <HTTP>
          <Get>
            <!-- The URL here for invoking GetCapabilities using HTTP GET
            is only a prefix to which a query string is appended. -->
            <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
              xlink:type="simple"
              xlink:href="http://hostname:port/path" />
          </Get>
        </HTTP>
      </DCPType>
    </GetMap>
    <GetFeatureInfo>
      <Format>application/vnd.ogc.gml</Format>
      <Format>text/plain</Format>
      <Format>text/html</Format>
      <DCPType>
        <HTTP>
          <Get>
            <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
              xlink:type="simple"
              xlink:href="http://hostname:port/path" />
          </Get>
        </HTTP>
      </DCPType>
    </GetFeatureInfo>
    <DescribeLayer><!--optional; used only by SLD-enabled WMS-->
      <Format>application/vnd.ogc.gml</Format>
      <DCPType>
        <HTTP>
          <Get>
            <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
              xlink:type="simple"
              xlink:href="http://hostname:port/path" />
          </Get>
        </HTTP>
      </DCPType>
    </DescribeLayer>
  </Request>
  <Exception>
    <Format>application/vnd.ogc.se_xml</Format>

```

```

    <Format>application/vnd.ogc.se_inimage</Format>
    <Format>application/vnd.ogc.se_blank</Format>
  </Exception>
  <!-- Any text or markup is allowed here, as required to describe
        vendor-specific capabilities. Please define elements and attributes
        in the DOCTYPE declaration at the start of the document. -->
  <!-- This example is empty because no VSPs were defined in preamble -->
  <VendorSpecificCapabilities />
  <!-- Place-holder for Styled Layer Descriptor (SLD)-enabled WMSes.
        This element is absent for a basic WMS. -->
  <UserDefinedSymbolization SupportSLD="1" UserLayer="1" UserStyle="1"
        RemoteWFS="1" />
  <Layer>
    <Title>Acme Corp. Map Server</Title>
    <SRS>EPSG:4326</SRS> <!-- all layers are available in at least this SRS -->
    <AuthorityURL name="DIF_ID">
      <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
        xlink:href="http://gcmd.gsfc.nasa.gov/difguide/whatisadif.html" />
    </AuthorityURL>
    <Layer>
      <!-- This parent layer has a Name and can therefore be requested from a Map Server,
            yielding a map of all subsidiary layers. -->
      <Name>ROADS_RIVERS</Name>
      <Title>Roads and Rivers</Title>
      <!-- See the spec to learn how some characteristics are inherited by
            subsidiary layers. -->
      <SRS>EPSG:26986</SRS> <!-- An additional SRS for this layer -->
      <LatLonBoundingBox minx="-71.63" miny="41.75" maxx="-70.78" maxy="42.90"/>
      <!-- The optional resx and resy attributes below indicate the X and Y spatial
            resolution in the units of that SRS. -->
      <!-- The EPSG:4326 BoundingBox duplicates some of the info in LatLonBoundingBox
            and is therefore optional, but using it here allows the additional
            resolution attributes. -->
      <BoundingBox SRS="EPSG:4326"
        minx="-71.63" miny="41.75" maxx="-70.78" maxy="42.90" resx="0.01" resy="0.01"/>
      <BoundingBox SRS="EPSG:26986"
        minx="189000" miny="834000" maxx="285000" maxy="962000" resx="1" resy="1" />
      <!-- Optional Title, URL and logo image of data provider. -->
      <Attribution>
        <Title>State College University</Title>
        <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
          xlink:href="http://www.university.edu/" />
        <LogoURL width="100" height="100">
          <Format>image/gif</Format>
          <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
            xlink:type="simple"
            xlink:href="http://www.university.edu/icons/logo.gif" />
        </LogoURL>
      </Attribution>
      <!-- Identifier whose meaning is defined in an AuthorityURL element -->
      <Identifier authority="DIF_ID">123456</Identifier>
      <FeatureListURL>
        <Format>application/vnd.ogc.se_xml</Format>
        <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
          xlink:href="http://www.university.edu/data/roads_rivers.gml" />
      </FeatureListURL>
      <Style>
        <Name>USGS</Name>
        <Title>USGS Topo Map Style</Title>
        <Abstract>Features are shown in a style like that used in USGS topographic
            maps.</Abstract>
        <!-- A picture of a legend for a Layer in this Style -->
        <LegendURL width="72" height="72">
          <Format>image/gif</Format>
          <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
            xlink:type="simple"
            xlink:href="http://www.university.edu/legends/usgs.gif" />
        </LegendURL>
        <!-- An XSL stylesheet describing how GML feature data will rendered to create
            a map of this layer. -->
        <StyleSheetURL>
          <Format>text/xsl</Format>
          <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
            xlink:type="simple"

```

```

        xlink:href="http://www.university.edu/stylesheets/usgs.xsl" />
    </StyleSheetURL>
</Style>
<ScaleHint min="4000" max="35000"></ScaleHint>
<Layer queryable="1">
    <Name>ROADS_1M</Name>
    <Title>Roads at 1:1M scale</Title>
    <Abstract>Roads at a scale of 1 to 1 million.</Abstract>
    <KeywordList>
        <Keyword>road</Keyword>
        <Keyword>transportation</Keyword>
        <Keyword>atlas</Keyword>
    </KeywordList>
    <Identifier authority="DIF_ID">123456</Identifier>
    <!-- Metadata specific to this particular layer. The same FGDC metadata is
offered in two formats. -->
    <MetadataURL type="FGDC">
        <Format>text/plain</Format>
        <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
            xlink:type="simple"
            xlink:href="http://www.university.edu/metadata/roads.txt" />
    </MetadataURL>
    <MetadataURL type="FGDC">
        <Format>text/xml</Format>
        <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
            xlink:type="simple"
            xlink:href="http://www.university.edu/metadata/roads.xml" />
    </MetadataURL>
    <!-- In addition to the Style specified in the parent Layer, this Layer is
available in this style. -->
    <Style>
        <Name>ATLAS</Name>
        <Title>Road atlas style</Title>
        <Abstract>Roads are shown in a style like that used in a commercial road
atlas.</Abstract>
        <LegendURL width="72" height="72">
            <Format>image/gif</Format>
            <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
                xlink:type="simple"
                xlink:href="http://www.university.edu/legends/atlas.gif" />
        </LegendURL>
    </Style>
</Layer>
<Layer queryable="1">
    <Name>RIVERS_1M</Name>
    <Title>Rivers at 1:1M scale</Title>
    <Abstract>Rivers at a scale of 1 to 1 million.</Abstract>
    <KeywordList>
        <Keyword>river</Keyword>
        <Keyword>canal</Keyword>
        <Keyword>waterway</Keyword>
    </KeywordList>
</Layer>
</Layer>
<Layer queryable="1">
    <Title>Weather Forecast Data</Title>
    <SRS>EPSG:4326</SRS> <!-- harmless repetition of common SRS -->
    <LatLonBoundingBox minx="-180" miny="-90" maxx="180" maxy="90" />
    <!-- These weather data are available daily from 1999-01-01 through
2000-08-22. -->
    <Dimension name="time" units="ISO8601" />
    <Extent name="time" default="2000-08-22">1999-01-01/2000-08-22/P1D</Extent>
    <Layer>
        <Name>Clouds</Name>
        <Title>Forecast cloud cover</Title>
    </Layer>
    <Layer>
        <Name>Temperature</Name>
        <Title>Forecast temperature</Title>
    </Layer>
    <Layer>
        <Name>Pressure</Name>
        <Title>Forecast barometric pressure</Title>
        <!-- Pressure is available at several elevations.

```

```

    EPSG:5030 is WGS 84 ellipsoid, units in metres.
    Pressure is also available at several times.
    NOTE: first list all Dimension elements, then all Extent elements. -->
    <Dimension name="time" units="ISO8601" />
    <Dimension name="elevation" units="EPSG:5030" />
    <Extent name="time" default="2000-08-22">1999-01-01/2000-08-22/P1D</Extent>
    <Extent name="elevation" default="0">0,1000,3000,5000,10000</Extent>
  </Layer>
</Layer>
<!-- Example of a layer which is a static map of fixed
      size which the server cannot subset or make transparent -->
<Layer opaque="1" noSubsets="1" fixedWidth="512" fixedHeight="256">
  <Name>ozone_image</Name>
  <Title>Global ozone distribution (1992)</Title>
  <LatLonBoundingBox minx="-180" miny="-90" maxx="180" maxy="90" />
  <Extent name="time" default="1992">1992</Extent>
</Layer>
<!-- Example of a layer which originated from another WMS and has been
      "cascaded" by this WMS. -->
<Layer cascaded="1">
  <Name>population</Name>
  <Title>World population, annual</Title>
  <LatLonBoundingBox minx="-180" miny="-90" maxx="180" maxy="90" />
  <Extent name="time" default="2000">1990/2000/P1Y</Extent>
</Layer>
</Layer>
</Capability>
</WMT_MS_Capabilities>

```

A.3 Service Exception DTD (Normative)

This annex contains the Service Exception Document Type Definition corresponding to this version of the specification. The DTD may also be found on-line at http://www.digitalearth.gov/wmt/xml/exception_1_1_0.dtd.

```

<!ELEMENT ServiceExceptionReport (ServiceException*)>
<!ATTLIST ServiceExceptionReport version CDATA #FIXED "1.1.0">

<!ELEMENT ServiceException (#PCDATA)>
<!ATTLIST ServiceException code CDATA #IMPLIED>

```

A.4 Sample Service Exception XML (Informative)

As an aid to understanding and a guide for implementation, this Annex contains **example** XML which is valid according to the DTD in Annex A.3. Implementors should consult the main body of the specification document and the DTD to ensure compliance rather than editing this XML without full understanding. This is a copy of http://www.digitalearth.gov/wmt/xml/exception_1_1_0.xml.

```

<?xml version='1.0' encoding="UTF-8" standalone="no" ?>
<!DOCTYPE ServiceExceptionReport SYSTEM
"http://www.digitalearth.gov/wmt/xml/exception_1_1_0.dtd">
<ServiceExceptionReport version="1.1.0">
<ServiceException>
Plain text message about an error.
</ServiceException>
<ServiceException code="InvalidUpdateSequence">
Another error message, this one with a SE code supplied.
</ServiceException>
<ServiceException>
<![CDATA[
Error in module <foo.c>, line 42

```

A message that includes angle brackets in text must be enclosed in a Character Data Section

as in this example. All XML-like markup is ignored except for this sequence of three closing characters:

```
]]>
</ServiceException>
<ServiceException>
<![CDATA[
<Module>foo.c</Module>
<Error>An error occurred</Error>
<Explanation>Similarly, actual XML
can be enclosed in a CDATA section.
A generic parser will ignore that XML,
but application-specific software may choose
to process it.</Explanation>
]]>
</ServiceException>
</ServiceExceptionReport>
```

Annex B (normative)

Formatting Dates and Times

B.1 Overview

This annex describes extensions to the ISO 8601 specification for denoting moments and periods in time. These allow OGC Web Services to support temporal data descriptions and requests. This specification is based on ISO 8601:1988(E) Date/Time Representations [24] for date+time and for periods of time; it also extends ISO 8601 to allow years before 1 AD.

This Annex specifies the formats for a moment in time, including years in the current era ("AD") and years before the current era or on geologic timescales, the format for a period of time, and the format for incomplete data fragments.

B.2 Time Format Details

B.2.1 Basic Syntax

The basic time format uses ISO 8601 "extended" format: up to 14 digits specifying century, year, month, day, hour, minute, seconds, and seconds, and optionally a decimal point followed by zero or more digits for fractional seconds, with non-numeric characters to separate each piece:

`ccyy-mm-ddThh:mm:ss.sssZ`

(ISO 8601 prefers a decimal comma but allows a decimal period.) All times are expressed in Coordinated Universal Time (UTC) by default, so the trailing Z (for "zulu") is optional but recommended.

The precision may be reduced by omitting least-significant digits, as in these examples.

EXAMPLE 1: `ccyy` Year only

EXAMPLE 2: `ccyy-mm` Year and month

EXAMPLE 3: `ccyy-mm-dd` Year, month and day

EXAMPLE 4: `ccyy-mm-ddThhZ` Year, month, day and hour (with Z indicator for UTC)

B.2.2 Extension for years B.C.E.

A non-ISO extension is used to handle years B.C.E. (before year 0001 AD). Because a leading hyphen has meaning in ISO 8601 (it can indicate missing parts of the time), preceding times with a minus sign is not advisable. Instead, the prefix 'B' is used to indicate B.C.E. years.

EXAMPLE: B1350 (1350 "BC", approximate time of Tutankhamen).

B.2.3 Extension for geologic datasets

A non-ISO extension is used to handle geologic datasets referring to the distant past. Geologists refer to years some number of thousand, million or billion (10^9) years ago as ka, Ma or Ga. Therefore, the prefixes K, M and G, followed by an integer or floating-point number of years, are used.

EXAMPLE 1: M150 (Jurassic period, 150 Ma)

EXAMPLE 2: K150000 (Jurassic period, 150 Ma)

EXAMPLE 3: K18 (last Ice Age, 18 ka)

EXAMPLE 4: M0.018 (last Ice Age, 18 ka)

EXAMPLE 5: G5 (Earth's crust solidifies, 5 Ga)

B.3 Period Format

A Period indicates the time resolution of the available data. The ISO 8601 specification for representing a duration of time is used: Designator P (for Period), number of years Y, months M, days D, time designator T, number of hours H, minutes M, seconds S. Unneeded elements may be omitted.

EXAMPLE 1: P1Y -- 1 year

EXAMPLE 2: P1M10D -- 1 month plus 10 days

EXAMPLE 3: PT2H -- 2 hours

EXAMPLE 4: PT1.5S -- 1.5 seconds

B.4 Time Lists and Ranges

As shown in Table C.1, a list of several times is expressed by separating valid time values with a comma (","). A temporal range is expressed using the syntax start/end/period, indicating the start time of the data, the ending time, and the time resolution or refresh rate. Data which are refreshed at regular intervals but not archived are represented using time/time/period, where both time values are the same (that of the latest data) and the period indicates the refresh rate.

B.5 Date Fragments

In some contexts, date information may be incomplete. For example, a GeoParser service that scans text documents for references to locations or times may encounter partial references such as "Tuesday" or "August 30." To allow this time specification to handle such fragments, we include the "truncated representation" allowed by ISO 8601, and we add 7 terms to represent days of the week.

B.5.1 Truncated representation

The "truncated representation" allowed by ISO 8601 uses hyphens to denote pieces missing from the most significant end of the time string. Add one hyphen for each missing two-digit century, year, month, day, hour or minute. See examples in B.6.2, below.

B.5.2 Days of the week

We add 7 terms to represent days of the week: 'MON', 'TUE', 'WED', 'THU', 'FRI', 'SAT', 'SUN'. These optional terms appear at the start of the date/time string, and a comma separates the weekday (without whitespace) from the ISO 8601 string. If the date fragment contains only a day of week, then the comma is not necessary. See examples in B.6.2, below.

B.6 Examples

B.6.1 Complete dates and times

A single moment (scalar value):

2000-06-23T20:07:48.11Z

Quarterly data (comma-separated list):

1999-01-01,1999-04-01,1999-07-01,1999-10-01

Daily data taken at noon since April 15 1995 (periodic interval):

1995-04-22T12:00Z/2000-06-21T12:00Z/P1D

Current data refreshed every 30 min (periodic interval):

2000-06-18T14:30Z/2000-06-18T14:30Z/PT30M

B.6.2 Date and time fragments

--11-21 (Nov. 21, truncated century and year, omitted hh:mm:ss)

TUE,--11-21 (Tuesday, Nov. 21)

TUE (Tuesday)

2000 (year 2000)

-99 (year 99, century not given)

T14 (2pm local time)

MON,---21T14 (Monday the 21st at 2pm)

T14Z (2pm UTC)

T14-04:00 (2pm EDT, 4hrs earlier than UTC)

T-10 (ten minutes past an unspecified hour)

Annex C (normative)

Handling Multi-Dimensional Data

C.1 Overview

This annex describes support for multi-dimensional data in the service metadata and operation request parameters of OGC Web Services. "Dimensions" in this Annex refers to Time, Elevation, and sample dimensions.

EXAMPLE: Satellite imagery may be available at several different times and several different wavelength bands. The valid times and band numbers can each be separately specified.

C.2 Declaring Dimensions

The optional element <Dimension> is used in Capabilities XML to declare that one or more dimensional parameters are relevant to the information holdings of that server. The Dimension element does not provide valid values for a Dimension; that is the role of the Extent element described below. A Dimension element includes a **required** name, a **required** measurement units specifier, and an **optional** unitSymbol.

The following is a DTD fragment expression for the Dimension element (from the full DTD in Annex A.1):

```
<!Element Dimension EMPTY>
<!ATTLIST Dimension
name          ID #REQUIRED
units         CDATA #REQUIRED
unitSymbol    CDATA #IMPLIED>
```

EXAMPLES: The following are some examples of server-defined dimensions:

```
<Dimension name="wavelength" units="Angstrom" unitSymbol="Ao" />
<Dimension name="temperature" units="Kelvin" unitSymbol="K" />
<Dimension name="pressure_scale_height" units="millibar" unitSymbol="mbar" />
<Dimension name="altitude" units="kilometer" unitSymbol="km" />
```

Dimension elements must comply with the following rules:

- Dimension name values **must** be case-insensitively unique within a service instance and **should** contain no whitespace.
- Unit names and abbreviations are from the Unified Code for Units of Measure (UCUM) <<http://aurora.rg.iupui.edu/~shadow/units/UCUM/>>. The **required** units attribute is from the UCUM "name" column. The optional (but **recommended**)

unitSymbol is from the UCUM "c/s" column (case-sensitive abbreviation). Prefix names and symbols **may** be present as well.

- If the dimensional quantity has no units (e.g., band number in a multi-wavelength sensor), use the null string: `units=""`.

All Dimensions in a Capabilities XML response are server-defined, with two exceptions. The dimensions named time and elevation are privileged special cases, predefined as follows:

```
<Dimension name="time" units="ISO8601" />
<Dimension name="elevation" units="EPSG:vertical_datum" />
```

The units "ISO8601" refers to the Time representation specified in Annex B. The units "EPSG:vertical_datum" refers to elevation units and a reference used by one of the European Petroleum Survey Group [ref. 13] vertical datums (use the actual datum number in place of the string "vertical_datum").

EXAMPLE: Elevation units given as "EPSG:5030" means "meters above the WGS84 ellipsoid."

Although predefined, a server **must** include Time and/or Elevation declarations if it provides extents for those dimensions.

If a server opts to specify elevation or time values in reference systems other than those listed above, it **must** use other dimension names instead of "elevation" and "time" (e.g., "altitude" and "date").

Dimension declarations are inherited by enclosed Layers, as specified in Section 7.1.5.6. A dimension **must not** be re-declared using the same name (case-insensitively) with conflicting information.

C.3 Specifying Dimensional Extents

Having declared the existence of a Dimension, the Capabilities XML response uses corresponding Extent elements to specify the bounds of a geodata object along zero or more independent single dimensions.

Format of Extent element:

```
<Extent name="dimension_name" default="default_value"
  multipleValues="0|1" nearestValue="0|1">
  extent_value</Extent>
```

Extent values:

The extent_value string declares what value(s) along the Dimension axis are appropriate for this specific geospatial data object. The extent_value has the syntax shown in Table C.1.

Table C.1 — Syntax for Listing One or More Dimensional Values

Syntax	Meaning
value	A single value
value1,value2,value3,... ^a	A list of multiple values
min/max/resolution	An interval defined by its lower and upper bounds and its resolution.
min1/max1/res1,min2/max2/res2,... ^a	A list of multiple intervals
^a Whitespace is allowed following commas in a list in an <Extent> element.	

Extent attributes:

Several attributes are defined for the Extent element. Only name is **required**.

- name (required): The value dimension_name is the same as the value of a name attribute of a <Dimension> element previously defined as described in Section C.2. The Dimension name and Extent name **must** match (case-insensitively).
- default (optional): The default_value declares what value would be used along that Dimension if a Web request omits a value for that Dimension. See Section C.5
- multipleValues (optional): This attribute, if absent or zero, indicates that requests **must** include only a single value for this dimension. If present and nonzero, then requests **may** include multiple values. See Section C.4.3.
- nearestValues (optional): This attribute, if absent or zero, indicates that the server will not round off inexact dimension values to the nearest valid value. If present and nonzero, then the server will perform rounding. See Section C.5.
- current (optional; only valid for Time extents): This attribute, if present and nonzero, indicates (1) that temporal data are normally kept current and (2) that the request parameter TIME **may** include the keyword 'current' instead of an ending value (see Section C.4.1).

Extent declarations are inherited by enclosed Layers, as specified in Section 7.1.5.6. A extent that is re-specified in an enclosed layer using the same name (case-insensitively) replaces the extent that had been inherited.

A resolution value of zero (as in min/max/0) means that the data are effectively at infinitely-fine resolution for the purposes of making requests on the service instance. For instance, an instrument which continuously monitors randomly-occurring data may have no explicitly defined temporal resolution.

EXAMPLE:

Here is a <Layer> element with Extents specified in WMS Capabilities XML ('...' indicates XML omitted for clarity):

```
<Layer>
  <!-- Declare dimensions in use. Declarations are inherited by enclosed Layers. -->
  <Dimension name="time" units="ISO8601" />
  <Dimension name="temperature" units="Kelvin" unitSymbol="K" />
  <Dimension name="elevation" units="EPSG:5030" />
  ...
  <Layer> ...
  <!-- Specify extent of Layer. Extents are inherited by enclosed Layers. -->
  <Extent name="time" default="2000-10-17">1996-01-01/2000-10-17/P1D</Extent>
  <Extent name="elevation" default="0">0/10000/100</Extent>
  <Extent name="temperature" default="300">230,300,400</Extent>
</Layer>
...
</Layer>
```

C.4 Including Dimensional Values in a Request

Requests against multi-dimensional data objects described with <Dimension> and <Extent> require additional parameters to formulate a complete query. This following section specifies how Clients should include dimensional parameters in those operations which support such parameters.

C.4.1 Elevation and Time Values in Requests

If a data object has an Elevation extent defined, then operation requests to retrieve that object may include

ELEVATION=*value*

If a Layer has a Time extent defined, then requests may include

TIME=*value*

In either case, *value* uses the format described in Table C.1 to provide a single value, a comma-separated list, or an interval of the form start/end without a resolution. *value* **must not** contain whitespace. An interval in a request value is a request for all the data from the start value up to and including the end value.

The absence of Time or Elevation parameters is equivalent to a request for the Layer's default value (if defined) along that dimension; see Section C.5. All parameter names are

case-insensitive as stated in Section 6.4.1, so, for example, 'TIME', 'Time' and 'time' are all acceptable.

For the TIME parameter, the special keyword 'current' **may** be used if advertised by the server as in Section C.3. The expression "TIME=current" means "send the most current data available." The expression "TIME=start_time/current" means "send data from start_time up to the most current data available."

C.4.2 Sample Dimension Values in Requests

Sample dimension parameters allow the Client to request a particular data object along one or more additional dimensional axes.

A request parameter name is constructed by concatenating the prefix 'dim_' with the sample dimension Name (the value of the name attribute of the corresponding <Dimension> and <Extent> elements in the Capabilities XML). The resulting "dim_name" is case-insensitive. The use of the 'dim_' prefix is to avoid clashes between server-defined dimension names and current or future OGC Web Service specifications. (Time and Elevation, being predefined, do not use the prefix.)

To include a sample dimension value in a request URL, the dim_name parameter is followed an equals sign '=' and a valid value, a comma-separated list, or an interval, as described in Table C.1.

Requests **may** omit a dimension parameter, or use null values, to request the default value (if supplied) along that dimension; see Section C.5.

EXAMPLE: A WMS Layer is described as having an Extent along a Dimension named "wavelength" with values 3000,4000,5000,6000 (Angstroms). WMS GetMap requests for that Layer may include the parameter "DIM_WAVELENGTH=4000".

C.4.3 Single- and Multiple-Valued Requests

Whether a request for a georeferenced object may include only a single value for each dimension or multiple values depends on the context. Multiple values are only acceptable when both the output format and the nature of the geospatial information permits it. An server **may** include a nonzero multipleValue attribute (Section C.2) in an Extent element to indicate that multi-valued requests are supported. A server **must** issue a Service Exception if it cannot comply with the dimensional parameters as specified.

EXAMPLE 1: A Web Map Service offers maps in PNG image format of vehicle traffic density updated hourly. A valid GetMap request would include only a single TIME to select the desired map.

EXAMPLE 2: A WMS offers daily ozone maps in QuickTime movie format. The movie can contain multiple frames, so a valid GetMap request could include multiple times.

EXAMPLE 3: A WMS offers maps showing crime locations. Each crime has an associated date. A GetMap image request for all crimes occurring within a defined period would include a TIME range.

EXAMPLE 4: A Web Coverage Service offers atmospheric pollutant data at multiple elevations. A GetCoverage request could include a single elevation to obtain only a slice through the data, or multiple elevations to obtain a three-dimensional slab of data.

C.4.4 Applicability to Multiple Data Objects

Some OGC Web Service requests allow multiple georeferenced objects to be requested at once. For example, the WMS GetMap request includes a comma-separated list of one or more Layers. In such a request, the TIME, ELEVATION and DIM_ parameters apply to all of objects individually as follows:

- If the georeferenced object has the corresponding dimension as a property, then the dimension value applies as if that object had been requested alone.
- If the georeferenced object does not have that dimension, then the dimension value is ignored for the purposes of responding to the request for that object.

EXAMPLE: A Client's WMS GetMap request includes "LAYERS=temperature,coastlines&TIME=20010509" where 'temperature' is a daily temperature map (for which time is relevant) and 'coastlines' is a coastline map (which is static). The WMS responds with the static coastlines drawn atop the temperature for the specified date, without complaining that TIME does not apply to coastlines. If the date is invalid for the temperature layer, then the server may issue a Service Exception to that effect.

C.4.5 Example requests

The following are examples of multi-dimensional GetMap and GetCoverage requests.

EXAMPLE 1: WMS GetMap (single ozone Map at specified time and height):

```
VERSION=x.y.z WIDTH=600
REQUEST=map HEIGHT=300
LAYERS=ozone TIME=2000-08-03
SRS=EPSG:4326 ELEVATION=1000
BBOX=-180,-90,180,90 FORMAT=image/gif
```

EXAMPLE 2: WMS GetMap (movie loop at specified frame times):

```
VERSION=x.y.z WIDTH=600
REQUEST=map HEIGHT=300
LAYERS=ozone TIME=2000-07-01/2000-07-31/P1D
SRS=EPSG:4326 ELEVATION=1000
BBOX=-180,-90,180,90 FORMAT=video/mpeg
```

EXAMPLE 3: WCS GetCoverage (4D block of data with dimension x,y,z,t):

```
VERSION=x.y.z SKIPX=10
REQUEST=coverage SKIPY=10
LAYERS=layer TIME=2000-07-01/2000-07-31/P1D
SRS=srs_identifier ELEVATION=0/10000/1000
BBOX=minx,miny,maxx,maxy FORMAT=application/x-hdf
```

C.5 Server Responses

C.5.1 Incorrect Values

If the dimension value is invalid or missing from the client request, and no default or nearest-value behavior was enabled as discussed in the following subsections, then the server **must** respond with a Service Exception (code="InvalidDimensionValue").

C.5.1 Default Values

An OGC Web Service **may** declare a default for any dimension. A 'default' is **optional** but **recommended**. If a request does not include a value along this dimension, and a default has been declared, then the server **must** send the default value. If there is no declared default, then the server **must** throw an Exception (code="MissingDimensionValue") to indicate that a value was required.

In order that the Client may determine what value was actually sent, the server **must** label the response object with the actual value used by default. In the HTTP environment, the labeling is performed using an HTTP response header. For each dimension to which a default value was applied, a header line of the following form **must** be sent:

```
Warning: 99 Default value used: DIM_NAME=value units
```

where "99" is a defined by HTTP[8] for use by miscellaneous warnings, DIM_NAME is the corresponding request parameter name, value is the value actually used, and units is the units attribute (see C.2) for that dimension.

C.5.2 Nearest Values

An OGC Web Service **may** declare that it will choose the closest available value for a dimension if an exact value is not specified. This allows, for example, hourly data whose actual recording time is precise to the millisecond to be requested simply by stating the desired date and hour. The nearestValue attribute of the <Extent> element, if present and nonzero, indicates that this behavior is enabled.

If a request does includes an imprecise dimensional value, and nearest value behavior has been declared, then the server **must** compute and send the nearest available value. The value must be rounded, not merely truncated. If rounding is not supported, then the server **must** throw an Exception (code="InvalidDimensionValue") to indicate that a value was required.

In order that the Client may determine what value was actually sent, the server **must** label the response object with the actual value found by rounding. In the HTTP environment, the labeling is performed using an HTTP response header. For each dimension to which a rounded-off value was applied, a header line of the following form **must** be sent:

```
Warning: 99 Nearest value used: DIM_NAME=value units
```

where "99" is defined by HTTP[8] for use by miscellaneous warnings, DIM_NAME is the corresponding request parameter name, value is the value actually used, and units is the units attribute (see C.2) for that dimension.

Annex D (normative)

Conformance Tests

Minimal conformance with this specification requires the following:

1. The GetCapabilities and GetMap operations **must** be supported.
2. The Extensible Markup Language (XML) document returned in response to a GetCapabilities request **must** be valid against the Document Type Definition in Annex A.1. Such validation may be performed using commonly available XML validating tools.
3. The maps returned in response to a valid GetMap request **must** be accurately registered according to the requested projection and bounding box.
4. All clauses in the normative sections of this specification that use the keywords "must", "must not", "required", "shall", and "shall not" have been satisfied.

Bibliography

- [1] OpenGIS Project Document 97-009: "WWW Mapping Framework", Allan Doyle, BBN Corporation.
- [2] OpenGIS Project Document 98-060: "User Interaction with Geospatial data", Adrian Cuthbert, Laser-Scan.
- [3] OpenGIS Project Document 98-068: "A Web Mapping Scenario", Kenn Gardels, University of California, Berkeley.
- [4] OpenGIS "Request for Technology In Support of a Web Mapping Technology Testbed", October 1998.
- [5] OpenGIS "Request For Quotation and Call For Participation in the OGC Web Mapping Testbed Initial Operating Capability and Demonstration", March 1999.
- [6] Cox, S., Cuthbert, A., Lake, R., and Martell, R. (eds.), "OpenGIS Recommendation - Geography Markup Language 2.0," February 2000, <http://www.opengis.org/techno/specs/>.
- [7] Vretanos, P. (ed.), "OpenGIS Discussion Paper #01-023: Web Feature Service Draft Candidate Implementation Specification 0.0.12," January 2001, <http://www.opengis.org/techno/discussions.htm>.
- [8] Fielding et. al., "Hypertext Transfer Protocol – HTTP/1.1," IETF RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>.
- [9] Bray, Paoli, Sperberg-McQueen, eds., "Extensible Markup Language (XML) 1.0", 2nd edition, October 2000, W3C Recommendation, <http://www.w3.org/TR/2000/REC-xml>.
- [10] Cuthbert, A. (ed.), "OpenGIS Discussion Paper #01-028: Styled Layer Descriptor Draft Candidate Implementation Specification 0.7.0," February 2001, <http://www.opengis.org/techno/discussions.htm>.
- [11] Freed, N. and Borenstein N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", IETF RFC 2045, November 1996, <http://www.ietf.org/rfc/rfc2045.txt>.
- [12] Berners-Lee, T., Fielding, N., and Masinter, L., "Uniform Resource Identifiers (URI): Generic Syntax", IETF RFC 2396, <http://www.ietf.org/rfc/rfc2396.txt>.
- [13] European Petroleum Survey Group, "EPSG Geodesy Parameters", <http://www.epsg.org/>.

- [14] Web Mapping Testbed, "WM Automatic Projections",
<http://www.digitalearth.gov/wmt/auto.html>.
- [15] Bradner, Scott, "RFC 2119 Key words for use in RFCs to Indicate Requirement Levels," March 1997, <ftp://ftp.isi.edu/in-notes/rfc2119.txt> .
- [16] National Center for Supercomputing Applications, "The Common Gateway Interface," <http://hoohoo.ncsa.uiuc.edu/cgi/>.
- [17] Doyle, A. (ed.), "OpenGIS Web Map Service Interface Specification 1.0.0," April 2000, <http://www.opengis.org/techno/specs/>.
- [18] Internet Assigned Numbers Authority, <http://www.isi.edu/in-notes/iana/assignments/media-types/>.