Open Geospatial Consortium Inc.

Date: 2006-03-15

Reference number of this document: OGC® 06-042

Version: 1.3.0

Category: OpenGIS® Implementation Specification

Editor: Jeff de la Beaujardiere

OpenGIS® Web Map Server Implementation Specification

Document type: OpenGIS® Implementation Specification
Document subtype: Not Applicable
Document stage: Final
Document language: English

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable

Contents

Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. OGC shall not be held responsible for identifying any or all such patent rights.

This version supercedes all previous versions of OpenGIS® Web Map Service Implementation Specification, including OGC 04-024, 01-068r3, 01-047r2, 01-109r1 and OGC 00-028.

Introduction

A Web Map Service (WMS) produces maps of spatially referenced data dynamically from geographic information. This International Standard defines a "map" to be a portrayal of geographic information as a digital image file suitable for display on a computer screen. A map is not the data itself. WMS-produced 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 International Standard defines three operations: one returns service-level metadata; another returns a map whose geographic and dimensional parameters are well-defined; and an optional third operation returns information about particular features shown on a map. Web Map Service operations can be invoked using a standard web browser by submitting requests in the form of Uniform Resource Locators (URLs). The content of such URLs depends on which operation is requested. In particular, when requesting a map the URL indicates what information is to be shown on the map, what portion of the Earth is to be mapped, the desired coordinate reference system, and the output image width and height. When two or more maps are produced with the same geographic parameters and output size, the results can be accurately overlaid to produce a composite map. The use of image formats that support transparent backgrounds (e.g. GIF or PNG) allows underlying maps to be visible. Furthermore, individual maps can be requested from different servers. The Web Map Service thus enables the creation of a network of distributed map servers from which clients can build customized maps. Illustrative examples of map request URLs and their resulting maps are shown in Annex G.

This International Standard applies to a Web Map Service instance that publishes its ability to produce maps rather than its ability to access specific data holdings. A basic WMS classifies its geographic information holdings into "Layers" and offers a finite number of predefined "Styles" in which to display those layers. This International Standard supports only named Layers and Styles, and does not include a mechanism for user-defined symbolization of feature data.

NOTE   The Open Geospatial Consortium (OGC) Styled Layer Descriptor (SLD) specification [6] defines a mechanism for user-defined symbolization of feature data instead of named Layers and Styles. In brief, an SLD-enabled WMS retrieves feature data from a Web Feature Service [7] and applies explicit styling information provided by the user in order to render a map.

Geographic information — Web map server interface

1 Scope

This International Standard specifies the behaviour of a service that produces spatially referenced maps dynamically from geographic information. It specifies operations to retrieve a description of the maps offered by a server to retrieve a map, and to query a server about features displayed on a map. This International Standard is applicable to pictorial renderings of maps in a graphical format; it is not applicable to retrieval of actual feature data or coverage data values.

2 Conformance

2.1 Conformance classes and requirements

This International Standard defines two conformance classes, one for a basic WMS, and the other for a queryable WMS. Each has two subclasses, one for clients and the other for servers.

2.2 Basic WMS

A basic WMS shall support the basic service elements (see Clause 6), the GetCapabilities operation (see 7.2), and the GetMap operation (see 7.3). To conform to this International Standard, a basic WMS shall satisfy the requirements of A.1 of the Abstract Test Suite in Annex A.

2.3 Queryable WMS

A queryable WMS shall satisfy all the requirements for a basic WMS, and shall also support the GetFeatureInfo operation (see 7.4). To conform to this International Standard, a queryable WMS shall satisfy all requirements of the Abstract Test Suite in Annex A.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 8601:2004, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO 19111, *Geographic information — Spatial referencing by coordinates*

ISO 19115:2003, *Geographic information — Metadata*

EPSG (February 2003), *European Petroleum Survey Group Geodesy Parameters*, Lott, R., Ravanas, B., Cain, J., Simonson, G, and Nicolai, R., eds., available at *<http://www.epsg.org/>*

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

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

IETF RFC 2616 (June 1999), *Hypertext Transfer Protocol – HTTP/1.1*, Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T., eds., available at *<http://www.ietf.org/rfc/rfc2616.txt>*

UCUM, *Unified Code for Units of Measure*, Schadow, G. and McDonald, C.J. (eds.), version 1.5 *<http://aurora.regenstrief.org/UCUM/ucum.html>*

XML 1.0, *Extensible Markup Language (XML) 1.0*, World Wide Web Consortium Recommendation, Bray, T., Paoli, J., Sperberg-McQueen, C.M., and Maler, E., eds., available at *<http://www.w3.org/TR/>*

XML Schema, *XML Schema Part 1: Structures*, World Wide Web Consortium Recommendation, Thompson, H.S., Beech, D., Maloney, M., and Mendelsohn, N., eds., available at *<http://www.w3.org/TR/>*

4   Terms and definitions

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

4.1
client
software component that can invoke an **operation** from a **server**

4.2
coordinate reference system
coordinate system that is related to the real world by a datum

[ISO 19111]

4.3
coordinate system
set of mathematical rules for specifying how coordinates are to be assigned to points

[ISO 19111]

4.4
geographic information
information concerning phenomena implicitly or explicitly associated with a location relative to the Earth

[ISO 19101]

4.5
interface
named set of **operations** that characterize the behaviour of an entity

[ISO 19119]

4.6
layer
basic unit of **geographic information** that may be requested as a **map** from a **server**

4.7
map
portrayal of **geographic information** as a digital image file suitable for display on a computer screen

4.8
operation
specification of a transformation or query that an object may be called to execute

[ISO 19119]

4.9
portrayal
presentation of information to humans

[ISO 19117]

4.10
request
invocation of an **operation** by a **client**

4.11
response
result of an **operation** returned from a **server** to a **client**

4.12
server
a particular instance of a service

4.13
service
distinct part of the functionality that is provided by an entity through **interfaces**

[ISO 14252]

4.14
service metadata
metadata describing the **operations** and **geographic information** available at a **server**

5   Abbreviated terms

CDATA   XML Character Data

CRS   Coordinate Reference System

CS   Coordinate System

DCP   Distributed Computing Platform

DTD   Document Type Definition

EPSG   European Petroleum Survey Group

GIF   Graphics Interchange Format

GIS	Geographic Information System
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IERS	International Earth Rotation Service
IETF	Internet Engineering Task Force
ITRF	International Terrestrial Reference Frame
ITRS	IERS Terrestrial Reference System
JPEG	Joint Photographic Experts Group
MIME	Multipurpose Internet Mail Extensions
NAD	North American Datum
OGC	Open GIS Consortium
PNG	Portable Network Graphics
RFC	Request for Comments
SVG	Scalable Vector Graphics
UCUM	Unified Code for Units of Measure
URL	Uniform Resource Locator
WebCGM	Web Computer Graphics Metafile
WCS	Web Coverage Service
WFS	Web Feature Service
WGS	World Geodetic System
WMS	Web Map Service
XML	Extensible Markup Language

6 Basic service elements

6.1 Introduction

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

6.2 Version numbering and negotiation

6.2.1 Version number form and value

The Web Map Service (WMS) defines a protocol version number. The version number applies to the XML schema and the request encodings defined in this International Standard. The version number contains three non-negative integers, separated by decimal points, in the form "$x.y.z$". The numbers "y" and "z" shall not exceed 99.

Implementations of this International Standard shall use the value "1.3.0" as the protocol version number.

6.2.2 Version number changes

The protocol version number shall be changed with each revision of this International Standard. 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 draft versions. Servers and their clients need not support all defined versions, but shall obey the negotiation rules below.

6.2.3 Appearance in requests and in service metadata

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

6.2.4 Version number negotiation

A WMS client may negotiate with a server to determine a mutually agreeable protocol version. Negotiation is performed using the GetCapabilities operation (described in 7.2) according to the following rules.

All service metadata shall include a protocol version number and shall comply with the XML DTD or Schema defined for that version. In response to a GetCapabilities request (for which the VERSION parameter is optional) that does not specify a version number, the server shall respond with the highest version it supports. In response to a GetCapabilities request containing a version number that the server implements, the server shall send that version. If the server does not support the requested version, the server shall respond with output that conforms to a version it does support, as determined by the following rules:

⸺ If a version unknown to the server and higher than the lowest supported version is requested, the server shall send the highest version it supports that is less than the requested version.

⸺ If a version lower than any of those known to the server is requested, then the server shall send the lowest version it supports.

⸺ If the client does not support the version sent by the server, it may either cease communicating with the server or send a new request with a different version number that the client does support.

The process may be 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.

The VERSION parameter is mandatory in requests other than GetCapabilities.

6.3   General HTTP request rules

6.3.1   Introduction

This International Standard defines the implementation of the WMS on a distributed computing platform (DCP) comprising Internet hosts that support the Hypertext Transfer Protocol (HTTP) (see IETF RFC 2616). Thus, the Online Resource of each operation supported by a server 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 shall conform to the description in IETF RFC 2616 (section 3.2.2 "HTTP URL") but is otherwise implementation-dependent; only the query portion comprising the service request itself is defined by this International Standard.

HTTP supports two request methods: GET and POST. One or both of these methods may be offered by a server, and the use of the Online Resource URL differs in each case. Support for the GET method is mandatory; support for the POST method is optional.

6.3.2   Reserved characters in HTTP GET URLs

The URL specification (IETF RFC 2396) reserves particular characters as significant and requires that these be escaped when they might conflict with their defined usage. This International Standard explicitly reserves several of those characters for use in the query portion of WMS requests. When the characters "?", "&", "=", "," and "+" appear in one of the roles defined in Table 1, they shall appear literally in the URL. When those characters appear elsewhere (for example, in the value of a parameter), they shall be encoded as defined in IETF RFC 2396.

The server shall be prepared to decode any character escaped in this manner, and to decode the "+" character as a space.

Table 1 — Reserved characters in WMS query string

Character	Reserved usage
?	Separator indicating start of query string.
&	Separator between parameters in query string.
=	Separator between name and value of parameter.
,	Separator between individual values in list-oriented parameters (such as BBOX, LAYERS and STYLES in the GetMap request).
+	Shorthand representation for a space character.

6.3.3   HTTP GET

A WMS shall support the "GET" method of the HTTP protocol (IETF RFC 2616).

An Online Resource URL intended for HTTP GET requests is in fact only a URL prefix to which additional parameters are appended in order to construct a valid Operation request. A URL prefix is defined in accordance with IETF RFC 2396 as a string including, in order, the scheme ("http" or "https"), Internet Protocol hostname or

numeric address, optional port number, path, mandatory question mark "?", and optional string comprising one or more server-specific parameters ending in an ampersand "&". The prefix defines the network address to which request messages are to be sent for a particular operation on a particular server. Each operation may have a different prefix. Each prefix is entirely at the discretion of the service provider.

This International Standard defines how to construct a query part that is appended to the URL prefix in order to form a complete request message. Every WMS operation has several mandatory or optional request parameters. Each parameter has a defined name. Each parameter may have one or more legal values, which are either defined by this International Standard or are selected by the client based on service metadata. To formulate the query part of the URL, a client shall append the mandatory request parameters, and any desired optional parameters, as name/value pairs in the form "name=value&" (parameter name, equals sign, parameter value, ampersand). The "&" is a separator between name/value pairs, and is therefore optional after the last pair in the request string.

When the HTTP GET method is used, the client-constructed query part is appended to the URL prefix defined by the server, and the resulting complete URL is invoked as defined by HTTP (IETF RFC 2616).

Table 2 summarizes the components of an operation request URL when HTTP GET is used.

Table 2 — Structure of WMS request using HTTP GET

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.
name=value&	One or more standard request parameter name/value pairs as defined for each operation by this International Standard.

6.3.4 HTTP POST

A WMS may support the "POST" method of the HTTP protocol (IETF RFC 2616).

An Online Resource URL intended for HTTP POST requests is a complete URL (not merely a prefix as in the HTTP GET case) that is valid according to IETF RFC 2396 to which clients transmit request parameters in the body of the POST message. A WMS shall not require additional parameters to be appended to the URL in order to construct a valid target for the operation request. When POST is used, the request message is formulated as an XML document.

6.4 General HTTP response rules

Upon receiving a valid request, the server shall send a response corresponding exactly to the request as detailed in Clause 7 of this International Standard, or send a service exception if unable to respond correctly. Only in the case of Version Negotiation (see 6.2.4) may the server offer a differing result. Upon receiving an invalid request, the server shall issue a service exception as described in 6.11.

A server may send an HTTP Redirect message (using HTTP response codes as defined in IETF RFC 2616) to an absolute URL that is different from the valid request URL that was sent by the client. HTTP Redirect causes the client to issue a new HTTP request for the new URL. Several redirects could in theory occur. Practically speaking, the redirect sequence ends when the server responds with a WMS response. The final response shall be a WMS response that corresponds exactly to the original request (or a service exception).

Response objects shall be accompanied by the appropriate Multipurpose Internet Mail Extensions (MIME) type (IETF RFC 2045) for that object. A list of MIME types in common use on the internet is maintained by the Internet Assigned Numbers Authority (IANA) [2]. Allowable types for operation responses and service exceptions are discussed below. The basic structure of a MIME type is a string of the form "type/subtype". MIME allows additional parameters in a string of the form "type/subtype; param1=value1; param2=value2". A server may include parameterized MIME types in its list of supported output formats. In addition to any parameterized variants, the server should offer the basic unparameterized version of the format.

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.

6.5 Numeric and Boolean values

Integer numbers shall be represented in a manner consistent with the specification for integers in XML Schema Datatypes ([8], section 3.3.13). This International Standard shall explicitly indicate where an integer value is mandatory.

Real numbers shall be represented in a manner consistent with the specification for double-precision numbers in XML Schema Datatypes ([8], section 3.2.5). This representation allows for integer, decimal and exponential notations. A real value is allowed in all numeric fields defined by this International Standard unless the value is explicitly restricted to integer.

Positive, negative and zero values are allowed unless explicitly restricted.

Boolean values shall be represented in a manner consistent with the specification for Boolean in XML Schema Datatypes ([8], section 3.2.2). The values "0" and "false" are equivalent. The values "1" and "true" are equivalent. Absence of an optional value is equivalent to logical false. This International Standard shall explicitly indicate where a Boolean value is mandatory.

6.6   Output formats

The response to a Web Map Service request is always a computer file that is transferred over the Internet from the server to the client. The file may contain text, or the file may represent a map image. As stated in 6.4, the type of the returned file shall be indicated by a MIME type string.

Text output formats are usually formatted as Extensible Markup Language (XML; MIME type text/xml). Text formats are used to convey service metadata, descriptions of error conditions, or responses to queries for information about features shown on a map.

Allowed map formats are either "picture" formats or "graphic element" formats. Picture formats constitute a rectangular pixel array of fixed size. Picture formats include file types such as Graphics Interchange Format (GIF; MIME type "image/gif"), Portable Network Graphics (PNG; MIME type "image/png"), Joint Photographics Expert Group (JPEG; MIME type "image/jpeg"), all of which can be displayed by common Web browsers, and file types such as Tagged Image File Format (TIFF: MIME type "image/tiff") that may require additional software (beyond a basic Web browser) for display. Graphic element formats constitute a scale-independent description of the graphic elements to be displayed (including points, lines, curves, text and images), such that the size of the display may be modified while preserving the relative arrangement of the graphic elements. Graphic element formats include Scalable Vector Graphics (SVG; MIME type "image/svg+xml") or Web Computer Graphics Metafile (WebCGM; MIME type "image/cgm;Version=4;ProfileId=WebCGM") formats.

NOTE 1   SVG is expressed using XML, and could therefore be considered to be a text output format, but for the purposes of this International Standard SVG is considered to be a map format.

NOTE 2   WebCGM is a profile of ISO/IEC 8632.

A server may offer multiple map formats. The formats it offers are enumerated in <Format> elements in its service metadata. Use of a specific format is not required by this International Standard. However, for maps that portray vector features the server should offer at least one format that supports transparency in order that maps may be overlaid without obscuring other maps below (see the discussion about transparency in 7.3.3.9). Also, for ease of use, the server should offer at least one format that can be displayed by common Web browsers without additional software. Based on these considerations, the server should offer at least the PNG format.

6.7   Coordinate systems

6.7.1   Introduction

This International Standard uses two principal classes of Coordinate Systems: a Map CS applicable to the map portrayal generated by the WMS, and a Layer CRS for a Bounding Box applied to the source data. During a portrayal operation, a WMS converts or transforms geographic information from a Layer CRS into a Map CS. In addition, a Layer may have an associated vertical, temporal or other coordinate system.

6.7.2   Map CS

A Map CS is a coordinate reference system for a map produced by a WMS. A WMS map is a rectangular grid of pixels displayed on a computer screen (or a digital file that could be so displayed). The Map CS has a horizontal axis denoted i, and a vertical axis denoted j. i and j shall have only nonnegative integer values. The origin $(i,j) =$ (0,0) is the pixel in the upper left corner of the map; i increases to the right and j increases downward. The Map CS is defined using ISO 19111 terminology in B.2. The Map CS is identified by the label "CRS:1".

The usual orientation of the Map CS shall be such that the i axis is parallel to the East-to-West axis of the Layer CRS and increases Eastward, and the j axis is parallel to the North-to-South axis of the Layer CRS and increases Southward. This orientation will not be possible in some cases, as (for example) in an orthographic projection over the South Pole. The convention to be followed is that, wherever possible, East shall be to the right edge and North shall be toward the upper edge of the Map CS.

The WIDTH and HEIGHT parameters used in the GetMap request (see 7.3.3.8) and by inclusion in the GetFeatureInfo request (7.4.3.3) correspond to i and j as follows:

⎯ WIDTH denotes the size of the map image in pixels along the i axis (that is, WIDTH-1 is the maximum value of i).

⎯ HEIGHT denotes the size of the map image in pixels along the j axis (that is, HEIGHT-1 is the maximum value of j).

The I and J parameters used in the GetFeatureInfo request (see 7.4.3.7) denote integer values along the i and j axes, respectively, of the Map CS.

6.7.3   Layer CRS

6.7.3.1   Introduction

A Layer CRS is a horizontal coordinate reference system for the geographic information that serves as the source for a map. As discussed below, many Layer CRSs are possible. A Layer CRS appears in the following entities relevant to the WMS:

⎯ the <BoundingBox> element in the service metadata (7.2.4.6.8);

⎯ the CRS parameter in the GetMap request (7.3.3.5);

⎯ the CRS parameter in the map request part of the GetFeatureInfo request (7.4.3.3).

A WMS must support at least one CRS, and maps from multiple servers may be overlaid only if all the selected servers have at least one CRS in common. This International Standard does not mandate support for any particular Layer CRS(s). Instead, it only defines how CRSs are identified and discusses several optional Layer CRSs, in this clause and in Annex B. Map providers may support the CRSs that are most useful and appropriate to their geographic locale or information community. To maximize interoperability among servers, providers should also support geographic coordinates by geocentric coordinate systems such as "CRS:84" (see 6.7.3.2), "EPSG:4326" (see 6.7.3.3) or other ITRF-based systems.

Every Layer CRS has an identifier that is a character string. Two types of Layer CRS identifiers are permitted: "label" and "URL" identifiers:

⎯ **Label**: The identifier includes a namespace prefix, a colon, a numeric or string code, and in some instances a comma followed by additional parameters. This International Standard defines three namespaces: CRS, EPSG and AUTO2, as discussed below.

— **URL**: The identifier is a fully-qualified URL that references a publicly-accessible file containing a definition of the CRS that is compliant with ISO 19111.

The Layer CRS has two axes, denoted x and y. The x axis is the first axis in the CRS definition, the y axis is the second axis. Depending on the particular CRS, the x axis may or may not be oriented West-to-East, and the y axis may or may not be oriented South-to-North. The WMS portrayal operation shall account for axis order, origin and direction in the Layer CRS when projecting geographic information from a Layer CRS to the Map CS.

Coordinates shall be listed in the order defined by the CRS and shall be mapped appropriately to the Map CS i and j axes, swapping axis order as needed during the projection operation. Many projected coordinate reference systems have an axis and coordinate order other than easting, northing. For example, the Uniform Coordinate System used in Finland (EPSG:2393) orders northing before easting. EPSG geographic coordinate reference systems follow ISO 6709 and always list latitude before longitude.

Most coordinate reference systems are orientated with one axis positive east (easting) and the other axis positive north (northing). These map conveniently to the bounding box i and j axes, respectively. However, some CRSs have coordinates incrementing in other directions. For example, the Hartebeesthoek94/Lo25 system used in South Africa (EPSG:2051) has axes with coordinates incrementing to the west and south. Tests for valid bounding box areas shall recognise and account for the positive orientation of the CRS axes.

In a geographic CRS, latitude shall have values within the range [-90, 90] and longitude shall have values within the range [-180, 180] degrees or equivalent if the CRS definition is in other units. See 7.3.5 regarding the projection of Layer CRS that is a geographic CRS into the Map CS. When the CRS code specifies a geographic 2D coordinate reference system with axes in units other than degrees or in a degree representation other than decimal degrees the representation shall be converted to decimal degrees.

NOTE   The use of geographic CRSs with axis order of longitude before latitude differs from historical convention. Users in the international aviation and marine sectors may expect latitude to be before longitude, and a different coordinate display may have safety implications, especially in an emergency response situation. Although this International Standard does not specify human user interfaces, developers of user interfaces for WMSs are cautioned that all references to latitude and longitude, for example user input of bounding box or readout of cursor coordinates, should show latitude before longitude.

6.7.3.2   CRS namespace for CRS

The "CRS" namespace prefix refers to coordinate reference systems that are defined in Annex B of this International Standard. These definitions are in the form specified by ISO 19111. Geographic CRSs in the "CRS" namespace are defined in Annex B for the WGS 84, NAD27 and NAD83 datums.

A "CRS" CRS label comprises the "CRS" prefix, the colon, and a numeric or string code.

EXAMPLE   "CRS:84" refers to WGS 84 geographic longitude and latitude expressed in decimal degrees, with longitude ranging from −180 degrees to +180 degrees and latitude from −90 degrees to +90 degrees.

6.7.3.3   EPSG namespace for CRS

The "EPSG" namespace prefix refers to the European Petroleum Survey Group geodetic dataset (EPSG), which defines numeric identifiers (the EPSG "CRS code," corresponding to the field "COORD_REF_SYS_CODE" in the EPSG dataset) for many common coordinate reference systems. Defining geodetic, map projection and coordinate reference system data is related to each CRS identifier.

An "EPSG" CRS label comprises the "EPSG" prefix, the colon, and a numeric code.

EXAMPLE   EPSG:4326 refers to WGS 84 geographic latitude, then longitude. That is, in this CRS the x axis corresponds to latitude, and the y axis to longitude.

NOTE   EPSG geographic coordinate reference systems with axes of "degree – vendor-defined representation" are taken in this International Standard to be in decimal degrees.

6.7.3.4   AUTO2 namespace for CRS

The "AUTO2" namespace is used for "automatic" coordinate reference systems; that is, for a class of CRSs that include a user-selected centre of projection. Several "AUTO2" CRSs are defined in Annex B.

A complete "AUTO2" CRS label comprises the "AUTO2" namespace prefix, a numeric CRS identifier from the AUTO2 namespace, a number indicating what conversion factor to apply to convert between bounding box units and AUTO2 CRS units, and values for the central longitude and latitude in degrees:

```
AUTO2:auto_crs_id,factor,lon0,lat0
```

The conversion factor shall be nonzero and positive. If factor=1, then the units of the BBOX values are the same as those stated in the CRS definition. If factor is not 1, then factor is the ratio of BBOX units to CRS units. In a GetMap request, the complete AUTO2 CRS is specified, including longitude, latitude and units. In service metadata, the longitude, latitude and units variables are omitted because they are chosen by the client in a request for an AUTO2 CRS.

EXAMPLE 1   A server indicates that it supports Auto Orthographic CRS by including the element "`<CRS>AUTO2:42003</CRS>`" in its service metadata. A client may issue a GetMap request for a map in this CRS, with bounding box in meters, centered at 100 degrees West longitude and 45 degrees North latitude, using the parameter "`CRS=AUTO2:42003,1,-100,45`".

EXAMPLE 2   Same request as in example 1, but with conversion factor allowing bounding box in U.S. feet: "`CRS=AUTO2:42003,0.3048006096012192,-100,45`".

6.7.3.5   Geographic information with undefined CRS

A server may offer two-dimensional geographic information whose precise spatial reference is undefined. For example, a hand-drawn historical map may represent an area of the Earth but not employ a modern coordinate reference system, and an aerial photograph may not be precisely georeferenced. Such types of graphical information shall be treated as an image, and the Map CS label "CRS:1" shall be used when declaring the Layer CRS of such an object. Clients should not attempt to overlay a layer for which CRS=CRS:1 with another layer.

6.7.4   Bounding boxes

Bounding box values specify the portion of the Earth to be mapped through two pairs of coordinates in a specified Layer CRS. The first pair specifies the minimum coordinate values in the Layer CRS, the second specifies maximum coordinate values. Although for most CRSs with axes incrementing to the east and north this would be the lower left and upper right corners of the area of interest, the minimum and maximum values might be at other points in some instances. For example, when using geographic coordinates to describe an area over a pole, or when the layer CRS axes increment in directions other than east and north. The order in which ordinates in each pair are listed shall be as defined by the Layer CRS; x corresponds to the first axis in the Layer CRS and y to the second. This order may not coincide with the Map CS axis order i, j. The bounding box coordinate values shall be in the units defined for the Layer CRS.

NOTE   The use of two corner points to specify the map region is not the only possible method in theory. Other possibilities include specifying a central point and a radius, or a central point and a zoom level or scale. Some services that use mapping (e.g. location-based services) may find a formalism based on the central point more appropriate. This International Standard is not intended to preclude other service types from using other map region definitions internally or in client interactions. For most coordinate reference systems, it is a simple mathematical operation to transform between corner-point and central-point methods. Thus, for example, user input of a central point and radius could be converted to a two-point bounding box when requesting a map from a WMS that implements this International Standard.

Bounding Box values appear in the following entities relevant to the WMS:

⎯ the <BoundingBox> element in the service metadata (7.2.4.6.8);

⎯ the BBOX parameter in the GetMap request (7.3.3.6);

⎯ the BBOX parameter in the map request part of the GetFeatureInfo request (7.4.3.3).

A Bounding Box shall not have zero area.

EXAMPLE 1   A <BoundingBox> metadata element for a Layer representing the entire Earth in the CRS:84 Layer CRS would be written as
```
<BoundingBox CRS="CRS:84" minx="-180" miny="-90" maxx="180" maxy="90">.
```
A BBOX parameter requesting a map of the entire Earth would be written in this CRS as
```
BBOX=-180,-90,180,90.
```

EXAMPLE 2   A <BoundingBox> representing the entire Earth in the EPSG:4326 Layer CRS would be written as
```
<BoundingBox CRS="EPSG:4326" minx="-90" miny="-180" maxx="90" maxy="180">.
```
A BBOX parameter requesting a map of the entire Earth would be written in this CRS as
```
BBOX=-90,-180,90,180.
```

6.7.5   Vertical CRS

Some geographic information may be available at multiple elevations (for example, ozone concentrations at different heights in the atmosphere). A WMS may announce available elevations in its service metadata, and the GetMap operation includes an optional parameter for requesting a particular elevation. A single elevation or depth value is a number whose units, and the direction in which ordinates increment, are declared through a one-dimensional vertical CRS. Depending on the context, elevation values may appear as a single value, a list of values, or an interval, as specified in Annex C.

A server may declare at most one vertical CRS for each layer. For the purposes of this International Standard, the horizontal and vertical CRSs are treated as independent metadata elements and request parameters.

A request for a map at a specific elevation includes an elevation value but does not include the vertical CRS identifier (the horizontal CRS, which is included along with the horizontal bounding box in the request parameters). When providing elevation information, a server should declare a default value in service metadata, and a server shall respond with the default value if one has been declared and the client request does not include a value.

Two types of Vertical CRS identifiers are permitted: "label" and "URL" identifiers:

⎯ **Label**: The identifier includes a namespace prefix, a colon, and a numeric or string code. B.6 defines an optional vertical CRS labelled "CRS:88" based on the North American Vertical Datum 1988. If the namespace prefix is "EPSG", then the vertical CRS is one of those defined in the European Petroleum Survey Group database.

⎯ **URL**: The identifier is a fully-qualified Uniform Resource Locator that references a publicly-accessible file containing a definition of the CRS that is compliant with ISO 19111.

If the height is the vertical component of a 3-dimensional CRS, the Vertical CRS identifier shall be that of the 3-dimensional CRS.

6.7.6   Temporal CS

Some geographic information may be available at multiple times (for example, an hourly weather map). A WMS may announce available times in its service metadata, and the GetMap operation includes a parameter for requesting a particular time. The format of a time string is specified in Annex D. Depending on the context, time values may appear as a single value, a list of values, or an interval, as specified in Annex C. When providing

temporal information, a server should declare a default value in service metadata, and a server shall respond with the default value if one has been declared and the client request does not include a value.

6.7.7 Other coordinate systems

Some geographic 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". A WMS may announce available sample dimensions in its service metadata, and the GetMap operation includes a mechanism for requesting dimensional values. Each sample dimension has a Name and one or more valid values. The declaration and use of sample dimensions are specified in C.3.3.

6.8 Request parameter rules

6.8.1 Parameter ordering and case

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

Parameters in a request may be specified in any order.

When request parameters are duplicated with conflicting values, the response from the server may be undefined. This International Standard does not mandate which of the duplicated values sent by the client are to be used by the server.

A WMS shall be prepared to encounter additional request parameters that are not part of this International Standard. In terms of producing results per this International Standard, a WMS shall not require such parameters.

6.8.2 Parameter lists

Parameters consisting of lists (for example, BBOX, LAYERS and STYLES in WMS GetMap) shall use the comma (",") as the separator between items in the list. Additional white space shall not be used to delimit list items. If a list item value includes a space or comma, it shall be escaped using the URL encoding rules (see 6.3.2 and IETF RFC 2396).

In some lists, individual entries may be empty, and shall be represented by the empty string (""). Thus, two successive commas indicate an empty item, as does a leading comma or a trailing comma. An empty list ("") shall be interpreted either as a list containing no items or as a list containing a single empty item, depending on context.

6.9 Common request parameters

6.9.1 VERSION

The VERSION parameter specifies the protocol version number. The format of the version number, and version negotiation, are described in 6.2.

6.9.2 REQUEST

The REQUEST parameter indicates which service operation is being invoked. The value shall be the name of one of the operations offered by the server.

6.9.3 FORMAT

The FORMAT parameter specifies the output format of the response to an operation.

A WMS may offer only a subset of the formats known for that type of operation. The server shall advertise in its service metadata those formats it does support and shall accept requests for any format it advertises. A server may optionally offer a new format not previously offered by other instances, with the recognition that clients shall not be required to accept or process an unknown format. If a request contains a format not offered by a particular server, the server shall respond with the default format for that operation if one has been defined, or throw a service exception (with code "InvalidFormat") if no default has been defined.

A client may accept only a subset of the formats known for that type of operation. If a client and server do not support any mutually agreeable formats, the client may, at its discretion, cease communicating with that server, 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 a helper application).

Formats are expressed in both service metadata 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.

6.9.4   EXCEPTIONS

The EXCEPTIONS request parameter states the format in which to report errors (see 6.11).

6.9.5   Extended capabilities and operations

The Web Map Service allows for optional extended capabilities and operations. Extensions may be defined within an information community when needed for additional functionality or specialization. A generic client shall not be required or expected to make use of such extensions. Extended capabilities or operations shall be defined when necessary by providing instances of the abstract <_ExtendedCapabilities> or <_ExtendedOperations> elements in the service metadata schema in Annex E. Extended capabilities provide additional metadata about the service, and may or may not enable optional new parameters to be included in operation requests. Extended operations allow additional operations to be defined.

A server shall produce a valid response to the operations defined in this International Standard, even if parameters used by extended capabilities are missing or malformed (*i*.e. the server shall supply a default value for any extended capabilities it defines), or if parameters are supplied that are not known to the server.

Service providers shall choose extension names with care to avoid conflicting with standard metadata fields, parameters and operations.

6.10 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 valid request.

6.11 Service exceptions

Upon receiving a request that is invalid according to this International Standard, the server shall issue a service exception report. The service exception report is meant to describe to the client application or its human user the reason(s) that the request is invalid. The EXCEPTIONS parameter in a request indicates the format in which the client wishes to be notified of service exceptions. The allowed service exception formats are defined for each operation below.

Errors may arise in software modules other than those which implement the WMS, and may result in exception messages other than those defined by this International Standard. For example, when an error condition occurs in the local computing environment of the WMS server instance (e.g. out of memory or disk space), the server may

be unable to process the WMS request and may issue an error message. Or, upon receiving a request that is invalid according to the rules of the Distributed Computing Platform in use, the server may issue a service exception of a type valid in that DCP (e.g. if the URL prefix is incorrect, an HTTP 404 status code (IETF RFC 2616) may be sent).

7   Web Map Service operations

7.1   Introduction

The three operations defined for a WMS are GetCapabilities, GetMap, and GetFeatureInfo. GetFeatureInfo is optional. This clause specifies the implementation and use of these WMS operations in the Hypertext Transfer Protocol (HTTP) Distributed Computing Platform (DCP).

7.2   GetCapabilities (mandatory)

7.2.1   General

The purpose of the mandatory GetCapabilities operation is to obtain service metadata, which is a machine-readable (and human-readable) description of the server's information content and acceptable request parameter values.

7.2.2   GetCapabilities request overview

The general form of a WMS request is defined in 6.3.3 and 6.3.4. When making the GetCapabilities request of a WMS server, which may offer other service types as well, it is necessary to indicate that the client seeks information about the Web Map Service in particular. Thus, the SERVICE parameter of the request shall have the value "WMS" as shown in Table 3.

Table 3 — The parameters of a GetCapabilities request URL

Request parameter	Mandatory/optional	Description
VERSION=version	O	Request version
SERVICE=WMS	M	Service type
REQUEST=GetCapabilities	M	Request name
FORMAT=MIME_type	O	Output format of service metadata
UPDATESEQUENCE=string	O	Sequence number or string for cache control

7.2.3   Request parameters

7.2.3.1   FORMAT

The optional FORMAT parameter states the desired format of the service metadata. Supported values for a GetCapabilities request on a WMS server are listed in one or more <Request><GetCapabilities><Format> elements of its service metadata. Every server shall support the default text/xml format defined in Annex A. Support for other formats is optional. The entire MIME type string in <Format> is used as the value of the FORMAT parameter. In an HTTP environment, the MIME type shall be set on the returned object using the HTTP

21

Content-type entity header. If the request specifies a format not supported by the server, the server shall respond with the default text/xml format.

7.2.3.2   VERSION

The optional VERSION parameter, and its use in version negotiation, is defined in 6.2.

7.2.3.3   SERVICE

The mandatory SERVICE parameter indicates which of the available service types at a particular server is being invoked. When invoking GetCapabilities on a WMS that implements this version of the International Standard or a later one, the value "WMS" shall be used.

7.2.3.4   REQUEST

The nature of the mandatory REQUEST parameter is defined in 6.9.2. To invoke the GetCapabilities operation, the value "GetCapabilities" shall be used.

7.2.3.5   UPDATESEQUENCE

The optional UPDATESEQUENCE parameter is for maintaining cache consistency. Its value can be either an integer, or a string that represents a timestamp in ISO 8601:2004 format (see Annex D), or any other string. The server may include an UpdateSequence value in its service metadata. If present, this value 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 shall 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 service metadata
any	none	most recent service metadata
equal	equal	Exception: code=CurrentUpdateSequence
lower	higher	most recent service metadata
higher	lower	Exception: code=InvalidUpdateSequence

7.2.4   GetCapabilities response

7.2.4.1   Introduction

When invoked on a WMS, the response to a GetCapabilities request shall be an XML document containing service metadata formatted according to the XML Schema in E.1. The schema specifies the mandatory and optional content of the service metadata and how the content is formatted. The XML document shall contain a root element named WMS_Capabilities in the "*http://www.opengis.net/wms*" namespace. This element shall contain an XML Schema instance schemaLocation attribute that binds the "*http://www.opengis.net/wms*" namespace to the schema in E.1. The schema bound to the root attribute may be a copy of the schema in E.1 instead of the master copy at the URL stated in the annex provided none of the normative content of the schema is changed. The schema copy shall be located at a fully-qualified and accessible URL to permit XML validating software to retrieve it. The response shall be valid according to XML Schema validation rules.

A server may comply with other published or experimental protocol versions, in which case it shall support version negotiation as described in 6.2.

7.2.4.2 Names and titles

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

7.2.4.3 General service metadata

The first part of the service metadata is a <Service> element providing general metadata for the server as a whole. It shall include a Name, Title, and Online Resource URL. Optional service metadata includes Abstract, Keyword List, Contact Information, Fees, Access Constraints, and limits on the number of layers in a request or the output size of maps.

The Service Name shall be "WMS" in the case of a WMS.

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 OnlineResource element within the Service element may be used to refer to the web site of the service provider. There are other OnlineResource elements used for the URL prefix of each supported operation (see below).

A list of keywords or keyword phrases describing the server as a whole should be included to help catalogue searching. Each keyword may be accompanied by a "vocabulary" attribute to indicate the defining authority for that keyword. One "vocabulary" attribute value is defined by this International Standard: the value "ISO 19115:2003" refers to the metadata topic category codes defined in ISO 19115:2003, B.5.27; information communities may define other "vocabulary" attribute values. No particular vocabulary is mandated by this International Standard.

Contact Information should be included.

The optional <LayerLimit> element in the service metadata is a positive integer indicating the maximum number of layers a client is permitted to include in a single GetMap request. If this element is absent, the server imposes no limit.

The optional <MaxWidth> and <MaxHeight> elements in the service metadata are positive integers indicating the maximum width and height values that a client is permitted to include in a single GetMap request. If either element is absent the server imposes no limit on the corresponding parameter.

The optional elements <Fees> and <AccessConstraints> may be omitted if they do not apply to the server. If either of those elements is present, the reserved word "none" (case-insensitive) shall 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 text content of these elements, but client applications may display the content for user information and action.

7.2.4.4 Capability metadata

The <Capability> element of the service metadata names the actual operations that are supported by the server, the output formats offered for those operations, and the URL prefix for each operation. The XML schema includes

placeholders for Distributed Computing Platforms other than HTTP, but currently only the HTTP platform is defined.

7.2.4.5   Layers and styles

The most critical parts of the WMS service metadata are the Layers and Styles it defines. The geographic information content offered for portrayal by a WMS server is organized into "layers": metadata about the content is subdivided into descriptions of each layer, and a request for a map names one or more layers. The terms "layer" and "map" are defined in Clause 4; the relationship between them in the context of this International Standard is that a map is the result of a request to portray information represented by a layer.

The Web Map Service provider decides what information to include, exclude or aggregate in each layer. However, each layer should be documented in accordance with ISO 19115 or FGDC-STD-001-1998 [1], and a link to that metadata should be provided in the <MetadataURL> element defined below.

Each available map is advertised by a <Layer> element in the service metadata. Conceptually, each Layer is a distinct entity. However, as a means of classifying and organizing layers, and as a means of reducing the size of the service metadata, a single parent Layer may enclose any number of additional layers, which may be hierarchically nested as desired. 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. Subclause 7.2.4.8 summarizes whether or how each property is inherited.

A server shall include at least one <Layer> element for each map layer offered. If desired, layers may be repeated in different categories (*i*.e. enclosed in more than one parent <Layer>) when relevant.

A list of keywords or keyword phrases describing each layer should be included to help catalogue searching. Each keyword may be accompanied by a "vocabulary" attribute to indicate the defining authority for the keyword (see 7.2.4.3).

No controlled vocabulary of Layer and Style Names or Titles is defined by this International Standard, so they may be chosen at the discretion of the service provider organization or information community.

7.2.4.6   Layer properties

7.2.4.6.1   Introduction

The <Layer> element can enclose child elements providing metadata about the Layer. The meanings of those metadata elements are defined in this subclause. The values of some of these elements are inherited by subsidiary layers as defined in 7.2.4.8.

Some of the metadata values, and additional metadata, may be available in ISO 19115 or FGDC-STD-001-1998 [1] format as referenced via the <MetadataURL> element defined below. Table 5 shows the relationship between Layer Properties in this International Standard and ISO 19115 metadata elements. When documenting layers with ISO 19115 metadata, service providers should ensure that fields listed as equivalent in Table 5 have identical values in the full 19115 and in the summary 19128 layer properties.

Table 5 — Relationship between ISO 19128 and ISO 19115 metadata fields

ISO 19128 layer property	ISO 19115 Metadata element	Relationship
Title	CI_Citation.title	equivalent
Abstract	MD_DataIdentification.abstract	equivalent

Keyword element in KeywordList	MD_TopicCategoryCode	equivalent if "vocabulary" attribute of 19128 <Keyword> element is "ISO 19115:2003". Other vocabularies are permitted by 19128.
EX_GeographicBoundingBox	EX_GeographicBoundingBox	equivalent
CRS	MD_CRS	Value of CRS in 19128 is a text string that identifies a coordinate reference system defined by another authority. Value of MD_CRS in 19115 is a list of projection, ellipsoid and datum identifiers.
BoundingBox	EX_BoundingPolygon	19128:BoundingBox is strictly a box (lower left and upper right corners in a specified CRS). 19115:EX_BoundingPolygon may be a polygonal boundary.
Attribution	CI_ResponsibleParty	Title element of 19128:Attribution is equivalent to organisationName element of 19115: CI_ResponsibleParty. Other elements of 19128:Attribution have no equivalent in 19115.
Identifier and AuthorityURL	CI_Citation.identifier	The value of 19128:Identifier is equivalent to 19115:CI_Citation.identifier.code. 19128:AuthorityURL contains only the URL of the authority for a namespace, and is equivalent to 19115: CI_Citation.identifier.citation.citedResponsibleParty. contactInfo.onlineResource. 19115 allows a fuller description of the namespace authority.
DataURL	MD_metadata.dataSetURI	equivalent

7.2.4.6.2 Title

A <Title> is mandatory for all layers; it is a human-readable string for presentation in a menu. The Title is not inherited by child Layers.

7.2.4.6.3 Name

If, and only if, a 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. A Layer that contains a <Name> element is referred to as a "named layer" in this International Standard. If the layer has a Title but no Name, then that layer is only a category title for all the layers nested within. A server that advertises a Layer containing a Name element shall be able to accept that Name as the value of LAYERS argument in a GetMap request and return the corresponding map. A client shall not attempt to request a layer that has a Title but no Name.

A server shall throw a service exception (code="LayerNotDefined") if an invalid layer is requested.

A containing category itself may include a Name by which a map portraying 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 Name is not inherited by child Layers.

7.2.4.6.4 Abstract and KeywordList

The optional <Abstract> and <KeywordList> elements are optional, but a server should provide them. Abstract is a narrative description of the map layer. KeywordList contains zero or more <Keyword> elements to aid in catalogue searches. The Abstract and KeywordList elements are not inherited by child Layers.

7.2.4.6.5   Style

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

A <Style> may contain several other elements. <Abstract> provides a narrative description while <LegendURL> contains the location of an image of a map legend appropriate to the enclosing style. A <Format> element in LegendURL indicates the MIME type of the legend image, and the optional attributes width and height state the size of the image in pixels. Servers should provide the width and height attributes if known at the time of processing the GetCapabilities request. The legend image should clearly represent the symbols, lines and colours used in the map portrayal. The legend image should not contain text that duplicates the Title of the layer, because that information is known to the client and may be shown to the user by other means.

Style declarations are inherited by child Layers. A child shall not redefine a Style with the same Name as one inherited from a parent. A child may define a new Style with a new Name that is not available for the parent Layer.

7.2.4.6.6   EX_GeographicBoundingBox

Every named Layer shall have exactly one <EX_GeographicBoundingBox> element that is either stated explicitly or inherited from a parent Layer. EX_GeographicBoundingBox states, via the elements westBoundLongitude, eastBoundLongitude, southBoundLatitude, and northBoundLatitude, the minimum bounding rectangle in decimal degrees of the area covered by the Layer. EX_GeographicBoundingBox shall be supplied regardless of what CRS the map server may support, but it may be approximate if the data are not natively in geographic coordinates. The purpose of EX_GeographicBoundingBox is to facilitate geographic searches without requiring coordinate transformations by the search engine.

7.2.4.6.7   CRS

Every Layer is available in one or more layer coordinate reference systems. 6.7.3 discusses the Layer CRS. In order to indicate which Layer CRSs are available, every named Layer shall have at least one <CRS> element that is either stated explicitly or inherited from a parent Layer. The root <Layer> element shall include a sequence of zero or more CRS elements listing all CRSs that are common to <u>all</u> subsidiary layers. A child layer may optionally add to the list inherited from a parent layer. Any duplication shall be ignored by clients.

When a Layer is available in several coordinate reference systems, the list of available CRS values shall be represented as a sequence of <CRS> elements, each of which contains only a single CRS name.

EXAMPLE   <CRS>CRS:84</CRS> <CRS>EPSG:26718</CRS>.

7.2.4.6.8   BoundingBox

WMS service metadata shall declare one or more bounding boxes (as defined in 6.7.4) for each Layer. A Bounding Box metadata element may either be stated explicitly or may be inherited from a parent Layer. In XML, the <BoundingBox> metadata element includes the following attributes:

⎯ CRS indicates the Layer CRS that applies to this bounding box.

⎯ minx, miny, maxx, maxy indicate the limits of the bounding box using the axis units and order of the specified CRS.

⎯ resx and resy (optional) indicate the spatial resolution of the data comprising the layer in those same units.

The <BoundingBox> metadata element has the following relationship to the BBOX request parameter defined in 7.3.3.6. The <BoundingBox> metadata element specifies the coordinate range for the layer as a whole. The

BBOX request parameter, on the other hand, specifies which area is to be portrayed on the map. The BBOX area may or may not overlap, contain, or be contained within the BoundingBox area.

A Layer may have multiple BoundingBox elements, but each one shall state a different CRS. A Layer inherits any BoundingBox values defined by its parents. A BoundingBox inherited from the parent Layer for a particular CRS is replaced by any declaration for the same CRS in the child Layer. A BoundingBox in the child for a new CRS not already declared by the parent is added to the list of bounding boxes for the child Layer. A single Layer element shall not contain more than one BoundingBox for the same CRS.

NOTE   There is no provision for describing disjoint bounding boxes. For example, consider a dataset which covers two areas separated by some distance. The server cannot provide two separate bounding boxes in the same Layer using the same CRS to separately describe those areas. To handle this type of situation, the server may either define a single larger bounding box which encloses both areas, or may define two separate Layers that each have distinct Name and BoundingBox values.

A Layer shall not provide a BoundingBox for a CRS it does not support. Conversely, a Layer may support CRSs for which it does not provide a BoundingBox: a server that has the ability to transform data to different CRSs may choose not to provide an explicit BoundingBox for every possible CRS available for each Layer. The server should provide BoundingBox information for at least the native CRS of the Layer (that is, the CRS in which the Layer is stored in the server's database).

The <EX_GeographicBoundingBox> element (see 7.2.4.6.6) is conceptually similar to a BoundingBox in which the attribute CRS="CRS:84" is implicit. However, <EX_GeographicBoundingBox> shall not be used as a substitute for <BoundingBox CRS="CRS:84">. If the server wishes to provide bounding box information in the CRS:84 CRS, then a separate BoundingBox element explicitly naming CRS:84 shall be included in the service metadata.

If the Layer CRS is "CRS:1" (Map CS; see B.2), then the BoundingBox units are pixels, the origin is at the upper left, the first (x) axis increases to the right, and the second (y) axis increases toward the bottom.

7.2.4.6.9   Scale denominators

The <MinScaleDenominator> and <MaxScaleDenominator> elements define the range of scales for which it is appropriate to generate a map of a Layer. Because maps from a WMS may be viewed on arbitrary displays rather than on paper, the values used are actually the scale denominators relative to a common display pixel size. The intent of scale denominators is not that the translation between "actual" and "standard" scales always be completely accurate. Rather, the intent is to reduce the amount of clutter or crowding of features portrayed on the map. The scale denominator values are guidelines for clients, not firm limits. Upon receiving a request for a map that is not within the scale denominator range, the server may return a blank map, or may return a portrayal of the Layer that is crowded with features or otherwise poorly suited for display; the server shall not respond with a service exception.

For the purposes of this International Standard, the common pixel size is defined to be 0,28 mm × 0,28 mm. Because arbitrary clients can request maps from a server, the true pixel size of the final rendering device is unknown to the server.

EXAMPLE   A scale denominator value of "10000000" means a scale of 1:10 million. Scientific notation is also allowed, so a more compact value of "10e6" could also be used for the scale denominator.

The MinScaleDenominator and MaxScaleDenominator elements, as their names suggest, define the range of scale denominators of maps for which a Layer is appropriate. The minimum scale is inclusive and the maximum scale is exclusive. So, for example, the following scale range:

```
<MinScaleDenominator>1e3</MinScaleDenominator>

<MaxScaleDenominator>1e6</MaxScaleDenominator>
```

corresponds to the logical condition:

$$\texttt{scale\_denom} \geq 10^3 \texttt{ AND scale\_denom} < 10^6$$

Both of the elements are optional. The absence of a MinScaleDenominator element means there is no minimum-scale term to the condition or logically that the default value is zero. The absence of a MaxScaleDenominator element means that there is no maximum-scale term to the condition or logically that the default value is infinity. So, the following scale constraint:

`<MinScaleDenominator>1e3</MinScaleDenominator>`

corresponds to the logical condition:

$$\texttt{scale\_denom} \geq 10^3$$

There is a similar correspondence for a lone MaxScaleDenominator element. The absence of both scale elements in Layer metadata means that there is no scale constraint and that the Layer is applicable to maps of all scales.

Because it is possible to integrate the output of multiple servers into a single displayed result, it is important that different servers have consistent behaviour with respect to processing scales, so that all of the independent servers will select or deselect layers at the same scales. To ensure consistent behaviour, scales relative to coordinate spaces must be handled consistently between map servers. For geographic coordinate systems, which use angular units, the angular coverage of a map shall be converted to linear units for computation of scale by using the circumference of the Earth at the equator and by assuming perfectly square linear units. For linear coordinate systems, the size of the coordinate space should be used directly without compensating for distortions in it with respect to the shape of the real Earth.

EXAMPLE   A map to be displayed covers a 2-degree by 2-degree area in the WGS 84 geographic coordinates. The linear size of this area for conversion to scales is:
2º × (6378137m × 2 × π) ÷ 360º = 222638,9816m.
Thus, the map extent would be approximately 222639m × 222639m linear distance for the purpose of calculating the scale. If the image size for the map is 600×600 pixels, then the standard scale denominator for the map would be:

222638,9816m ÷ 600 pixels ÷ 0.00028m/pixel = 1325226,19.

Floating-point roundoff-error control should also be applied to these calculations, to ensure consistency between systems. Because the scale denominators used in Layer metadata will often be "round" numbers, such as 250000, if a calculation of the current scale results in a value of 249999,99999999, it should be considered to match 250000. A reasonable test for range matching of scale denominators would be defined as follows:

$$\texttt{scale} \geq \texttt{(min\_scale − epsilon) AND scale < (max\_scale + epsilon)}$$

where epsilon is defined as 1e-6.

Scale denominators are inherited by child Layers. A scale denominator declaration in the child replaces the declaration inherited from the parent.

7.2.4.6.10 Sample dimensions

Optional <Dimension> elements enclose metadata for multi-dimensional data. See the discussion of vertical, temporal and other coordinate systems in 6.7.5 through 6.7.7, and the explanation of how dimensional axes and allowed values are declared and requested in Annex C.

Dimension declarations are inherited from parent Layers. Any new Dimension declaration in the child with the same name attribute as one inherited from the parent replaces the value declared by the parent.

7.2.4.6.11 MetadataURL

A server should use one or more <MetadataURL> elements to offer detailed, standardized metadata about the data corresponding to a particular layer. The "type" attribute indicates the standard to which the metadata complies. Two "type" attribute values are defined by this International Standard: the value "ISO 19115:2003" refers to ISO 19115:2003; the value "FGDC:1998" refers to FGDC-STD-001-1998 [1]. An information community may define meanings for other "type" attribute values. The enclosed <Format> element indicates the file format MIME type of the metadata record.

MetadataURL elements are not inherited by child Layers.

7.2.4.6.12 Attribution

The optional <Attribution> element provides a way to identify the source of the geographic information used in a Layer or collection of Layers. Attribution encloses several optional elements: <OnlineResource> states the data provider's URL; <Title> is a human-readable string naming the data provider; <LogoURL> is the URL of a logo image. Client applications may choose to display one or more of these items. A <Format> element in LogoURL indicates the MIME type of the logo image, and the attributes `width` and `height` state the size of the image in pixels.

The Attribution element is inherited by child layers. Any redefinition by a child replaces the inherited value.

7.2.4.6.13 Identifier and AuthorityURL

A server may use zero or more <Identifier> elements to list ID numbers or labels defined by a particular Authority. The text content of the Identifier element is the ID value. The `authority` attribute of the Identifier element corresponds to the name attribute of a separate <AuthorityURL> element. AuthorityURL encloses an <OnlineResource> element which states the URL of a document defining the meaning of the Identifier values.

NOTE   The semantics of how an authority defines the meaning of an identifier have not yet been precisely defined. The Identifier and AuthorityURL elements are provided as a convenience for servers that wish to indicate the correspondence between the WMS Layers they offer and a classification of those Layers defined by the organization that operates the server.

EXAMPLE   The Global Change Master Directory (http://gcmd.gsfc.nasa.gov/) defines a DIF_ID label for every dataset that it has catalogued. A WMS that renders one of these datasets may associate a Layer with its DIF_ID in the following manner:

```
<AuthorityURL name="gcmd"><OnlineResource xlink:href="some_url" ... /></AuthorityURL>
<Identifier authority="gcmd">id_value</Identifier>
```

AuthorityURL is inherited by subsidiary layers. A child Layer shall not define an AuthorityURL with the same name attribute as one inherited from a parent. The Identifier element is not inherited. A Layer shall not declare an Identifier unless a corresponding AuthorityURL has been declared or inherited earlier in the service metadata.

7.2.4.6.14 FeatureListURL

A server may use a <FeatureListURL> element to point to a list of the features represented in a Layer. FeatureListURL is not inherited by child layers. The enclosed Format element indicates the file format MIME type of the feature list.

FeatureListURL is not inherited by child layers.

7.2.4.6.15 DataURL

A server may use DataURL to offer a link to the underlying data represented by a particular layer. The enclosed Format element indicates the file format MIME type of the data file.

DataURL is not inherited by child layers.

7.2.4.7 Layer attributes

7.2.4.7.1 Introduction

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. Each of these attributes can be inherited or replaced by subsidiary layers. The meaning of each attribute is summarized in Table 6 and detailed in the following subclauses.

Table 6 — Layer attributes

Attribute	Allowed values	Meaning (0 is default value)
queryable	0, false, 1, true	0, false: layer is not queryable. 1, true: layer is 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, false, 1, true	0, false: map data represents vector features that probably do not completely fill space. 1, true: map data are mostly or completely opaque.
noSubsets	0, false, 1, true	0, false: WMS can map a subset of the full bounding box. 1, true: WMS can only map the entire bounding box.
fixedWidth	0, positive integer	0: WMS can produce map of arbitrary width. nonzero: value is fixed map width that cannot be changed by the WMS.
fixedHeight	0, positive integer	0: WMS can produce map of arbitrary height. nonzero: value is fixed map height that cannot be changed by the WMS.

7.2.4.7.2 Queryable layers

The Boolean attribute **queryable** indicates whether the server supports the GetFeatureInfo operation on that Layer. A server may support GetFeatureInfo on some of its layers, but need not support it on all layers. A server shall issue a service exception (code="LayerNotQueryable") if GetFeatureInfo is requested on a Layer that is not queryable.

7.2.4.7.3 Cascaded layers

A Layer is said to have been "cascaded" if it was obtained from an originating server and then included in the service metadata 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 reprojection to other coordinate reference systems.

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

7.2.4.7.4   Opaque vs. transparent layers

If the optional Boolean attribute **opaque** is absent or false, 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 true value for **opaque** indicates that the Layer represents an area-filling coverage. For example, a map that represents topography and bathymetry as regions of differing colours 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 shall still comply with 7.3.3.9 regarding the GetMap TRANSPARENT parameter: that is, the server shall send an image with a transparent background if and only if the client requests TRANSPARENT=TRUE and a picture FORMAT that supports transparency.

7.2.4.7.5   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 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 browser 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 reference system, in which case the server shall declare CRS=CRS:1 as described in 6.7.2.

When set to a true value, **noSubsets** indicates that the server is not able to make a map of a geographic area other than the layer's bounding box.

When present and nonzero, **fixedWidth** and **fixedHeight** indicate that the server is not able to produce a map of the layer at a width and height different from the fixed sizes indicated.

7.2.4.8   Inheritance of layer properties

Table 7 summarizes how the properties of an enclosing parent <Layer> element are inherited by child <Layer> elements. Properties may be not inherited at all, or inherited as-is, or replaced if the child redefines them, or inherited and added to if the child also defines them

In Table 7, the **number** column states the number of times each element may appear in a Layer, either explicitly or though inheritance. Thus, it is more restrictive than the constraints enforced by the schema in E.1. The meanings of the values in this column are as follows:

⸺ 1:   appears exactly once in each Layer.

⸺ 0/1: appears either once or not at all.

⸺ 0+:   appears zero or more times.

⸺ 1+:   appears one or more times.

The **Inheritance** column indicates whether or how the element is inherited by child Layers. The meanings of the values in this column are as follows:

⸺ no: Not inherited. If the element is defined to be mandatory by this International Standard, then each Layer element shall include that element.

⸺ replace: Value can be inherited from parent and omitted by child, but if specified by child then the parent value is ignored.

⸺ add: Values can be inherited from parent and omitted by child. "Add" is only relevant for elements that may appear more than one time. Child inherits any value(s) supplied by parent and adds any value(s) of its own to the list. Any duplicated definition by the child is ignorable.

Table 7 — Inheritance of Layer properties

Element	Number	Inheritance
Layer	0+	no
Name	0/1 [a]	no
Title	1	no
Abstract	0/1	no
KeywordList	0/1	no
Style	0+	add
CRS	1+ [b]	add
EX_GeographicBoundingBox	1 [b]	replace
BoundingBox	1+ [b]	replace
Dimension	0+	replace
Attribution	0/1	replace
AuthorityURL	0+	add
Identifier	0+	no
MetadataURL	0+	no
DataURL	0/1	no
FeatureListURL	0/1	no
MinScaleDenominator, MaxScaleDenominator	0/1	replace
Attributes listed in Table 6	0/1	replace

[a]　See 7.2.4.6.3 regarding distinction between named and unnamed layers.

[b]　May be 0 only if a value is inherited from an enclosing Layer element; see 7.2.4.6.6 through 7.2.4.6.8.

7.2.5　Format specifiers

Format specifiers appear in several places in service metadata: as valid output formats for an operation, as supported Exception formats, and as the format of content at external URLs. Output formats are discussed in 6.6.

7.3　GetMap (mandatory)

7.3.1　General

The GetMap operation returns a map. Upon receiving a GetMap request, a WMS shall either satisfy the request or issue a service exception.

7.3.2 GetMap request overview

Table 8 describes the query part of the GetMap request.

Table 8 — The Parameters of a GetMap request

Request parameter	Mandatory/optional	Description
VERSION=1.3.0	M	Request version.
REQUEST=GetMap	M	Request name.
LAYERS=layer_list	M	Comma-separated list of one or more map layers.
STYLES=style_list	M	Comma-separated list of one rendering style per requested layer.
CRS=namespace:identifier	M	Coordinate reference system.
BBOX=minx,miny,maxx,maxy	M	Bounding box corners (lower left, upper right) in CRS units.
WIDTH=output_width	M	Width in pixels of map picture.
HEIGHT=output_height	M	Height in pixels of map picture.
FORMAT=output_format	M	Output format of map.
TRANSPARENT=TRUE|FALSE	O	Background transparency of map (default=FALSE).
BGCOLOR=color_value	O	Hexadecimal red-green-blue colour value for the background color (default=0xFFFFFF).
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default=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.

7.3.3 Request parameters

7.3.3.1 VERSION

The mandatory VERSION parameter is defined in 6.9.1. The value "1.3.0" shall be used for GetMap requests that comply with this International Standard.

7.3.3.2 REQUEST

The mandatory REQUEST parameter is defined in 6.9.2. To invoke the GetMap operation, the value "GetMap" shall be used.

7.3.3.3 LAYERS

The mandatory LAYERS parameter lists the map layer(s) to be returned by this GetMap request. 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 service metadata.

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

The optional <LayerLimit> element in the service metadata is a positive integer indicating the maximum number of layers a client is permitted to include in a single GetMap request. If this element is absent, the server imposes no limit.

7.3.3.4   STYLES

The mandatory STYLES parameter lists the style in which each layer is to be rendered. The value of the STYLES parameter is a comma-separated list of one or more valid style 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 shall be one that was defined in a <Style><Name> element that is either directly contained within, or inherited by, the associated <Layer> element in service metadata. (In other words, the client may not request a Layer in a Style that was only defined for a different Layer.) A server shall throw a service exception (code = StyleNotDefined) 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 shall include 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.

If the server advertises several styles for a layer, and the client sends a request for the default style, the choice of which style to use as default is at the discretion of the server. The ordering of styles in the service metadata does not indicate which is the default.

7.3.3.5   CRS

The CRS request parameter states what Layer CRS (see 6.7.3) applies to the BBOX request parameter. The value of the CRS parameter in a request to a particular server shall be one of the values defined in that server's service metadata in a <CRS> element defined or inherited by the requested layer. The same CRS applies to all layers in a single request.

A WMS server is not required to support all possible CRSs, but it shall advertise in its service metadata those CRSs which it does offer and shall accept requests for all advertised CRSs. If a request contains a CRS not offered by a particular server, the server shall throw a service exception (code = "InvalidCRS").

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

This International Standard does not define a facility for clients to explicitly request reprojection or coordinate transformation. That is, only a single CRS request parameter is defined, so it is not possible to specify a source CRS for selecting geographic information and a different target CRS for the portrayal output. Reprojection may occur implicitly, however, if the server offers multiple CRSs but internally stores its geographic information in a particular CRS.

If the WMS server has declared CRS=CRS:1 for a Layer, as discussed in 6.7.3.5, then the Layer does not have a well-defined coordinate reference system and should not be shown in conjunction with other layers. The client shall specify CRS=CRS:1 in the GetMap request and the server may issue a service exception otherwise. When this CRS is used in the request, the BBOX parameter units shall be pixels.

7.3.3.6 BBOX

The mandatory BBOX parameter allows a client to request a particular Bounding Box. Bounding Boxes are defined in 6.7.4. The value of the BBOX parameter in a GetMap request is a list of comma-separated real numbers (see 6.5) in the form `"minx,miny,maxx,maxy"`. These values specify the minimum X, minimum Y, maximum X, and maximum Y values of a region in the Layer CRS of the request. The units, ordering and direction of increment of the x and y axes are as defined by the Layer CRS (see 6.7.3) The four bounding box values indicate the outside limits of the region. The relation of the Bounding Box to the map pixel matrix is that the bounding box goes around the "outside" of the pixels of the map rather than through the centres of the map's border pixels. In this context, individual pixels represent an area on the ground.

If a request contains an invalid BBOX (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 shall throw a service exception.

If a request contains a BBOX whose area does not overlap at all with the <BoundingBox> element in the service metadata for the requested layer, the server shall return empty content (that is, a blank map or an graphic element file with no elements) for that map. Any features that are partly or entirely contained in the Bounding Box shall be returned in the appropriate format.

If the Bounding Box values are not defined for the given CRS (e.g. latitudes greater than 90 degrees in CRS:84), the server should return empty content for areas outside the valid range of the CRS.

If the WMS server has declared that a Layer is not subsettable, as described in 7.2.4.7.5, then the client shall specify exactly the declared Bounding Box values in the GetMap request and the server may issue a service exception otherwise.

7.3.3.7 FORMAT

The mandatory FORMAT parameter states the desired format of the map. Supported values for a GetMap request on a WMS server are listed in one or more <Request><GetMap><Format> elements of its service metadata. The entire MIME type string in <Format> is used as the value of the FORMAT parameter. There is no default format. In an HTTP environment, the MIME type shall be set on the returned object using the Content-type entity header. If the request specifies a format not supported by the server, the server shall issue a service exception (code = InvalidFormat).

7.3.3.8 WIDTH, HEIGHT

The mandatory WIDTH and HEIGHT parameters specify the size in integer pixels of the map to be produced. The Map CS (see 6.7.2 and B.2) applies to the map. WIDTH-1 specifies the maximum value of the i axis in the Map CS, and HEIGHT-1 specifies the maximum value of the j axis in the Map CS.

If the request is for a picture format, the returned picture, regardless of its MIME type, shall have exactly the specified width and height in pixels. In the case where the aspect ratio of the BBOX and the ratio width/height are different, the WMS shall stretch the returned map so that the resulting pixels could themselves be rendered in the aspect ratio of the BBOX. In other words, it shall 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.

Map distortions will be introduced if the aspect ratio WIDTH/HEIGHT is not commensurate with X, Y and the pixel aspect. Client developers should minimize the possibility that users will inadvertently request or unknowingly receive distorted maps.

If a request is for a graphic element format that does not have explicit width and height, the client shall include the WIDTH and HEIGHT values in the request and a server may use them as helpful information in constructing the output map.

The optional <MaxWidth> and <MaxHeight> elements in the service metadata are integers indicating the maximum width and height values that a client is permitted to include in a single GetMap request. If either element is absent, the server imposes no limit on the corresponding parameter.

If the WMS server has declared that a Layer has fixed width and height, as described in 7.2.4.7.5, then the client shall specify exactly those WIDTH and HEIGHT values in the GetMap request and the server may issue a service exception otherwise.

7.3.3.9   TRANSPARENT

The optional TRANSPARENT parameter specifies whether the map background is to be made transparent or not. TRANSPARENT can take on two values, `"TRUE"` or `"FALSE"`. The default value is `FALSE` if this parameter is absent from the request.

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 that could sensibly be overlaid above others.

NOTE   The image/gif format provides transparency and is properly displayed by common web clients. The image/png format provides a range of transparency options but support in viewing applications is less common. 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 shall 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. If the picture format does not support transparency, then the server shall respond with a non-transparent image (in other words, it is not an error for the client to always request transparent maps regardless of format). When TRANSPARENT is set to `FALSE`, non-data pixels shall be set to the value of BGCOLOR (see 7.3.3.10).

When the Layer has been declared "opaque" as in 7.2.4.7.4, then significant portions, or the entirety, of the map may not be able to made transparent. Clients may still request TRANSPARENT=true.

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.

7.3.3.10   BGCOLOR

The optional BGCOLOR parameter is a string that specifies the colour to be used as the background (non-data) pixels of the map. 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 colour values. The values can range between 00 and FF (0 and 255, base 10) for each. 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". The default value is `0xFFFFFF` (corresponding to the colour white) if this parameter is absent from the request.

When FORMAT is a picture format, a WMS shall set the background pixels to the colour specified by BGCOLOR. When FORMAT is a graphic element format (which does not have an explicit background), or a picture format, 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 colour.

When the Layer has been declared as "opaque" as in 7.2.4.7.4 (or is an area-filling coverage despite the absence of an "opaque" declaration), then significant portions, or the entirety, of the map may not show any background at all.

7.3.3.11 EXCEPTIONS

The optional EXCEPTIONS parameter is defined in 6.9.4. The default value is "XML" if this parameter is absent from the request.

A WMS shall offer one or more of the following exception reporting formats by listing them in separate <Format> elements inside the <Exceptions> element of its service metadata. The first of these formats must be offered by every WMS; the others are optional.

XML (mandatory)

Errors are reported using service exception XML, as specified in E.2. This is the default exception format if none is specified in the request.

The remaining exception formats are optional. A server may issue a service exception in the default XML format if a request specifies a different exception format not supported by the server.

INIMAGE (optional)

If the EXCEPTIONS parameter is set to INIMAGE, the WMS shall, upon detecting an error, return an object of the MIME type specified in the FORMAT parameter whose content includes text describing the nature of the error. In the case of a picture format, the error message shall be drawn on the returned picture. In the case of a graphic element format, the text of the error message shall be rendered in the manner that text is normally represented in that format.

BLANK (optional)

If the EXCEPTIONS parameter is set to 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 a picture format, that response shall be an image containing only pixels of one colour (the background colour). In the case of a picture format supporting transparency, if TRANSPARENT=TRUE is specified the pixels shall all be transparent. In the case of a graphic element output format, no visible graphic elements shall be included in the response output.

7.3.3.12 TIME

The use of Time values is specified in 6.7.6 and in Annexes C and D.

7.3.3.13 ELEVATION

The use of Elevation values is specified in 6.7.5 and in Annex C.

7.3.3.14 Other sample dimensions

The use of sample dimension values is specified in 6.7.7 and in Annex C.

7.3.4 GetMap response

The response to a valid GetMap request shall be a map of the spatially referenced information layer requested, in the desired style, and having the specified coordinate reference system, bounding box, size, format and transparency.

An invalid GetMap request shall 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 shall match the format of the return value.

7.3.5   Projection of geographic CRSs into Map CS

When the CRS parameter specifies a geographic coordinate reference system (such as CRS:84 or EPSG:4326), the spatial data is internally projected using the Pseudo Plate Carrée coordinate operation method and thereafter transformed to an image coordinate reference system with the i axis parallel and proportional to longitude and the j axis parallel and proportional to latitude to enable direct screen rendering. Because increments of latitude and longitude are both not consistent in linear terms, the image axes are then of variable scale. The geographic information is projected during the portrayal operation using a modified equidistant cylindrical projection as follows:

— The minimum and maximum values of longitude and latitude requested (min_lon, min_lat, max_lon, max_lat) are determined from the BBOX parameter, taking into account the x and y axis order specified by the CRS parameter.

— The angular distance (max_lon - min_lon) in degrees is scaled to the number of pixels specified by the WIDTH parameter.

— The angular distance (max_lat - min_lat) in degrees is scaled to the number of pixels specified by the HEIGHT parameter.

— The map is produced with longitude parallel to the i axis and latitude parallel to the j axis.

7.4   GetFeatureInfo (optional)

7.4.1   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. A client shall not issue a GetFeatureInfo request for other layers. A WMS shall respond with a properly formatted service exception (XML) response (code = OperationNotSupported) if it receives a GetFeatureInfo 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 (I,J) 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, CRS, WIDTH, HEIGHT) in that GetMap request, along with the I,J position the user chose, the WMS can (possibly) return additional information about that position.

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

7.4.2   GetFeatureInfo request overview

The parameters of a GetFeatureInfo request are listed in Table 9.

Table 9 — The Parameters of a GetFeatureInfo Request

Request parameter	Mandatory/optional	Description
VERSION=1.3.0	M	Request version.
REQUEST=GetFeatureInfo	M	Request name.
map request part	M	Partial copy of the Map request parameters that generated the map for which information is desired.
QUERY_LAYERS=layer_list	M	Comma-separated list of one or more layers to be queried.
INFO_FORMAT=output_format	M	Return format of feature information (MIME type).
FEATURE_COUNT=number	O	Number of features about which to return information (default=1).
I=pixel_column	M	i coordinate in pixels of feature in Map CS.
J=pixel_row	M	j coordinate in pixels of feature in Map CS.
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default= XML).

7.4.3   Request parameters

7.4.3.1   VERSION

The mandatory VERSION parameter is defined in 6.9.1. The value "1.3.0" shall be used for GetFeatureInfo requests that comply with this International Standard.

7.4.3.2   REQUEST

The mandatory REQUEST parameter is defined in 6.9.2. For GetFeatureInfo, the value "GetFeatureInfo" shall be used.

7.4.3.3   Map request part

The mandatory "map request part" entry in Table 9 represents a sequence of parameters from the GetMap request that generated the original map. Two of the GetMap parameters are omitted because GetFeatureInfo provides its own values: VERSION and REQUEST. The remainder of the GetMap request shall be embedded contiguously in the GetFeatureInfo request.

7.4.3.4   QUERY_LAYERS

The mandatory QUERY_LAYERS parameter states the map layer(s) from which feature information is desired to be retrieved. Its value is a comma-separated list of one or more map layers. This parameter shall contain at least one layer name, but may contain fewer layers than the original GetMap request.

If any layer in the QUERY_LAYERS parameter is not defined in the service metadata of the WMS, the server shall issue a service exception (code = LayerNotDefined).

7.4.3.5   INFO_FORMAT

The mandatory INFO_FORMAT parameter indicates what format to use when returning the feature information. Supported values for a GetFeatureInfo request on a WMS server are listed as MIME types in one or more <Request><FeatureInfo><Format> elements of its service metadata. The entire MIME type string in <Format> is used as the value of the INFO_FORMAT parameter. In an HTTP environment, the MIME type shall be set on the returned object using the Content-type entity header. If the request specifies a format not supported by the server, the server shall issue a service exception (code = InvalidFormat).

EXAMPLE   The parameter INFO_FORMAT=text/xml requests that the feature information be formatted in XML.

7.4.3.6   FEATURE_COUNT

The optional FEATURE_COUNT parameter states the maximum number of features per layer for which feature information shall be returned. Its value is a positive integer. The default value is 1 if this parameter is omitted or is other than a positive integer.

7.4.3.7   I, J

The mandatory I and J request parameters are integers that indicate a point of interest on the map that was produced by the embedded GetMap request (the "map request part" described in 7.4.3.3). The point (I,J) is a point in the (i,j) space defined by the Map CS (see 6.7.2). Therefore:

⎯ the value of I shall be between 0 and the maximum value of the i axis;

⎯ the value of J shall be between 0 and the maximum value of the j axis;

⎯ the point I=0, J=0 indicates the pixel at the upper left corner of the map;

⎯ I increases to the right and J increases downward.

The point (I,J) represents the centre of the indicated pixel.

If the value of I or of J is invalid, the server shall issue a service exception (code = InvalidPoint).

7.4.3.8   EXCEPTIONS

The optional EXCEPTIONS parameter is defined in 6.9.4. If this parameter is absent from the request, the default value is `"XML"`. No other values are defined by this International Standard for the WMS GetFeatureInfo request.

7.4.4   GetFeatureInfo response

The server shall return a response according to the requested INFO_FORMAT if the request is valid, or issue a service exception otherwise. The nature of the response is at the discretion of the service provider, but it shall pertain to the feature(s) nearest to (I,J).

Annex A
(normative)

Conformance tests

A.1 Basic WMS

A.1.1 Basic WMS client

A.1.1.1   Basic service elements

Information for the basis service elements is as follows:

a)   Test Purpose: Verify that a WMS client satisfies the requirements for request parameter rules.

b)   Test Method: Generate an adequate sample of requests from the client and verify that each is a valid request.

c)   Reference: 6.3, 6.5, 6.6, 6.7

d)   Test Type: Basic

A.1.1.2   GetCapabilities request

Information for the GetCapabilities request is as follows:

a)   Test Purpose: Verify that a basic WMS client satisfies all requirements for a GetCapabilities request.

b)   Test Method: Generate an adequate sample of GetCapabilities requests from the client and verify that each is a valid request.

c)   Reference: 7.2

d)   Test Type: Basic

A.1.1.3   GetMap request

Information for the GetMap request is as follows:

a)   Test Purpose: Verify that a basic WMS client satisfies all requirements for a GetMap request.

b)   Test Method: Generate an adequate sample of GetMap requests from the client and verify that each is a valid request.

c)   Reference: 7.3

d)   Test Type: Basic

A.1.2   Basic WMS server

A.1.2.1   Version negotiation

Information for the version negotiation is as follows:

a)   Test Purpose: Verify that a basic WMS server satisfies the requirements for version negotiation.

b)   Test Method: Submit requests containing version numbers both lower than and higher than the version supported by the server. Verify that the server responses are in accord with the rules for version negotiation.

c)   Reference: 6.2

d)   Test Type: Basic

A.1.2.2   Request parameter rules

Information for the request parameter rules is as follows:

a)   Test Purpose: Verify that a basic WMS server satisfies the requirements for request parameter rules.

b)   Test Method: Generate a sample of requests from a client. Include both invalid requests and valid request that vary within the limits allowed by the rules. Verify that the server provides an appropriate response in each case.

c)   Reference: 6.3, 6.4, 6.5, 6.6, 6.7

d)   Test Type: Basic

A.1.2.3   GetCapabilities response

Information for the GetCapabilities response is as follows:

a)   Test Purpose: Verify that a basic WMS server satisfies all requirements of the GetCapabilities operation.

b)   Test Method: Make several GetCapabilities requests using a variety of input parameters. Verify that an appropriate response is returned in each case.

c)   Reference: 7.2

d)   Test Type: Basic

A.1.2.4   GetMap response

Information for the GetMap response is as follows:

a)   Test Purpose: Verify that a basic WMS client satisfies all requirements of the GetMap operation.

b)   Test Method: Make several GetMap requests using a variety of input parameters. Verify that an appropriate response is returned in each case.

c)   Reference: 7.3

d)   Test Type: Basic

A.2 Queryable WMS

A.2.1 Client for queryable WMS – GetFeatureInfo request

Information for the GetFeatureInfo request is as follows:

a)   Test Purpose: Verify that a basic WMS client satisfies all requirements for a GetFeatureInfo request.

b)   Test Method: Generate an adequate sample of GetFeatureInfo requests from the client and verify that each is a valid request.

c)   Reference: 7.4

d)   Test Type: Basic

A.2.2 Queryable WMS server – GetFeatureInfo response

Information for the GetFeatureInfo response is as follows:

a)   Test Purpose: Verify that a WMS interface satisfies all requirements for the operation GetFeatureInfo.

b)   Test Method: Make several GetFeatureInfo requests using a variety of input parameters. Verify that an appropriate response is returned in each case.

c)   Reference: 7.4

d)   Test Type: Basic

Annex B
(normative)

CRS Definitions

B.1 Introduction

This annex defines several coordinate systems, including the Map CS (see 6.7.2), several Layer CRSs in the "CRS" namespace (see 6.7.3.2) and AUTO2 namespace (see 6.7.3.4), and a Vertical CRS (see 6.7.5). (Layer CRSs in the "EPSG" namespace are defined by the European Petroleum Survey Group, as described in 6.7.3.3).

All of the Layer CRSs and Vertical CRSs defined in this annex (or by another authority such as EPSG) are optional; the choice of which CRSs to support is at the discretion of the WMS provider. However, support for the Map CS defined in B.2 is mandatory, because that CS applies to the output map generated by the WMS.

The definitions in this annex use either ISO 19111 tabular format (Tables B.1 to B.5) or ISO 19125 [5] Well-Known Text representations.

B.2 Map CS (CRS:1)

Table B.1 — Definition of map coordinate system

Element name	Entry	Comment
Coordinate system identifier	CRS:1	See 6.7.2
Coordinate system type	Cartesian	
Datum identifier	computer display	
Datum type	engineering	
Coordinate system dimension	2	
Coordinate system axis name	i	
Coordinate system axis direction	east	Some maps, especially over the poles, may not have a well-defined linear axis direction.
Coordinate system axis unit identifier	pixel	
Coordinate system axis name	j	
Coordinate system axis direction	south	Some maps, especially over the poles, may not have a well-defined linear axis direction.
Coordinate system axis unit identifier	pixel	
Coordinate system remarks	Origin is in upper left. Axis values are non-negative integers.	

B.3 Layer CRS using WGS 84 longitude-latitude (CRS:84)

Table B.2 — Definition of Layer CRS using WGS 84 longitude-latitude

Element name	Entry	Comment
Coordinate reference system kind code	1	Single (not compound) CRS
Coordinate reference system identifier	CRS:84	
Coordinate reference system valid area	world	
Datum identifier	WGS 84	
Datum type	geodetic	
Prime meridian identifier	Greenwich	
Prime meridian Greenwich longitude	0 degree	
Ellipsoid identifier	WGS 84	
Ellipsoid semi-major axis	6378137.0 m	
Ellipsoid shape	true	
Ellipsoid inverse flattening	298.257223563	
Coordinate system identifier	Geodetic Coordinate System	
Coordinate system type	geodetic	
Coordinate system dimension	2	
Coordinate system axis name	longitude	The first (x) axis is longitude in this CRS.
Coordinate system axis direction	east	
Coordinate system axis unit identifier	degree	
Coordinate axis minimum value	-180 degrees	
Coordinate axis maximum value	180 degrees	
Coordinate system axis name	latitude	The second (y) axis is latitude in this CRS.
Coordinate system axis direction	north	
Coordinate system axis unit identifier	degree	
Coordinate axis minimum value	-90 degrees	
Coordinate axis maximum value	90 degrees	

NOTE   CRS:84 is permitted to be used at large scales (>10m/pixel) for describing or requesting map layers in any lon/lat CRS that is equivalent to within ~1m.

B.4 Layer CRS using NAD83 longitude-latitude (CRS:83)

Table B.3 — Definition of Layer CRS using NAD83 longitude-latitude

Element name	Entry	Comment
Coordinate reference system kind code	1	Single (not compound) CRS
Coordinate reference system identifier	CRS:83	
Coordinate reference system valid area	North America	
Datum identifier	NAD83	
Datum type	geodetic	
Prime meridian identifier	Greenwich	
Prime meridian Greenwich longitude	0 degree	
Ellipsoid identifier	GRS80	
Ellipsoid semi-major axis	6378137.0 m	
Ellipsoid shape	true	
Ellipsoid inverse flattening	298.257222101	
Coordinate system identifier	Geodetic Coordinate System	
Coordinate system type	geodetic	
Coordinate system dimension	2	
Coordinate system axis name	longitude	The first (x) axis is longitude in this CRS
Coordinate system axis direction	east	
Coordinate system axis unit identifier	degree	
Coordinate axis minimum value	−180 degrees	
Coordinate axis maximum value	180 degrees	
Coordinate system axis name	latitude	The second (y) axis is latitude in this CRS
Coordinate system axis direction	north	
Coordinate system axis unit identifier	degree	
Coordinate axis minimum value	−90 degrees	
Coordinate axis maximum value	90 degrees	

NOTE   This is an optional Layer CRS. See CRS valid area above. Within this area some applications may require an alternative system.

B.5 Layer CRS using NAD27 longitude-latitude (CRS:27)

Table B.4 — Definition of Layer CRS using NAD27 longitude-latitude

Element name	Entry	Comment
Coordinate reference system kind code	1	Single (not compound) CRS
Coordinate reference system identifier	CRS:27	
Coordinate reference system valid area	North America	
Datum identifier	NAD27	
Datum type	geodetic	
Prime meridian identifier	Greenwich	
Prime meridian Greenwich longitude	0 degree	
Ellipsoid identifier	Clarke 1866	
Ellipsoid semi-major axis	6378206.4 m	
Ellipsoid shape	true	
Ellipsoid inverse flattening	294.9786982	
Coordinate system identifier	Geodetic Coordinate System	
Coordinate system type	geodetic	
Coordinate system dimension	2	
Coordinate system axis name	longitude	The first (x) axis is latitude in this CRS
Coordinate system axis direction	east	
Coordinate system axis unit identifier	degree	
Coordinate axis minimum value	−180 degrees	
Coordinate axis maximum value	180 degrees	
Coordinate system axis name	latitude	The second (y) axis is longitude in this CRS
Coordinate system axis direction	north	
Coordinate system axis unit identifier	degree	
Coordinate axis minimum value	−90 degrees	
Coordinate axis maximum value	90 degrees	

NOTE   This is an optional Layer CRS. See CRS valid area above. Within this area, some applications may require an alternative system.

B.6 Vertical CRS using NAVD 88 (CRS:88)

Table B.5 — Definition of Vertical CRS using NAVD 88

Element name	Entry	Comment
Coordinate reference system kind code	1	
Coordinate reference system identifier	CRS:88	
Coordinate reference system valid area	North America	
Datum identifier	North American Vertical Datum 1988	
Datum type	vertical	
Coordinate system identifier	orthometric height	
Coordinate system type	gravity-related height	
Coordinate system dimension	1	
Coordinate system axis name	H	
Coordinate system axis direction	up	
Coordinate system axis unit identifier	meter	

NOTE   This is an optional vertical CRS. See CRS valid area above. Within this area some applications may require an alternative system.

B.7 Auto universal transverse mercator Layer CRS (AUTO2:42001)

In the notation below, "${var}" denotes a reference to the value of a variable "var". The variables "lat0" and "lon0" are the central point of the projection appearing in the CRS parameter of the map request (see 6.7.3.4). The coordinate operation method uses ellipsoidal formulas.

```
PROJCS["WGS 84 / Auto UTM",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
        SPHEROID["WGS_1984", 6378137, 298.257223563]],
        PRIMEM["Greenwich", 0],
        UNIT["Decimal_Degree", 0.0174532925199433]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["Latitude_of_Origin", 0],
    PARAMETER["Central_Meridian", ${central_meridian}],
    PARAMETER["False_Easting", 500000],
    PARAMETER["False_Northing", ${false_northing}],
    PARAMETER["Scale_Factor", 0.9996],
    UNIT["Meter", 1]]
```

where

```
${zone} = min( floor( (${lon0} + 180.0) / 6.0 ) + 1, 60 )
${central_meridian} = -183.0 + ${zone} * 6.0
```

```
${false_northing} = (${lat0} >= 0.0) ? 0.0 : 10000000.0
```

B.8 Auto transverse mercator Layer CRS (AUTO2:42002)

In the notation below, "`${var}`" denotes a reference to the value of a variable "`var`". The variables "lat0" and "lon0" are the central point of the projection appearing in the CRS parameter of the map request (see 6.7.3.4). The coordinate operation method uses ellipsoidal formulas.

```
PROJCS["WGS 84 / Auto Tr. Mercator",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
        SPHEROID["WGS_1984", 6378137, 298.257223563]],
        PRIMEM["Greenwich", 0],
        UNIT["Decimal_Degree", 0.0174532925199433]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["Latitude_of_Origin", 0],
    PARAMETER["Central_Meridian", ${central_meridian}],
    PARAMETER["False_Easting", 500000],
    PARAMETER["False_Northing", ${false_northing}],
    PARAMETER["Scale_Factor", 0.9996],
    UNIT["Meter", 1]]
```

where

```
${central_meridian} = ${lon0}
${false_northing} = (${lat0} >= 0.0) ? 0.0 : 10000000.0
```

B.9 Auto orthographic Layer CRS (AUTO2:42003)

In the notation below, "`${var}`" denotes a reference to the value of a variable "`var`". The variables "lon0" and "lat0" are the central point of the projection appearing in the CRS parameter of the map request (see 6.7.3.4). The coordinate operation method uses spherical formulas.

```
PROJCS["WGS 84 / Auto Orthographic",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
        SPHEROID["WGS_1984", 6378137, 298.257223563]],
        PRIMEM["Greenwich", 0],
        UNIT["Decimal_Degree", 0.0174532925199433]],
    PROJECTION["Orthographic"],
    PARAMETER["Latitude_of_Origin", ${latitude_of_origin}],
    PARAMETER["Central_Meridian", ${central_meridian}],
    UNIT["Meter", 1]]
```

where

```
${latitude_of_origin} = ${lat0}
${central_meridian} = ${lon0}
```

B.10 Auto equirectangular Layer CRS (AUTO2:42004)

In the notation below, "`${var}`" denotes a reference to the value of a variable "`var`". The variables "lon0" and "lat0" are the central point of the projection appearing in the CRS parameter of the map request (see 6.7.3.4). The coordinate operation method uses ellipsoidal formulas.

```
PROJCS["WGS 84 / Auto Equirectangular",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
        SPHEROID["WGS_1984", 6378137, 298.257223563]],
        PRIMEM["Greenwich", 0],
        UNIT["Decimal_Degree", 0.0174532925199433]],
    PROJECTION["Equirectangular"],
    PARAMETER["Latitude_of_Origin", 0],
    PARAMETER["Central_Meridian", ${central_meridian}],
    PARAMETER["Standard_Parallel_1", ${standard_parallel}],
    UNIT["Meter", 1]]
```

where

```
${standard_parallel} = ${lat0}
${central_meridian} = ${lon0}
```

B.11 Auto Mollweide Layer CRS (AUTO2:42005)

In the notation below, "${var}" denotes a reference to the value of a variable "var". The variables "lon0" and "lat0" are the central point of the projection appearing in the CRS parameter of the map request (see 6.7.3.4). The coordinate operation method uses ellipsoidal formulas.

```
PROJCS["WGS 84 / Auto Mollweide",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
        SPHEROID["WGS_1984", 6378137, 298.257223563]],
        PRIMEM["Greenwich", 0],
        UNIT["Decimal_Degree", 0.0174532925199433]],
    PROJECTION["Mollweide"],
    PARAMETER["Central_Meridian", ${central_meridian}],
    UNIT["Meter", 1]]
```

where

```
${central_meridian} = ${lon0}
```

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 the WMS. "Dimensions" in this annex refers to Time, Elevation, and sample dimensions; horizontal dimensions are expressed using Layer CRS (see 6.7.3). Optional elements in WMS metadata declare available values along one or more dimensional axes applicable to a Layer. GetMap requests for that layer may include parameters specifying dimensional value(s).

For example, consider a server that offers daily satellite composite images of the Earth. If the provision defined in this annex for handling multi-dimensional data were not available, then the server could give each layer a different name (e.g. by appending a string indicating the date to a base name), and a client would request the desired time by supplying the appropriate name in the GetMap request. The number of layers in the service metadata would grow during the lifetime of the satellite as new images were added to the server's database. If the imagery were also available in multiple wavelength bands, then number of layers would be the product of the number of available times and bands. By following the provisions in this annex, a more compact representation of available times and bands is possible. The server may declare a single name for the layer and enumerate available times and wavelength bands in its service metadata. A client then adds additional parameters to the GetMap request to request a specific time and band.

C.2 Declaring dimensions and their allowed values

The optional <Dimension> element is used in service metadata to declare that one or more dimensional parameters are relevant to a layer or group of layers. Table C.1 shows the fields in a Dimension element: a mandatory name, a mandatory measurement units specifier with an optional unitSymbol, a mandatory string indicating available value(s) of the corresponding dimension, an optional default value used when the client does not specify one, and several optional Boolean attributes indicating whether multiple values of the dimension may be requested at the same time, whether the nearest available value of a dimension will be returned in response to a request for a nearby value, and in the case of temporal extents, whether the most recent map may be requested without specifying its exact time.

Table C.1 — Contents of a dimension element

Field	Mandatory/ optional	Meaning
name	M	Attribute stating name of dimensional axis.
units	M	Attribute indicating units of dimensional axis.
unitSymbol	O	Attribute specifying symbol.
default	O	Attribute indicating default value that will be used if GetMap request does not specify a value. If attribute is absent, then shall respond with a service exception if request does not include a value for that dimension.
multipleValues	O	Boolean attribute indicating whether multiple values of the dimension may be requested. 0 (or "false") = single values only; 1 (or "true") = multiple values permitted. Default = 0.
nearestValue	O	Boolean attribute indicating whether nearest value of the dimension will be returned in response to a request for a nearby value. 0 (or "false") = request value(s) must correspond exactly to declared extent value(s); 1 (or "true") = request values may be approximate. Default = 0.
current	O	Boolean attribute valid only for temporal extents (i.e. if attribute name="time"). This attribute, if it either 1 or "true", indicates (a) that temporal data are normally kept current and (b) that the request parameter TIME may include the keyword "current" instead of an ending value (see C.4.1). Default = 0.
extent	M	Text content indicating available value(s) for dimension.

A Dimension element has the following format in XML:

```
<Dimension name="dimension_name" units=unit_name unitSymbol="symbol"
default="default_value" multipleValues="0|1" nearestValue="0|1"
current="0|1">extent</Dimension>
```

Dimension elements shall comply with the following rules:

— Dimension names shall be interpreted case-insensitively and should contain no whitespace.

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

— If the dimensional quantity has units, unit names should be taken from the Unified Code for Units of Measure (UCUM) if UCUM has an appropriate entry. When UCUM is used, the mandatory units attribute shall be an appropriate entry from the UCUM "name" column.

— If present, the unitSymbol shall be a 7-bit ASCII character string. When UCUM is used, the unitSymbol shall be the entry from the UCUM "c/s" column (case-sensitive abbreviation) corresponding to the UCUM name. UCUM prefix names and symbols may be present as well.

The extent string declares what value(s) along the Dimension axis are appropriate for the corresponding layer. The extent string has the syntax shown in Table C.2.

Table C.2 — Syntax for listing one or more extent 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 a <Dimension> element of service metadata.	

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 server. For instance, an instrument which continuously monitors randomly-occurring data may have no explicitly defined temporal resolution.

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

```
<Dimension name="time" units="ISO8601" />
<Dimension name="elevation" units="verticalCRSid" unitSymbol="symbol" />
```

The time units identifier "ISO 8601" refers to the ISO 8601:2000 time representation specified in Annex D. No unitSymbol is defined for the time dimension. If a server opts to specify time values in a different representation it shall use a dimension name other than "time" (e.g. "date" or "seconds").

The elevation units identifier "verticalCRSid" refers to a vertical CRS as in 6.7.5. The unitSymbol attribute shall be chosen to match the units of the specified vertical CRS.

EXAMPLE 1   The verticalCRSid "CRS:88" refers to the vertical CRS defined in B.6 (elevation in meters in the North American Vertical Datum 1988). The unitSymbol "m" would be used.

EXAMPLE 2   The following are some examples of server-defined dimensions:

```
<Dimension name="time" units="ISO8601" default="2003-10-17">1996-01-01/2003-10-17/P1D</Dimension>
<Dimension name="temperature" units="Kelvin" unitSymbol="K" default="300">230,300,400</Dimension>
<Dimension name="elevation" units="CRS:88" unitSymbol="m" default="0">0/10000/100</Dimension>
```

Dimension declarations are inherited by enclosed Layers, as specified in 7.2.4.8.

A layer may provide a dimension declaration that has the same name attribute (case-insensitive matching) as a declaration in another layer. However, the dimension shall not be re-declared using the same name with conflicting units or unitSymbol attributes. Extent values, and the attributes default, nearestValue and multipleValues, may be different for each layer.

C.3 Including dimensional values in a request

C.3.1 Introduction

Requests for a layer with one or more <Dimension> elements in its service metadata may require additional parameters to formulate a complete GetMap query. This following subclauses specifies how clients shall include dimensional parameters in the GetMap request.

C.3.2 Elevation and time values in requests

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

`ELEVATION=`*`value.`*

If a Layer has a Time dimension defined, then requests may include the parameter

`TIME=`*`value.`*

In either case, *value* uses the format described in Table C.2 to provide a single value, a comma-separated list, or an interval of the form start/end without a resolution. *value* shall not contain white space. 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 in the request is equivalent to a request for the Layer's default value (if defined) along that dimension (see C.4.2). All parameter names are case-insensitive as stated in 6.8.1, so, for example, "TIME", "Time" and "time" are all acceptable.

For the TIME parameter, the special keyword "current" may be used if the <Dimension name="time"> service metadata element includes a nonzero value for the "current" attribute, as described in C.2. 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.3.3 Sample dimension values in requests

Sample dimension parameters allow the client to request a particular layer along one or more dimensional axes other than time or elevation.

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> element in the service metadata). The resulting "dim_name" is case-insensitive. The use of the "dim_" prefix is to avoid clashes between server-defined dimension names and other request parameters defined by this International Standard. (The "time" and "elevation" dimensions, being predefined, do not use the prefix.)

To include a sample dimension value in a request URL, the dim_name parameter is followed by an equals sign "=" and a valid value, a comma-separated list, or an interval, as described in Table C.2.

EXAMPLE   A WMS Layer is described as having an extent along a dimension named "wavelength" as follows:
<Dimension name="wavelength" units="Angstrom" unitSymbol="Ao">3000,4000,5000,6000<Dimension>.
A GetMap request for a portrayal of the data at 4000 Angstroms would include the parameter "DIM_WAVELENGTH=4000".

A request may omit a dimension parameter, or use a null value, to request the default value (if supplied) along that dimension; see C.4.2.

C.3.4 Single- and multiple-valued requests

Whether a request for a spatially referenced 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 geographic information permit it. A server may include a nonzero multipleValue attribute in a Dimension element to indicate that multi-valued requests are supported. A server shall issue a service exception (code = InvalidDimensionValue) if it cannot comply with the dimensional parameters as specified.

EXAMPLE 1   A WMS server 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 server 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 server offers maps showing crime locations. Each crime has an associated date. A GetMap image request for all crimes occurring within a specific period would include a TIME range.

C.3.5 Applicability to multiple layers

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 layers individually as follows:

⎯ If the spatially referenced object has the corresponding dimension as a property, then the dimension value applies as if that object had been requested alone.

⎯ If the spatially referenced 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 GetMap request includes "LAYERS=temperature,coastlines&TIME=2001-05-09" 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 issuing an exception because TIME does not apply to its coastlines layer. If the date is invalid for the temperature layer, then the server may issue a service exception (code = InvalidDimensionValue) to that effect.

C.3.6 Example requests

The following are example parameter value combinations for multi-dimensional GetMap requests.

EXAMPLE 1   Request for single ozone map at specified time and height:

```
VERSION=1.3.0     WIDTH=600
REQUEST=GetMap    HEIGHT=300
LAYERS=ozone      TIME=2000-08-03
CRS=CRS:84        ELEVATION=1000
BBOX=-180,-90,180,90  FORMAT=image/gif
```

EXAMPLE 2   Request for movie loop at specified frame times:

```
VERSION=1.3.0     WIDTH=600
REQUEST=GetMap    HEIGHT=300
LAYERS=ozone      TIME=2000-07-01/2000-07-31/P1D
CRS=CRS:84        ELEVATION=1000
BBOX=-180,-90,180,90  FORMAT=video/mpeg
```

C.4 Server responses

C.4.1 Incorrect values

If the dimension value is invalid or missing from the client request, and no default or nearest-value behaviour was enabled as discussed in the following subclauses, then the server shall respond with a service exception (code = InvalidDimensionValue or code = MissingDimensionValue, as appropriate).

C.4.2 Default values

A WMS may declare a default for any dimension by including the "default" attribute in the <Dimension> element. A default value is optional but recommended. If a request does not include a value along this dimension, and a default has been declared, then the server shall send the default value. If there is no declared default, then the server shall 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 shall label the response object with the actual value used by default. In the HTTP environment, the labelling is performed using an HTTP response header. For **each** dimension to which a default value was applied, a header line of the following form shall be sent:

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

where "99" is a defined by HTTP (IETF RFC 2616) 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.4.3 Nearest values

A WMS 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 <Dimension> element, if present and nonzero, indicates that this behaviour is enabled.

If a request includes an imprecise dimensional value, and nearest value behaviour has been declared, then the server shall compute and send the nearest available value. The value shall be rounded, not merely truncated. If nearestValue selection is not supported, then the server shall 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 shall label the response object with the actual value found by rounding. In the HTTP environment, the labelling 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 shall be sent:

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

where "99" is a defined by HTTP (IETF RFC 2616) 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)

Web Map Service profile of ISO 8601

D.1 Overview

This Annex specifies the encoding of moments and periods in time to allow the Web Map Service to support temporal data descriptions and requests. This Annex is based on ISO 8601 and extends ISO 8601 by defining a syntax for expressing in a single string the start, end and periodicity of a data collection.

D.2 Time format details

D.2.1 Basic syntax

The basic time format uses ISO 8601:2000 "extended" format: up to 14 digits specifying century, year, month, day, hour, minute, 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

The precision may be reduced by omitting least-significant digits, as in the examples below. ISO 8601:2000 prefers a decimal comma before fractional seconds but allows a decimal period as in this International Standard. The century digits shall be included with the year; two-digit years shall not be used.

A time zone suffix is mandatory if the hours field appears in the time string. All times should be expressed in Coordinated Universal Time (UTC), indicated by the suffix Z (for "zulu"). When a local time applies, a numeric time zone suffix as defined by ISO 8601:2004, 5.3.4.1 shall be used. The absence of any suffix at all means local time in an undefined zone, which shall not be used in the global network of map servers enabled by this International Standard.

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 in UTC

D.2.2 Representation of years before 0001

As specified in ISO 8601, the year immediately preceding 0001 shall be denoted 0000. Years before 0000 shall be denoted by a leading minus sign (hyphen character). To denote years in the distant past, the year field shall be expanded beyond 4 digits and preceded by a minus sign.

EXAMPLE 1 -1350 (1350 "BC", approximate time of Tutankhamen)

EXAMPLE 2 -18000 (last Ice Age, 18 thousand years ago)

EXAMPLE 3   −150000000   (Jurassic period, 150 million years ago)

EXAMPLE 4   −5000000000   (Earth's crust solidifies, 5000 million years ago)

D.3 Period format

An ISO 8601 Period is used to indicate the time resolution of the available data. The ISO 8601 format for representing a period of time is used to represent the resolution: 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

D.4 Time lists and ranges

C.3 defines a syntax for representing single or multiple values of dimensions including time. Thus, 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 to indicate 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.

NOTE   This is an extension to ISO 8601:2000, which represents interval formats as start/end, or start/period, or period/end, or period alone. ISO 19128 allows start/end/period, where "period" in this context means "periodicity" rather than "amount of time between start and end of interval".

D.5 Examples

EXAMPLE 1   A single moment (scalar value): 2000-06-23T20:07:48.11Z

EXAMPLE 2   Quarterly data (comma-separated list): 1999-01-01,1999-04-01,1999-07-01,1999-10-01

EXAMPLE 3   Daily data taken at noon since 15 April 1995 (periodic interval):
             1995-04-22T12:00Z/2000-06-21T12:00Z/P1D

EXAMPLE 4   Current data refreshed every 30 min (periodic interval): 2000-06-18T14:30Z/2000-06-18T14:30Z/PT30M

Annex E
(normative)

XML Schemas

E.1 WMS Capabilities XML schema

This annex contains the XML schema for WMS capabilities against which service metadata from a server compliant with this International Standard may be validated. Comments and documentation elements in the schema are informative; in case of conflict with the main body of this International Standard, the main body takes precedence. The schemaLocation attributes are also informative.

The schema may also be found on-line at <http://schemas.opengis.net/wms/1.3.0/>.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/wms"
        xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:wms="http://www.opengis.net/wms"
        xmlns:xlink="http://www.w3.org/1999/xlink"
        elementFormDefault="qualified">

  <import namespace="http://www.w3.org/1999/xlink"
          schemaLocation="http://schemas.opengis.net/gml/2.1.2/xlinks.xsd"/>

  <!-- ******************************************************************* -->
  <!-- **  The Top-Level Element.                                    ** -->
  <!-- ******************************************************************* -->

  <element name="WMS_Capabilities">
    <annotation>
      <documentation>
        A WMS_Capabilities document is returned in response to a
        GetCapabilities request made on a WMS.
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref="wms:Service"/>
        <element ref="wms:Capability"/>
      </sequence>
      <attribute name="version" type="string" fixed="1.3.0"/>
      <attribute name="updateSequence" type="string"/>
    </complexType>
  </element>

  <!-- ******************************************************************* -->
  <!-- **  Elements Used In Multiple Places.                         ** -->
  <!-- ******************************************************************* -->

  <element name="Name" type="string">
    <annotation>
      <documentation>
        The Name is typically for machine-to-machine communication.
      </documentation>
    </annotation>
  </element>

  <element name="Title" type="string">
    <annotation>
      <documentation>
```

```
      The Title is for informative display to a human.
    </documentation>
  </annotation>
</element>

<element name="Abstract" type="string">
  <annotation>
    <documentation>
      The abstract is a longer narrative description of an object.
    </documentation>
  </annotation>
</element>

<element name="KeywordList">
  <annotation>
    <documentation>
      List of keywords or keyword phrases to help catalog searching.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="wms:Keyword" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>

<element name="Keyword">
  <annotation>
    <documentation>
      A single keyword or phrase.
    </documentation>
  </annotation>
  <complexType>
    <simpleContent>
      <extension base="string">
        <attribute name="vocabulary" type="string" />
      </extension>
    </simpleContent>
  </complexType>
</element>

<element name="OnlineResource">
  <annotation>
    <documentation>
      An OnlineResource is typically an HTTP URL.  The URL is placed in
      the xlink:href attribute, and the value "simple" is placed in the
      xlink:type attribute.
    </documentation>
  </annotation>
  <complexType>
    <attributeGroup ref="xlink:simpleLink"/>
  </complexType>
</element>

<element name="Format" type="string">
  <annotation>
    <documentation>
      A container for listing an available format's MIME type.
    </documentation>
  </annotation>
</element>

<!-- ********************************************************************** -->
<!-- **  General Service Metadata.                                     ** -->
<!-- ********************************************************************** -->

<element name="Service">
  <annotation>
    <documentation>
      General service metadata.
    </documentation>
```

```
          </annotation>
          <complexType>
            <sequence>
              <element name="Name">
                <simpleType>
                  <restriction base="string">
                    <enumeration value="WMS"/>
                  </restriction>
                </simpletype>
              </element>
              <element ref="wms:Title"/>
              <element ref="wms:Abstract" minOccurs="0"/>
              <element ref="wms:KeywordList" minOccurs="0"/>
              <element ref="wms:OnlineResource"/>
              <element ref="wms:ContactInformation" minOccurs="0"/>
              <element ref="wms:Fees" minOccurs="0"/>
              <element ref="wms:AccessConstraints" minOccurs="0"/>
              <element ref="wms:LayerLimit" minOccurs="0"/>
              <element ref="wms:MaxWidth" minOccurs="0"/>
              <element ref="wms:MaxHeight" minOccurs="0"/>
            </sequence>
          </complexType>
        </element>

        <element name="ContactInformation">
          <annotation>
            <documentation>
              Information about a contact person for the service.
            </documentation>
          </annotation>
          <complexType>
            <sequence>
              <element ref="wms:ContactPersonPrimary" minOccurs="0"/>
              <element ref="wms:ContactPosition" minOccurs="0"/>
              <element ref="wms:ContactAddress" minOccurs="0"/>
              <element ref="wms:ContactVoiceTelephone" minOccurs="0"/>
              <element ref="wms:ContactFacsimileTelephone" minOccurs="0"/>
              <element ref="wms:ContactElectronicMailAddress" minOccurs="0"/>
            </sequence>
          </complexType>
        </element>

        <element name="ContactPersonPrimary">
          <complexType>
            <sequence>
              <element ref="wms:ContactPerson"/>
              <element ref="wms:ContactOrganization"/>
            </sequence>
          </complexType>
        </element>

        <element name="ContactPerson" type="string"/>
        <element name="ContactOrganization" type="string"/>

        <element name="ContactPosition" type="string"/>

        <element name="ContactAddress">
          <complexType>
            <sequence>
              <element ref="wms:AddressType"/>
              <element ref="wms:Address"/>
              <element ref="wms:City"/>
              <element ref="wms:StateOrProvince"/>
              <element ref="wms:PostCode"/>
              <element ref="wms:Country"/>
            </sequence>
          </complexType>
        </element>

        <element name="AddressType" type="string"/>
        <element name="Address" type="string"/>
```

```
<element name="City" type="string"/>
<element name="StateOrProvince" type="string"/>
<element name="PostCode" type="string"/>
<element name="Country" type="string"/>

<element name="ContactVoiceTelephone" type="string"/>
<element name="ContactFacsimileTelephone" type="string"/>
<element name="ContactElectronicMailAddress" type="string"/>

<element name="Fees" type="string"/>
<element name="AccessConstraints" type="string"/>
<element name="LayerLimit" type="positiveInteger"/>
<element name="MaxWidth" type="positiveInteger"/>
<element name="MaxHeight" type="positiveInteger"/>
```

```
<!-- ********************************************************************* -->
<!-- **  The Capability Element.                                     ** -->
<!-- ********************************************************************* -->

<element name="Capability">
  <annotation>
    <documentation>
      A Capability lists available request types, how exceptions may be
      reported, and whether any extended capabilities are defined.
      It also includes an optional list of map layers available from this
      server.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="wms:Request"/>
      <element ref="wms:Exception"/>
      <element ref="wms:_ExtendedCapabilities" minOccurs="0"
               maxOccurs="unbounded"/>
      <element ref="wms:Layer" minOccurs="0"/>
    </sequence>
  </complexType>
</element>

<!-- ********************************************************************* -->
<!-- **  The Request Element.                                        ** -->
<!-- ********************************************************************* -->

<element name="Request">
  <annotation>
    <documentation>
      Available WMS Operations are listed in a Request element.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="wms:GetCapabilities"/>
      <element ref="wms:GetMap"/>
      <element ref="wms:GetFeatureInfo" minOccurs="0"/>
      <element ref="wms:_ExtendedOperation" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>

<element name="GetCapabilities" type="wms:OperationType"/>
<element name="GetMap" type="wms:OperationType"/>
<element name="GetFeatureInfo" type="wms:OperationType"/>
<element name="_ExtendedOperation" type="wms:OperationType" abstract="true"/>

<complexType name="OperationType">
  <annotation>
    <documentation>
      For each operation offered by the server, list the available output
      formats and the online resource.
    </documentation>
  </annotation>
  <sequence>
    <element ref="wms:Format" minOccurs="1" maxOccurs="unbounded"/>
    <element ref="wms:DCPType" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<element name="DCPType">
  <annotation>
    <documentation>
      Available Distributed Computing Platforms (DCPs) are listed here.
      At present, only HTTP is defined.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
```

```
            <element ref="wms:HTTP"/>
          </sequence>
        </complexType>
      </element>

      <element name="HTTP">
        <annotation>
          <documentation>
            Available HTTP request methods.  At least "Get" shall be supported.
          </documentation>
        </annotation>
        <complexType>
          <sequence>
            <element ref="wms:Get"/>
            <element ref="wms:Post" minOccurs="0"/>
          </choice>
        </complexType>
      </element>

      <element name="Get">
        <annotation>
          <documentation>
            The URL prefix for the HTTP "Get" request method.
          </documentation>
        </annotation>
        <complexType>
          <sequence>
            <element ref="wms:OnlineResource"/>
          </sequence>
        </complexType>
      </element>

      <element name="Post">
        <annotation>
          <documentation>
            The URL prefix for the HTTP "Post" request method.
          </documentation>
        </annotation>
        <complexType>
          <sequence>
            <element ref="wms:OnlineResource"/>
          </sequence>
        </complexType>
      </element>

      <!-- ********************************************************************* -->
      <!-- **  The Exception Element.                                        ** -->
      <!-- ********************************************************************* -->

      <element name="Exception">
        <annotation>
          <documentation>
            An Exception element indicates which error-reporting formats are
            supported.
          </documentation>
        </annotation>
        <complexType>
          <sequence>
            <element ref="wms:Format" minOccurs="1" maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>

      <!-- ********************************************************************* -->
      <!-- **  Extended Capabilities.                                        ** -->
      <!-- ********************************************************************* -->

      <element name="_ExtendedCapabilities" abstract="true">
        <annotation>
          <documentation>
            Individual service providers may use this element to report extended
```

```
      capabilities.
    </documentation>
  </annotation>
</element>

<!-- ******************************************************************* -->
<!-- **  The Layer Element.                                        ** -->
<!-- ******************************************************************* -->
```

```
<element name="Layer">
  <annotation>
    <documentation>
      Nested list of zero or more map Layers offered by this server.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="wms:Name" minOccurs="0"/>
      <element ref="wms:Title"/>
      <element ref="wms:Abstract" minOccurs="0"/>
      <element ref="wms:KeywordList" minOccurs="0"/>
      <element ref="wms:CRS" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="wms:EX_GeographicBoundingBox" minOccurs="0"/>
      <element ref="wms:BoundingBox" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="wms:Dimension" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="wms:Attribution" minOccurs="0"/>
      <element ref="wms:AuthorityURL" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="wms:Identifier" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="wms:MetadataURL" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="wms:DataURL" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="wms:FeatureListURL" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="wms:Style" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="wms:MinScaleDenominator" minOccurs="0"/>
      <element ref="wms:MaxScaleDenominator" minOccurs="0"/>
      <element ref="wms:Layer" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="queryable" type="boolean" default="0"/>
    <attribute name="cascaded" type="nonNegativeInteger"/>
    <attribute name="opaque" type="boolean" default="0"/>
    <attribute name="noSubsets" type="boolean" default="0"/>
    <attribute name="fixedWidth" type="nonNegativeInteger"/>
    <attribute name="fixedHeight" type="nonNegativeInteger"/>
  </complexType>
</element>

<element name="CRS" type="string">
  <annotation>
    <documentation>
      Identifier for a single Coordinate Reference System (CRS).
    </documentation>
  </annotation>
</element>

<element name="EX_GeographicBoundingBox">
  <annotation>
    <documentation>
      The EX_GeographicBoundingBox attributes indicate the limits of the enclosing
      rectangle in longitude and latitude decimal degrees.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element name="westBoundLongitude" type="wms:longitudeType" />
      <element name="eastBoundLongitude" type="wms:longitudeType" />
      <element name="southBoundLatitude" type="wms:latitudeType" />
      <element name="northBoundLatitude" type="wms:latitudeType" />
    </sequence>
  </complexType>
</element>

<element name="BoundingBox">
  <annotation>
    <documentation>
      The BoundingBox attributes indicate the limits of the bounding box
      in units of the specified coordinate reference system.
    </documentation>
  </annotation>
  <complexType>
    <attribute name="CRS" type="string" use="required"/>
    <attribute name="minx" type="double" use="required"/>
```

```
        <attribute name="miny" type="double" use="required"/>
        <attribute name="maxx" type="double" use="required"/>
        <attribute name="maxy" type="double" use="required"/>
        <attribute name="resx" type="double"/>
        <attribute name="resy" type="double"/>
    </complexType>
</element>

<element name="Dimension">
  <annotation>
    <documentation>
      The Dimension element declares the existence of a dimension and indicates what
      values along a dimension are valid.
    </documentation>
  </annotation>
  <complexType>
    <simpleContent>
      <extension base="string">
        <attribute name="name" type="string" use="required"/>
        <attribute name="units" type="string" use="required"/>
        <attribute name="unitSymbol" type="string"/>
        <attribute name="default" type="string"/>
        <attribute name="multipleValues" type="boolean" />
        <attribute name="nearestValue" type="boolean" />
        <attribute name="current" type="boolean" />
      </extension>
    </simpleContent>
  </complexType>
</element>

<element name="Attribution">
  <annotation>
    <documentation>
      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.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="wms:Title" minOccurs="0"/>
      <element ref="wms:OnlineResource" minOccurs="0"/>
      <element ref="wms:LogoURL" minOccurs="0"/>
    </sequence>
  </complexType>
</element>

<element name="LogoURL">
  <complexType>
    <sequence>
      <element ref="wms:Format"/>
      <element ref="wms:OnlineResource"/>
    </sequence>
    <attribute name="width" type="positiveInteger"/>
    <attribute name="height" type="positiveInteger"/>
  </complexType>
</element>

<element name="MetadataURL">
  <annotation>
    <documentation>
      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.  The format element indicates how the metadata is structured.
    </documentation>
  </annotation>
  <complexType>
```

```
      <sequence>
        <element ref="wms:Format"/>
        <element ref="wms:OnlineResource"/>
      </sequence>
      <attribute name="type" type="NMTOKEN" use="required" />
    </complexType>
  </element>

  <element name="AuthorityURL">
    <annotation>
      <documentation>
        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.
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref="wms:OnlineResource"/>
      </sequence>
      <attribute name="name" type="NMTOKEN" use="required"/>
    </complexType>
  </element>

  <element name="Identifier">
    <complexType>
      <simpleContent>
        <extension base="string">
          <attribute name="authority" type="string" use="required"/>
        </extension>
      </simpleContent>
    </complexType>
  </element>

  <element name="DataURL">
    <annotation>
      <documentation>
        A Map Server may use DataURL offer a link to the underlying data represented
        by a particular layer.
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref="wms:Format"/>
        <element ref="wms:OnlineResource"/>
      </sequence>
    </complexType>
  </element>

  <element name="FeatureListURL">
    <annotation>
      <documentation>
        A Map Server may use FeatureListURL to point to a list of the
        features represented in a Layer.
      </documentation>
    </annotation>
    <complexType>
      <sequence>
        <element ref="wms:Format"/>
        <element ref="wms:OnlineResource"/>
      </sequence>
    </complexType>
  </element>

  <element name="Style">
    <annotation>
      <documentation>
```

```
      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.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="wms:Name"/>
      <element ref="wms:Title"/>
      <element ref="wms:Abstract" minOccurs="0"/>
      <element ref="wms:LegendURL" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="wms:StyleSheetURL" minOccurs="0"/>
      <element ref="wms:StyleURL" minOccurs="0"/>
    </sequence>
  </complexType>
</element>

<element name="LegendURL">
  <annotation>
    <documentation>
      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 may be provided to assist client applications in laying out
      space to display the legend.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="wms:Format"/>
      <element ref="wms:OnlineResource"/>
    </sequence>
    <attribute name="width" type="positiveInteger"/>
    <attribute name="height" type="positiveInteger"/>
  </complexType>
</element>

<element name="StyleSheetURL">
  <annotation>
    <documentation>
      StyleSheeetURL provides symbology information for each Style of a Layer.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="wms:Format"/>
      <element ref="wms:OnlineResource"/>
    </sequence>
  </complexType>
</element>

<element name="StyleURL">
  <annotation>
    <documentation>
      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.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="wms:Format"/>
      <element ref="wms:OnlineResource"/>
    </sequence>
  </complexType>
</element>
```

```
<element name="MinScaleDenominator" type="double">
  <annotation>
    <documentation>
      Minimum scale denominator for which it is appropriate to
      display this layer.
    </documentation>
  </annotation>
</element>
<element name="MaxScaleDenominator" type="double">
  <annotation>
    <documentation>
      Maximum scale denominator for which it is appropriate to
      display this layer.
    </documentation>
  </annotation>
</element>

<!-- ********************************************************************* -->
<!-- **  Type Definitions.                                            ** -->
<!-- ********************************************************************* -->

<simpleType name="longitudeType">
  <restriction base="double">
    <minInclusive value="-180" />
    <maxInclusive value="180" />
  </restriction>
</simpleType>

<simpleType name="latitudeType">
  <restriction base="double">
    <minInclusive value="-90" />
    <maxInclusive value="90" />
  </restriction>
</simpleType>

</schema>
```

E.2 Service Exception schema

This annex contains the XML schema for service exception reports and also summarizes the defined exception codes and their meanings.

The schema may also be found on-line at <http://schemas.opengis.net/wms/1.3.0/>.

Service exception XML shall be valid according to this service exception schema. In an HTTP environment, the MIME type of the returned XML shall be "text/xml". Individual error messages appear as <ServiceException> elements within the <ServiceExceptionReport> element. The messages can be formatted either as chunks of plain text or, if included in a character data (CDATA) section, as XML-like text containing angle brackets ("<" and ">"), as shown in the example service exception Report in H.2.

The "locator" attribute defined below is an optional string that a server may use to indicate what part(s) of a service request caused the exception to be generated. The locator is intended to be interpreted by humans rather than by client software, and as such has no predefined syntax.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
    targetNamespace="http://www.opengis.net/ogc"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">

    <xsd:element name="ServiceExceptionReport">
        <xsd:complexType>
```

```
        <xsd:sequence>
            <xsd:element name="ServiceException"
                         type="ogc:ServiceExceptionType"
                         minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="version" type="xsd:string" fixed="1.3.0"/>
    </xsd:complexType>
</xsd:element>

<xsd:complexType name="ServiceExceptionType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="code" type="xsd:string"/>
            <xsd:attribute name="locator" type="xsd:string"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
</xsd:schema>
```

This International Standard defines several exception codes in Table E.1. Servers shall not use these codes for meanings other than those specified. Clients may use these codes to automate responses to service exceptions.

Table E.1 — Service exception codes

Exception code	Meaning
InvalidFormat	Request contains a Format not offered by the server.
InvalidCRS	Request contains a CRS not offered by the server for one or more of the Layers in the request.
LayerNotDefined	GetMap request is for a Layer not offered by the server, or GetFeatureInfo request is for a Layer not shown on the map.
StyleNotDefined	Request is for a Layer in a Style not offered by the server.
LayerNotQueryable	GetFeatureInfo request is applied to a Layer which is not declared queryable.
InvalidPoint	GetFeatureInfo request contains invalid I or J value.
CurrentUpdateSequence	Value of (optional) UpdateSequence parameter in GetCapabilities request is equal to current value of service metadata update sequence number.
InvalidUpdateSequence	Value of (optional) UpdateSequence parameter in GetCapabilities request is greater than current value of service metadata update sequence number.
MissingDimensionValue	Request does not include a sample dimension value, and the server did not declare a default value for that dimension.
InvalidDimensionValue	Request contains an invalid sample dimension value.
OperationNotSupported	Request is for an optional operation that is not supported by the server.

Annex F
(normative)

UML model

This annex provides Unified Modeling Language (UML) class diagrams for the WMS.

Figure F.1 shows WMS datatypes used in the UML model. Table F.1 maps these classes and their attributes to WMS service metadata elements and request parameters in the HTTP DCP. Figure F.2 is a class diagram that summarizes the WMS interface. The service is composed of a set of operations and a body of data acted upon by those operations. That data includes both geographic information and metadata about the service. An instance of the service is called a server.

The WMS operations are GetCapabilities (see 7.2), GetMap (see 7.3), and GetFeatureInfo (see 7.4); the latter is optional. Figure F.3 provides a class diagram for each request message. Request messages in the HTTP DCP are URL query strings as described generally in 6.8 and 6.9, and specifically for each operation in 7.2.3, 7.3.3, and 7.4.3. Table F.2 shows the mapping of UML request classes to the HTTP DCP request names. Tables F.3, F.4 and F.5 shows the mapping of the attributes of each UML request class to request parameters in the HTTP DCP.

Class diagrams for the WMS responses are shown in Figure F.4. The responses are discussed in 7.2.4, 7.3.4, and 7.4.4.
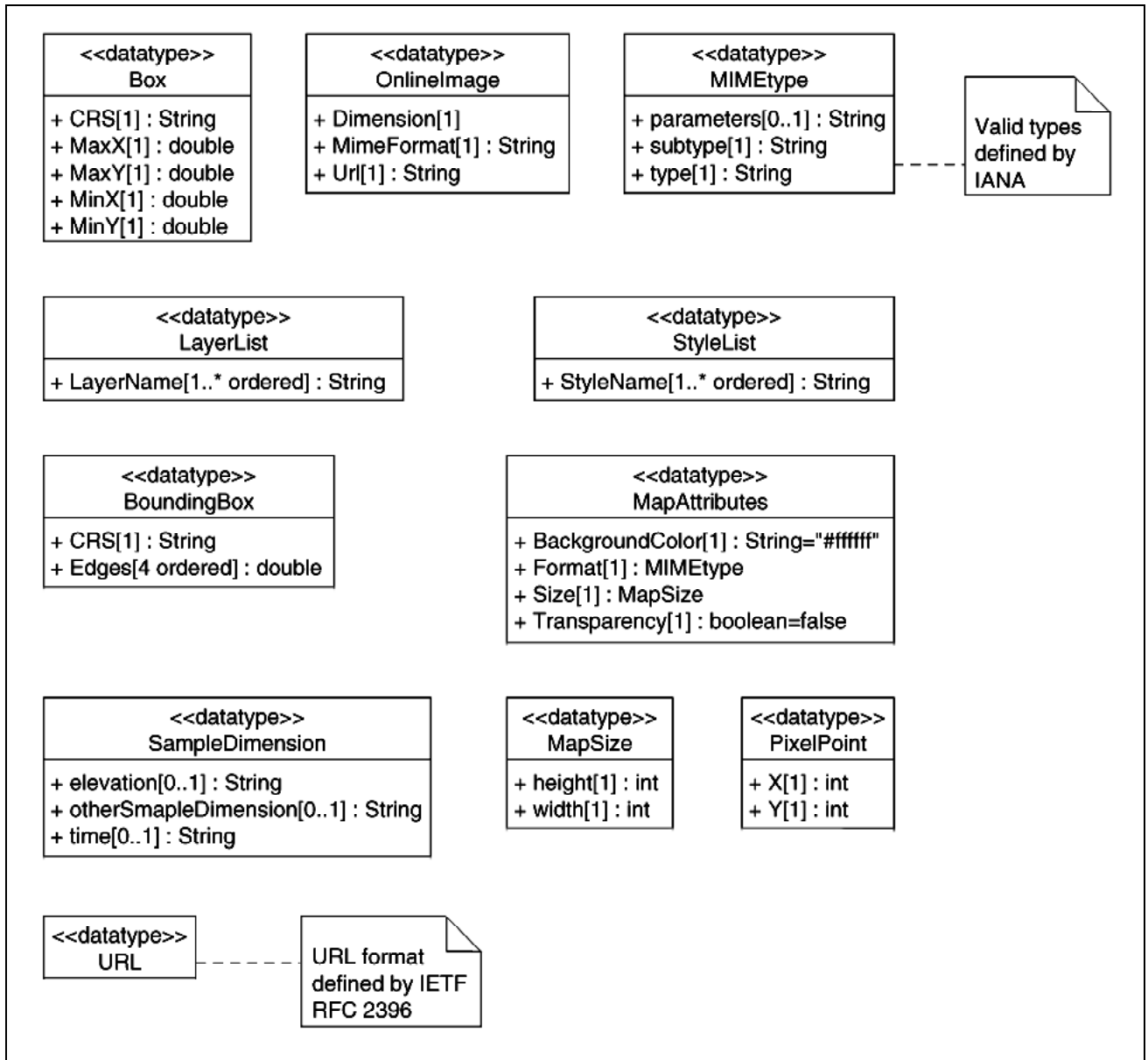
Figure F.1 — Web Map Service Datatypes

Table F.1 — Mapping of UML datatypes to HTTP DCP

UML Class	Service Metadata Element	Request Parameter (GetMap unless otherwise noted)	Other
Bounding Box	BoundingBox, CRS	BBOX, CRS	
OnlineImage	LegendURL, LogoURL		GetMap response
MIMEtype	Format	FORMAT	
LayerList		LAYERS	
StyleList		STYLES	
MapAttributes		BGCOLOR, FORMAT, WIDTH, HEIGHT, TRANSPARENT	
SampleDimension	Dimension	TIME, ELEVATION, DIM_dimension_name	
MapSize		WIDTH, HEIGHT	
PixelPoint		I, J (GetFeatureInfo)	
URL	OnlineResource, LegendURL, LogoURL		
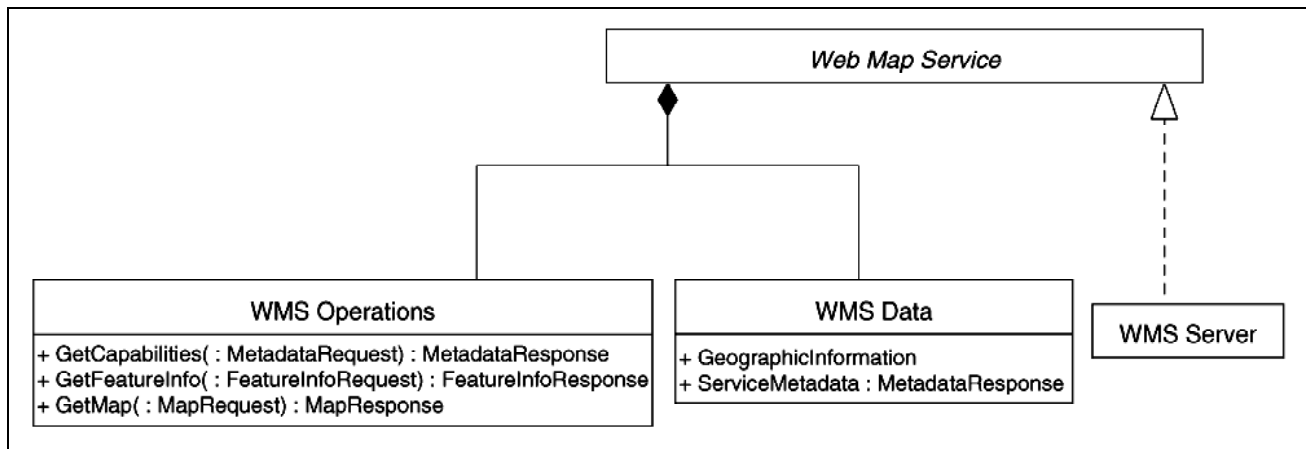
Figure F.2 — Web Map Service interface

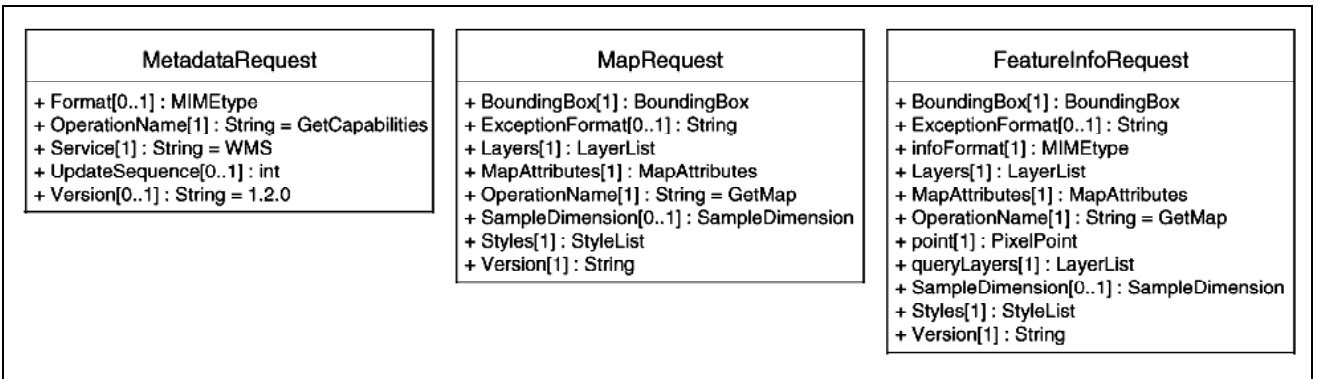**Figure F.3 — Web map service requests**

Table F.2 — Mapping of UML request classes to HTTP DCP

UML class	HTTP request name
MetadataRequest	GetCapabilities
MapRequest	GetMap
FeatureInfoRequest	GetFeatureInfo

Table F.3 — Mapping of UML MetadataRequest attributes HTTP GetCapabilities request parameters

UML attribute	HTTP request parameter
Format	FORMAT
OperationName	REQUEST
Service	SERVICE
UpdateSequence	UPDATESEQUENCE
Version	VERSION

Table F.4 — Mapping of UML MapRequest attributes HTTP GetMap request parameters

UML attribute	HTTP request parameters
BoundingBox	CRS, BBOX
ExceptionFormat	EXCEPTIONS
Layers	LAYERS
MapAttributes	BGCOLOR, FORMAT, WIDTH, HEIGHT, TRANSPARENT
OperationName	REQUEST
SampleDimension	TIME, ELEVATION, DIM_dimension_name
Styles	STYLES
Version	VERSION

Table F.5 — Mapping of UML FeatureInfoRequest attributes HTTP GetFeatureInfo request parameters

UML attribute	HTTP request parameters
BoundingBox	CRS, BBOX
ExceptionFormat	EXCEPTIONS
InfoFormat	INFO_FORMAT
Layers	LAYERS
MapAttributes	BGCOLOR, FORMAT, WIDTH, HEIGHT, TRANSPARENT
OperationName	REQUEST
point	I, J
queryLayers	QUERY_LAYERS
SampleDimension	TIME, ELEVATION, DIM_dimension_name
Styles	STYLES
Version	VERSION
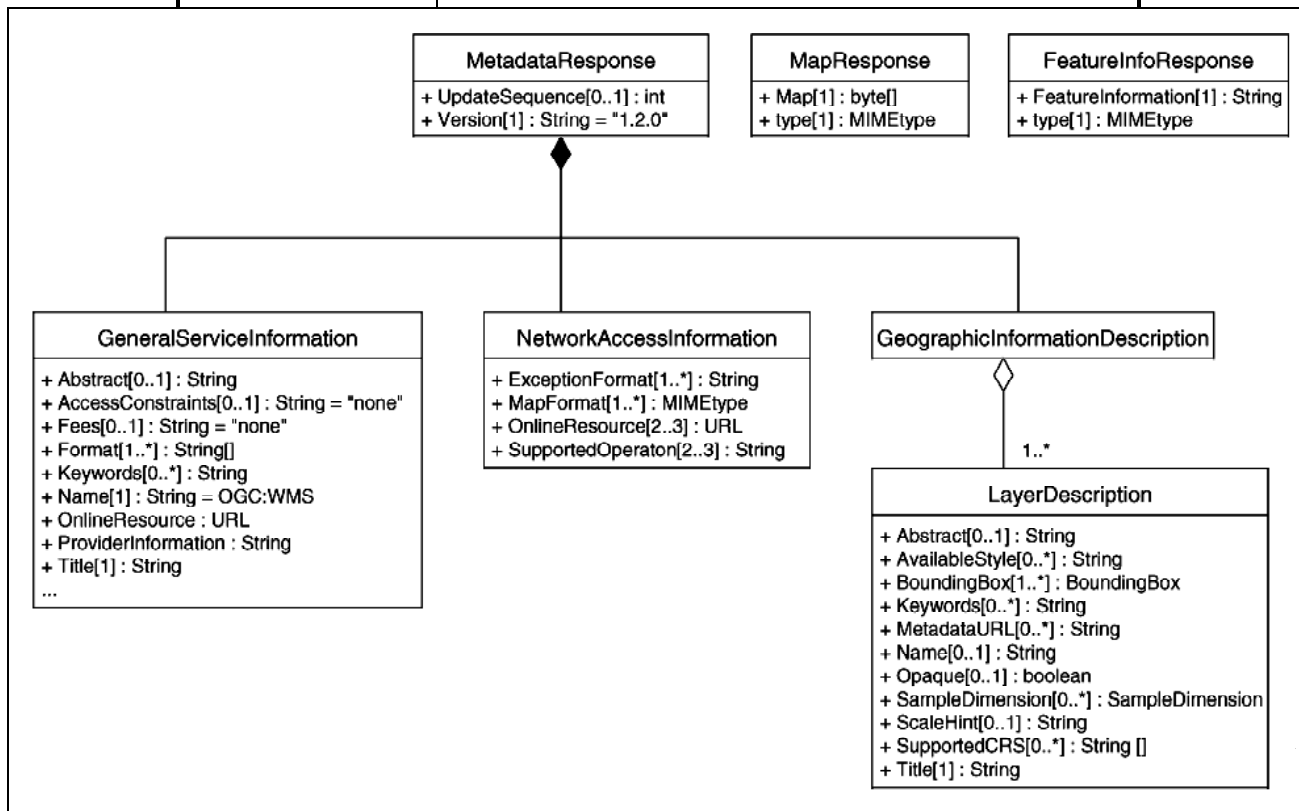
Figure F.4 — Web Map Service responses

Annex G
(informative)

Web Mapping Examples

This annex provides some illustrative URLs and their resulting maps.

G.1 Example 1: One Server, One Layer, Default Style

The following hypothetical URL requests the US National Oceanographic and Atmospheric Administration hurricane image shown in Figure G.1:

```
http://a-map-co.com/mapserver.cgi?VERSION=1.3.0&REQUEST=GetMap&
CRS=CRS:84&BBOX=-97.105,24.913,-78.794,36.358&
WIDTH=560&HEIGHT=350&LAYERS=AVHRR-09-27&STYLES=&
FORMAT=image/png&EXCEPTIONS=INIMAGE
```
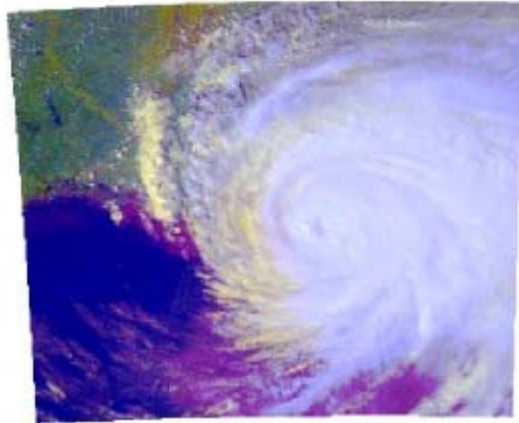
Figure G.1 — NOAA hurricane image of the Gulf of Mexico

G.2 Example 2: One Server, Three Layers, Named Styles

The following hypothetical URL requests three layers — built-up areas, coastlines, and political boundaries — to produce the map shown in Figure G.2. A transparent background has been requested for the map.

```
http://b-maps.com/map.cgi?VERSION=1.3.0&REQUEST=GetMap&
CRS=CRS:84&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=0xFFFFFF&
TRANSPARENT=TRUE&EXCEPTIONS=INIMAGE
```
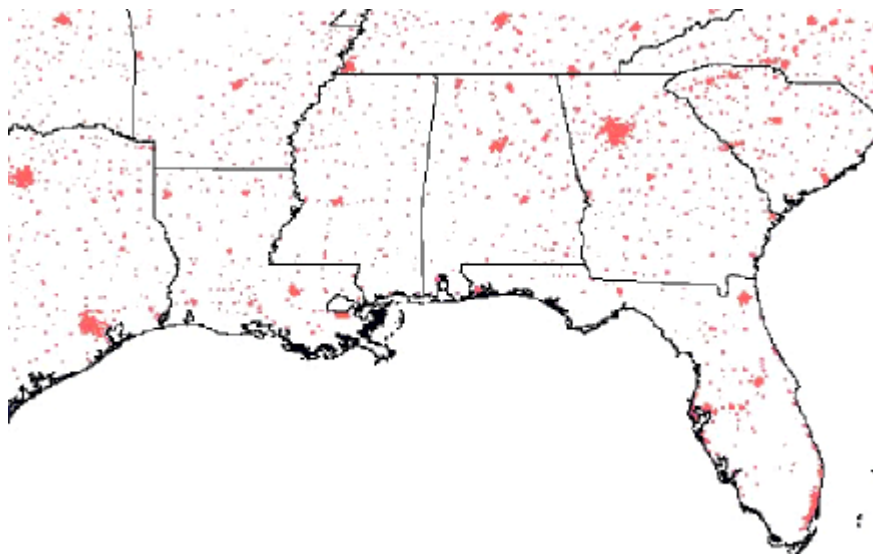

Figure G.2 — Political, coastline, and populated areas, Southeastern United States

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

```
CRS=CRS:84&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.

G.3 Example 3: Two Servers, Four Layers

Figure G.3 shows the result of overlaying Figure G.1 on Figure G.2 to produce a composite map from two separate servers.
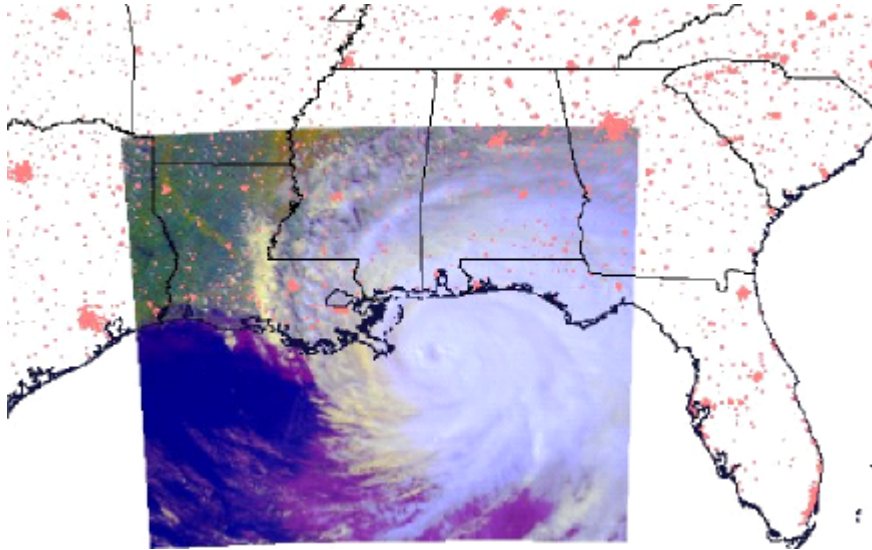

Figure G.3 — Combined hurricane image and population map

Annex H
(informative)

XML examples

H.1 Sample WMS service metadata

As an aid to understanding and a guide for implementation, this annex contains example XML which is valid according to the schema in E.1. Implementers should consult the main body of this International Standard and the schema to ensure compliance rather than editing this XML without full understanding, as the examples herein apply only to a limited range of cases.

This example XML may also be found online at <http://schemas.opengis.net/wms/1.3.0/>.

```
<?xml version='1.0' encoding="UTF-8"?>
<WMS_Capabilities version="1.3.0" xmlns="http://www.opengis.net/wms"
 xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/wms
http://schemas.opengis.net/wms/1.3.0/capabilities_1_2_0.xsd">
<!-- Service Metadata -->
<Service>
  <!-- The WMT-defined name for this type of service -->
  <Name>WMS</Name>
  <!-- Human-readable title for pick lists -->
  <Title>Acme Corp. Map Server</Title>
  <!-- Narrative description providing additional information -->
  <Abstract>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 Smith</ContactPerson>
      <ContactOrganization>NASA</ContactOrganization>
    </ContactPersonPrimary>
    <ContactPosition>Computer Scientist</ContactPosition>
    <ContactAddress>
      <AddressType>postal</AddressType>
      <Address>NASA Goddard Space Flight Center</Address>
      <City>Greenbelt</City>
      <StateOrProvince>MD</StateOrProvince>
      <PostCode>20771</PostCode>
      <Country>USA</Country>
    </ContactAddress>
    <ContactVoiceTelephone>+1 301 555-1212</ContactVoiceTelephone>
    <ContactElectronicMailAddress>user@host.com</ContactElectronicMailAddress>
  </ContactInformation>
  <!-- Fees or access constraints imposed. -->
  <Fees>none</Fees>
  <AccessConstraints>none</AccessConstraints>
  <LayerLimit>16</LayerLimit>
```

```
    <MaxWidth>2048</MaxWidth>
    <MaxHeight>2048</MaxHeight>
</Service>
<Capability>
  <Request>
    <GetCapabilities>
      <Format>text/xml</Format>
      <DCPType>
        <HTTP>
          <Get>
            <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
             xlink:type="simple"
             xlink:href="http://hostname/path?" />
          </Get>
          <Post>
            <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
             xlink:type="simple"
             xlink:href="http://hostname/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/path?" />
          </Get>
        </HTTP>
      </DCPType>
    </GetMap>
    <GetFeatureInfo>
      <Format>text/xml</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/path?" />
          </Get>
        </HTTP>
      </DCPType>
    </GetFeatureInfo>
  </Request>
  <Exception>
    <Format>XML</Format>
    <Format>INIMAGE</Format>
    <Format>BLANK</Format>
  </Exception>
  <Layer>
    <Title>Acme Corp. Map Server</Title>
    <CRS>CRS:84</CRS> <!-- all layers are available in at least this CRS -->
    <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. -->
  <CRS>EPSG:26986</CRS> <!-- An additional CRS for this layer -->
  <EX_GeographicBoundingBox>
    <westBoundLongitude>-71.63</westBoundLongitude>
    <eastBoundLongitude>-70.78</eastBoundLongitude>
    <southBoundLatitude>41.75</southBoundLatitude>
    <northBoundLatitude>42.90</northBoundLatitude>
  </EX_GeographicBoundingBox>
  <!-- The optional resx and resy attributes indicate the X and Y spatial
       resolution in the units of that CRS. -->
  <BoundingBox CRS="CRS:84"
   minx="-71.63" miny="41.75" maxx="-70.78" maxy="42.90" resx="0.01" resy="0.01"/>
  <BoundingBox CRS="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>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 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>
  <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>
<MetadataURL type="FGDC:1998">
           <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="ISO19115:2003">
          <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>
    <CRS>CRS:84</CRS> <!-- harmless repetition of common CRS -->

    <EX_GeographicBoundingBox>
      <westBoundLongitude>-180</westBoundLongitude>
      <eastBoundLongitude>180</eastBoundLongitude>
      <southBoundLatitude>-90</southBoundLatitude>
      <northBoundLatitude>90</northBoundLatitude>
    </EX_GeographicBoundingBox>
    <!-- These weather data are available daily from 1999-01-01 through
         2000-08-22. -->
    <Dimension name="time" units="ISO8601" default="2000-08-22">
      1999-01-01/2000-08-22/P1D
    </Dimension>
    <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>
          <!-- This Pressure layer is available at several elevations and times. -->
      <Dimension name="elevation" units="EPSG:5030" />
      <Dimension name="time" units="ISO8601" default="2000-08-22">
        1999-01-01/2000-08-22/P1D</Dimension>
      <Dimension name="elevation" units="CRS:88" default="0" nearestValue="1">
        0,1000,3000,5000,10000</Dimension>
    </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>
    <EX_GeographicBoundingBox>
      <westBoundLongitude>-180</westBoundLongitude>
      <eastBoundLongitude>180</eastBoundLongitude>
```

```
      <southBoundLatitude>-90</southBoundLatitude>
      <northBoundLatitude>90</northBoundLatitude>
    </EX_GeographicBoundingBox>
    <Dimension name="time" units="ISO8601" default="1992">1992</Dimension>
  </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>
    <EX_GeographicBoundingBox>
      <westBoundLongitude>-180</westBoundLongitude>
      <eastBoundLongitude>180</eastBoundLongitude>
      <southBoundLatitude>-90</southBoundLatitude>
      <northBoundLatitude>90</northBoundLatitude>
    </EX_GeographicBoundingBox>
    <Dimension name="time" units="ISO8601" default="2000">1990/2000/P1Y</Dimension>
  </Layer>
</Layer>
</Capability>
</WMS_Capabilities>
```

H.2 Sample service exception XML

As an aid to understanding and a guide for implementation, this Annex contains example XML which is valid according to the schema in E.2.

This example XML may also be found online at <http://schemas.opengis.net/wms/1.3.0/>.

```
<?xml version='1.0' encoding="UTF-8"?>
<ServiceExceptionReport version="1.3.0"
  xmlns="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/ogc
http://schemas.opengis.net/wms/1.3.0/exceptions_1_3_0.xsd">
<ServiceException>
Plain text message about an error.
</ServiceException>
<ServiceException code="InvalidUpdateSequence">
Another error message, this one with a service exception 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>
```

Bibliography

[1]   FGDC-STD-001-1998, *Content Standard for Digital Geospatial Metadata* (version 2), US Federal Geographic Data Committee. Available at: *<http://www.fgdc.gov/metadata/contstan.html>*

[2]   Internet Assigned Numbers Authority, *MIME Media Types*. from World Wide Web: *<http://www.iana.org/assignments/media-types/>*

[3]   ISO 19117:2005, *Geographic information — Portrayal*

[4]   ISO 19119:2005, *Geographic information — Services*

[5]   ISO 19125-1:2004, *Geographic information — Simple feature access — Part 1: Common architecture*

[6]   LALONDE, W. (ed.), *Styled Layer Descriptor Implementation Specification 1.0.0*, OGC Document #02-070, August 2002. Available at *<http://www.opengis.org/techno/specs/>*

[7]   VRETANOS, P. (ed.), *Web Feature Service Implementation Specification 1.0.0*, OGC Document #02-058, September 2003. Available from World Wide Web: *<http://www.opengis.org/techno/specs/>*

[8]   XML Schema (May 2001), *XML Schema Part 2: Datatypes*, World Wide Web Consortium Recommendation, Biron, P. and Malhotra, A., eds., Available at *<http://www.w3.org/TR/>*

[9]   ISO 6709:1983, *Standard representation of latitude, longitude and altitude for geographic point locations*

[10]   ISO/IEC 8632 (all parts), *Information technology — Computer graphics — Metafile for the storage and transfer of picture description information*

[11]   ISO/IEC TR 14252:1996, Information technology — Guide to the POSIX Open System Environment (OSE)