



DGWIG - 104

DGIWG Profile of JPEG2000 for Georeferenced Imagery

Document type:	Standard
Document date:	4 February 2014
Edition:	1.0.0
Responsible Party:	DGIWG
Audience:	This document is approved for public release and is available on the DGIWG website, http://www.dgiwg.org/dgiwg/
Abstract:	This document provides a profile for JPEG2000 for use as a compression format for raster imagery. JPEG2000 uses <i>discrete wavelet transform (DWT)</i> for compressing raster data, as opposed to the JPEG standard, which uses <i>discrete cosine transform (DCT)</i> . It is a compression technology which is best suited for continuous raster data, such as satellite imagery and aerial photography.
Copyright:	(C) Copyright DGIWG, some rights reserved - (CC) (By:) Attribution You are free: <ul style="list-style-type: none">- to copy, distribute, display, and perform/execute the work- to make derivative works- to make commercial use of the work Under the following conditions: <ul style="list-style-type: none">- (By:) Attribution. You must give the original author (DGIWG) credit.- For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the copyright holder DGIWG. Your fair use and other rights are in no way affected by the above. This is a human-readable summary of the Legal Code (the full license is available from Creative Commons < http://creativecommons.org/licenses/by/2.0/ >).

Table of Contents

1	Scope	2
2	Conformance	2
3	Normative references.....	2
3.1	ISO Standards	2
3.1.1	JPEG2000	2
3.1.2	Metadata.....	2
3.1.3	Embedded image annotations	2
3.1.4	Video annotations.....	2
3.2	OGC Standards	2
3.2.1	GMLJP2	2
3.2.2	GML.....	3
3.3	W3C Standards	3
3.3.1	SVG.....	3
3.3.2	CSS2.....	3
3.4	DGIWG and US national Specifications	3
3.4.1	The ARC System.....	3
3.4.2	NIMA TR 8350.2.....	3
3.4.3	NIMA TM 8358.2.....	3
4	Terms, definitions, and abbreviations.....	3
4.1	Definitions.....	3
4.1.1	Codestream [ISO/IEC 15444-1].....	3
4.1.2	Compression ratio [Taubman and Marcellin, 2002, 8].....	3
4.1.3	Bit-rate [Taubman and Marcellin, 2002, 8].....	3
4.1.4	Wavelet Transform [Taubman and Marcellin, 2002, 247].....	4
4.1.5	Discrete Wavelet Transform [Taubman and Marcellin, 2002, 247].....	4
4.1.6	Rate Allocation [Taubman and Marcellin, 2002, 209].....	4
4.1.7	Decomposition levels [ISO/IEC 15444-1].....	4
4.1.8	Resolution [ISO/IEC 15444-1].....	4
4.1.9	Precinct [ISO/IEC 15444-1].....	5
4.1.10	Packet [ISO/IEC 15444-1].....	5
4.1.11	Code-block [ISO/IEC 15444-1].....	5
4.1.12	Top level [OGC GMLJP2]	5
4.2	Abbreviations.....	5
5	Overview of the JPEG2000 standard	6
5.1	JPEG2000 Part 1.....	6
5.2	JPEG2000 Part 2.....	6
5.3	Codestream	6
5.3.1	Compression of the codestream	6
5.3.2	Partition of the codestream	6
5.3.3	Codestream information.....	8
5.3.4	Error resilience.....	9
5.3.5	The EBCOT algorithm.....	10
5.3.6	Progression order	10
5.3.7	JPEG2000 Profiles	10
5.3.8	Multiple codestreams.....	12
5.3.9	JPIP.....	12
5.4	JPEG2000 Box	12
5.4.1	Box structure	13
5.4.2	Superbox	13
5.5	Opacity	13
5.5.1	Opacity channels (JPEG2000 Part 1)	13
5.5.2	Opacity in a separate codestream (JPEG2000 Part 2).....	13
5.5.3	Opacity box (JPEG2000 Part 2).....	14

5.6	Integer values constraint of raster and gridded data.....	14
6	GML objects.....	14
	<i>(Informative)</i>	14
6.1	Coverage.....	14
6.2	Coordinate reference system.....	14
6.3	Axis order.....	15
6.4	Annotation.....	15
6.5	Feature and annotation styling.....	15
6.6	Metadata.....	15
7	Embedding GML in JP2 files.....	15
7.1	JPX file format.....	16
7.2	Compatibility list in File type box.....	16
7.2.1	If opacity of Part 2 type is used.....	16
7.2.2	Profile.....	16
7.3	Brand field in File type box.....	16
7.4	Reader requirement box.....	16
7.5	File suffix.....	16
8	Annotations.....	17
8.1	Annotation conformance classes.....	17
8.1.1	Class A.....	17
8.1.2	Class XA.....	17
Annex A	19
A.1	Introduction.....	19
A.1.1	Test classes.....	19
A.2	Class B tests.....	19
A.2.1	Georeferenced with GML according to GMLJP2.....	19
A.2.2	GML code passes XML schema validation.....	19
A.2.3	GML code has RectifiedGridCoverage.....	19
A.2.4	The RectifiedGrid contains all required content.....	20
A.2.5	The number of grid cells is the same as the number of pixels.....	20
A.2.6	Coordinate lists in GML, have sufficient number of coordinates.....	21
A.2.7	The RectifiedGrid refers to a reference system.....	21
A.2.8	Left-hand order coordinate system is used for WGS84 LL coordinates.....	22
A.2.9	Image coordinate system gives the coordinates in the correct order.....	22
A.2.10	Metadata extent is coherent with the GML extent.....	22
A.2.11	Security classification in metadata.....	23
A.2.12	Security classification in Intellectual property rights box.....	23
A.2.13	References to embedded codestreams and XML instances are valid.....	23
A.2.14	The Reader Requirement box signals the use of GML.....	24
A.2.15	Precinct partition size.....	24
A.2.16	Brand field.....	24
A.2.17	Compatibility list.....	25
A.2.18	File suffix.....	25
A.3	Class A tests.....	25
A.3.1	Allowed style elements.....	25
A.3.2	References to embedded symbols are valid.....	25
A.4	Class XA tests.....	26
A.4.1	References to embedded images and video streams are valid.....	26
Annex B	27
B.1	Purpose.....	27
B.2	JPEG2000 Profiles.....	27
B.3	Conformance classes.....	27
B.3.1	Class B – Baseline.....	27
B.3.2	Class A – GMLJP2 with Annotations.....	27
B.3.3	Class XA – GMLJP2 with extended annotations.....	28

B.4	Implementation Conformance Report.....	28
Annex C	32
C.1	Required boxes.....	32
C.1.1	JPEG 2000 Signature box	32
C.1.2	File type box	32
C.1.3	Reader requirement box	32
C.1.4	JPEG 2000 header box.....	33
C.1.5	Intellectual property rights box	36
C.1.6	Codestream box	38
C.2	Annotation boxes	39
C.2.1	Image annotation boxes.....	39
C.2.2	Video clip box	39
C.2.3	UUID boxes	39
Annex D	41
D.1	GML objects.....	41
D.1.1	GML box structure	41
D.1.2	gml.root-instance	41
D.1.3	GMLJP2 URI	42
D.1.4	DGIWGJP2 URI.....	42
D.2	GML Georeferencing	42
D.2.1	GML structure.....	43
D.2.2	RectifiedGridCoverage.....	43
D.2.3	EPSG georeference in RectifiedGrid@srsName	49
D.2.4	CRSDictionary.gml	49
D.3	Annotation	49
D.3.1	Annotation elements	50
D.3.2	Annotation types.....	50
D.3.3	Content of the annotation.....	50
D.3.4	Annotation styling	51
D.4	Metadata.....	54
Annex E	55
E.1	Metadata location	55
E.2	Metadata requirements.....	55
E.2.1	Metadata conforms to DMF.....	55
E.2.2	Coverage information in metadata	55
E.3	Security classification.....	55
Annex F	56
F.1	GML Schemas	56
F.2	The DGIWG GMLJP2 schemas	56
F.2.1	DGIWGgmlJP2Profile.xsd.....	56
F.2.2	DGIWGannotation.xsd	56
Annex G	57
G.1	Use cases.....	57
Annex H	59
H.1	Extensions used by DGIWG Profile	59
H.1.1	Extensions required by baseline GMLJP2.....	59
H.1.2	Additional extensions used by DGIWG GMLJP2.....	59
Annex I	61
I.1	Software recommendations	61
I.2	Kdu_show	61
I.3	XML Spy.....	61
I.3.1	XML Schema reference	61
I.4	Microsoft .Net Framework 4.0	61

List of Figures

Figure 1 – Marker segment structure	8
Figure 2 – JPEG2000 box structure	13
Figure 3 – Left and right hand order	15
Figure C.1 – Annotation box structure	39
Figure C.2 – UUID box	40
Figure D.1 - GML box structure	41
Figure D.2 – GML structure	43
Figure D.3 – Image coordinate system	44

List of Tables

Table 4.1 – Typical bit-rates at different quality levels	4
Table 5.1 – Items in a Coding style default structure	9
Table 5.2 – Items in a Coding style component structure	9
Table 5.3 – List of progression orders	10
Table 6.1 – EPSG codes	14
Table 7.1 – File suffixes	16
Table 8.1 – Examples of <i>DGIWGJP2 URI:s</i>	18
Table B.1 – JPEG2000 Profiles	27
Table B.2 – Implementation Conformance Report	31
Table C.1 – File type box	32
Table C.2 – Reader requirement box	33
Table C.3 – Image header box	34
Table C.4 – Colour specification box	35
Table C.5 – Channel definition box	36
Table C.6 – IPR fields	37
Table C.7 – Security classification keywords	37
Table C.8 – UUID fields	40
Table C.9 – File types	40
Table D.1 – Required RectifiedGrid content	45
Table D.2 – GML extent in ISO 19139 metadata	45
Table D.3 – Axis order for different reference systems	45
Table D.4 – Required file type content	46
Table D.5 – Reference codes from the EPSG namespace	49
Table D.6 – Annotation elements	50
Table D.7 – Annotation types	50
Table D.8 – Label elements	51
Table D.9 – Image elements	51
Table D.10 – Video element	51
Table D.11 – Styling features	52
Table D.12 – Overview of styling options for different annotations	52
Table D.13 – Attribute for the gml:FeatureStyle element	52
Table D.14 – Attributes for the gml:geometryStyle element	53
Table D.15 – Attributes for the gml:symbol element	53
Table E.1 – DMF requirement classes	55
Table F.1 – Schema locations	56
Table G.1 – Use cases	58

i. Contributors

Nation	Organization
Sweden	Swedish Armed Forces Geo SE
France	Institut Géographique National
United States	National Geospatial-intelligence Agency
Czech republic	Czech republic – Military
Germany	Bundeswehr
European Union	EUSC
Italy	Italian Ministry of Defence

ii. Document point of contact

The point of contact for this document is the DGIWG Secretariat, secretariat@dgiwg.org

iii. Revision history

Date	Release	Primary clauses modified	Description
2011-03-10	0.1.0	All	First working draft
2011-06-30	0.2.0	Annex G	Added appendix with use cases
2011-10-26	0.3.0	All	Worked with the metadata and added TOC for figures and tables.
2011-12-09	0.4.0	All	Resolved comments by FR and CZ; and added annex about which parts of Part 2 the profile is using
2012-03-23	0.5.0	8, Annex A, Annex B, Annex C, Annex D	Added annotation support for embedded images and embedded MPEG-4 video. Added conformance classes.
2012-06-08	0.6.0	6, Annex A, Annex C, Annex E	Added requirements regarding axis order. Specified an additional way for storing security classification. Removed specification of metadata elements.
2013-01-16	0.7.0	All	Resolved comments by FR.
2013-05-17	0.8.0	All	Resolved comments by US, CZ, IT and FR. Changed the mechanism for embedding files.
2013-09-02	0.8.1	All	Resolved comments by FR and DEU.
2013-09-19	0.8.2	Annex D	Corrected an error about the attributes for GeometryStyle. The attribute geometryStyle is renamed to geometryType, as it is in GML 3.1.1.
2013-12-17		All	Resolved comments by FR

Date	Release	Primary clauses modified	Description
2013-12-20		Annex D	Clarified a figure which illustrates image coordinates

iv. Future work

- The OGC GMLJP2 uses GML v3.1.1. The most recent accepted standard is GML v3.2.1. The differences mean that GML v3.2.1 is **not** backward compatible with GML v3.1.1. OGC will most likely release a new version of GMLJP2 which uses a more recent version of GML. When that happens, this profile will be adapted to GML v3.2.1. Until then, this GMLJP2 profile will use GML v3.1.1.

- There is a need for JPEG2000 for sensor imagery which has not been rectified (i.e. ReferenceableGridCoverage), which requires to encode the sensor and image geopositioning parameters. SensorML is an OGC language which can describe different aspects of the sensor that captured an image and is the envisioned solution for encoding these parameters.

Introduction

JPEG2000 is a compression format for raster imagery. JPEG2000 uses *discrete wavelet transform (DWT)* for compressing raster data, as opposed to the JPEG standard, which uses *discrete cosine transform (DCT)*. It is a compression technology which is best suited for continuous raster data, i.e. satellite imagery, aerial photography, etc. JPEG2000 images are commonly compressed in a lossy fashion, but there is a lossless compression mode that can be used.

DWT offers better compression than DCT in several ways. The compression ratio is higher. The artifacts aren't as disturbing as in the DCT case. The JPEG2000 format offers many capabilities allowing derivatives of the image to be retrieved in low bandwidth situations. These options include reduced resolution levels, low bit-rate layers (quality layers), and independently encoded blocks allowing access to regions of interest.

Geography Markup Language (GML) Encoding Standard is used for the georeference of the image and for optional metadata, annotation and geographic features that may be provided with the image.

1 Scope

This JPEG2000 profile is developed for the use of georeferenced JPEG2000 files on the basis of the GMLJP2 standard. The standard covers raster data, including orthoimagery and elevation.

2 Conformance

JPEG2000 files must follow Annex A, B and C to conform to this profile. Additionally, the GMLJP2 files can follow different conformance classes. Three conformance classes are specified:

- Class B: Baseline profile – A JPEG2000 file georeferenced according to the constraints defined by this GMLJP2 profile.
- Class A: Annotation conformance class – Support for simple annotations (i.e. text, points, lines, polygons and symbols). The points, lines and polygons use GML elements. The symbols are in SVG format.
- Class XA: Extended annotations conformance class – Support for embedded image files and embedded video.

Class A and XA inherit from Class B.

3 Normative references

3.1 ISO Standards

3.1.1 JPEG2000

ISO/IEC 15444-1:2004 (JPEG2000 Part 1 – core coding system)

ISO/IEC 15444-2:2004 (JPEG2000 Part 2 – extensions)

The latter standard is required for the *label* and *association* boxes, specified by JPX format (Annex M of Part 2 – M.11.11 and M.11.13 resp.), which are required by GMLJP2.

3.1.2 Metadata

ISO 19115-2 – Metadata information

ISO 19139-2 – XML Schema implementation of ISO 19115

DMF – DGIWG Metadata Foundation

3.1.3 Embedded image annotations

ISO 10918-1:1993 – JPEG

3.1.4 Video annotations

ISO 14496-14:2003 – MPEG-4 encoded video

3.2 OGC Standards

3.2.1 GMLJP2

OGC 05-047r3 - GML in JPEG 2000 for Geographic Imagery (GMLJP2) Encoding Specification

3.2.2 GML

OGC 03-105r1 – GML 3.1.1, which is used in current GMLJP2 standard.

3.3 W3C Standards

3.3.1 SVG

Scalable Vector Graphics (SVG) 1.1 (Second Edition)

3.3.2 CSS2

Cascading Style Sheets, level 2 CSS2 Specification

3.4 DGIWG and US national Specifications

3.4.1 The ARC System

The ARC System as defined in DIGEST Support Document 3 on http://www.dgiwg.org/DGIWG_Geodetic_Codes/

3.4.2 NIMA TR 8350.2

NIMA TR 8350.2: Third Edition, Amendment 1: Department of Defense World Geodetic System 1984; Its Definition and Relationships with Local Geodetic Systems. U. S. National Imagery and Mapping Agency 3 January 2000.
<http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>

3.4.3 NIMA TM 8358.2

NIMA TM 8358.2: The Universal Grids: Universal Transverse Mercator (UTM), Universal Polar Stereographic (UPS)
http://earth-info.nima.mil/GandG/publications/tm8358.2/TM8358_2.pdf

4 Terms, definitions, and abbreviations

4.1 Definitions

4.1.1 Codestream [ISO/IEC 15444-1]

A collection of one or more bit streams, and the main header, tile-part headers, and the EOC required for their decoding and expansion into image data. This is the image data in a compressed form with all of the signaling needed to decode.

4.1.2 Compression ratio [Taubman and Marcellin, 2002, 8]

A measure of the compression. It is the ratio between the uncompressed original and the compressed file. If $N1 = \text{image height in pixels}$, $N2 = \text{image width in pixels}$, $B = \text{bits per sample (in the uncompressed image)}$ and $c = \text{codestream length}$, the compression ratio is given by:

$$\text{compression ratio} = (N1 N2 B) / c$$

4.1.3 Bit-rate [Taubman and Marcellin, 2002, 8]

A measure of the compression. In literature about JPEG2000, bit-rate is abbreviated *bps (bits per sample)*¹. Bps is given by:

¹ *Bps* is a common abbreviation for *bits per second*, which is also a bit-rate. The context gives the correct meaning.

$$Bps = c / (N1 N2)$$

This measure doesn't take the original image's number of bits per sample into account. This means that it is impossible to deduce compression ratio from the bit-rate. But the bit-rate gives a measure for compression performance.

Lossless		B – 3 bps
	High	1 bps
Lossy	Moderate	0.5 bps
	Usable	0.25 bps

Table 4.1 – Typical bit-rates at different quality levels

In table 4.1, B represents the bitrate in the uncompressed original. The table in itself is from *JPEG2000 – Image Compression Fundamentals, Standards and Practice*, David S. Taubman, Michael W. Marcellin. The table gives only rough estimates of the quality. The assumption is that the image is viewed on a 90 ppi (pixels per inch) computer monitor. A higher resolution on the display device, would allow for a higher compression ratio, i.e. lower bit-rate, since it would be more difficult for the eye to discern the details.

4.1.4 Wavelet Transform [Taubman and Marcellin, 2002, 247]

Wavelet transform is a mathematic tool for extracting information from different kinds of data that can be represented as continuous signals. That includes continuous raster data, such as satellite photos. The signal is projected on a number of frequency bands. In the raster data context, a frequency could be measured as *lines per mm* (or any other length unit). The original signal can be restored by integration over the frequency components.

4.1.5 Discrete Wavelet Transform [Taubman and Marcellin, 2002, 247]

A transformation that iteratively transforms one signal into two or more filtered and decimated signals corresponding to different frequency bands. This transformation operates on spatially discrete samples.

4.1.6 Rate Allocation [Taubman and Marcellin, 2002, 209]

Individual compression for a specific code-block in order to achieve a particular bit-rate.

4.1.7 Decomposition levels [ISO/IEC 15444-1]

A collection of wavelet sub-bands where each coefficient has the same spatial impact or span with respect to the source component samples. These include the HL, LH, and HH sub-bands of the same two dimensional sub-band decomposition. For the last decomposition level the LL sub-band is also included.

4.1.8 Resolution [ISO/IEC 15444-1]

The spatial mapping of samples to a physical space. In ISO 15444-1, the decomposition levels of the wavelet transform relate to each other with relative resolutions differing by powers of two.

4.1.9 Precinct [ISO/IEC 15444-1]

A rectangular region of a transformed tile-component within each resolution level used for limiting the size of packets.

4.1.10 Packet [ISO/IEC 15444-1]

A part of the bit stream comprising a packet header and the compressed image data from one quality layer of one precinct.

4.1.11 Code-block [ISO/IEC 15444-1]

A rectangular grouping of coefficients from the same sub-band of a tile-component.

4.1.12 Top level [OGC GMLJP2]

The GML structure in a GMLJP2 file uses a hierarchical information structure. The top level denotes the top level in that hierarchy.

4.2 Abbreviations

bps	Bits per sample
CRS	Coordinate reference system.
CSS2	Cascading Style Sheets, level 2
DCT	Discrete Cosine Transform.
DMF	DGIWG Metadata Foundation
DWT	Discrete Wavelet Transform.
EBCOT	Embedded Block Coding with Optimal Truncation
EOC	End Of Codestream
GML	Geography Markup Language.
GMLJP2	OGC's standard for embedding GML code in JPEG2000 files.
ICC profile	Colour profile according to <i>International Color Consortium</i> . A colour profile defines how the colours shall be represented on a specific device.
ISO	International Standardization Organization.
JP2	File format specified in JPEG2000.Part 1 (ISO/IEC 15444-1) in its Annex I.
JPEG	Joint Photographic Expert Group and a file format using DCT.
JPEG2000	Standards compression using DWT, as opposed to JPEG, which is using DCT.
JPX	File format specified in JPEG2000.Part 2 (ISO/IEC 15444-2) in its Annex M.
MPEG	Motion Picture Experts Group
MPEG-4	Video format from MPEG
OGC	Open GIS Consortium.
PCRD	Post Compression Rate Distortion
SVG	Scalable Vector Graphics

5 Overview of the JPEG2000 standard

(Informative)

JPEG2000 is a raster data compression method which uses DWT. The compression method is best suited for continuous raster data. The compression can be either lossy or lossless, but lossless compression is the less efficient method.

5.1 JPEG2000 Part 1

JPEG2000 Part 1 (Core Coding System) is described in ISO/IEC 15444-1. The JPEG Committee has striven to ensure that implementations of Part 1 can be royalty and licence-fee free.

5.2 JPEG2000 Part 2

ISO/IEC 15444-2 (Part 2) contains extensions to the JPEG2000 standard. This part is described in ISO/IEC 15444-2. GMLJP2 uses a few extensions included in this part. License and royalties may be required for use of some technologies described in Part 2. The DGIWG profile strives for using extensions that do not require royalty or license-fees. Annex H lists the extensions that are used by the DGIWG profile.

5.3 Codestream

One important property in the JPEG 2000 codestream, which is the compressed raster data, is the possibility to locate and extract data without fully decoding the codestream. This allows extraction of data from the compressed codestream to form a reconstructed image with lower resolution, lower bitrate or a region of the image. A codestream could therefore be efficiently used by even small image devices.

5.3.1 Compression of the codestream

The codestream can be compressed by an arbitrary compression ratio, and even lossless. When encoding elevation data, it is critical to preserve a high precision in the data. Therefore it is recommended to use lossless compression or a low compression ratio for elevation data so that a high precision in the data is maintained.

When encoding imagery, it may not be critical to preserve a high precision in order to extract useful information from the image. Therefore a higher compression ratio may be used in that case.

5.3.2 Partition of the codestream

JPEG2000 codestreams can be partitioned in several different ways.

The codestream is divided in a number of steps:

1. Decorrelated into components if it is multi-band data. Otherwise it is only one component.
2. Components are divided into tiles.
3. The components of the tiles are wavelet transformed with N_L decomposition levels into sub-bands. Each decomposition level gives a low-pass band representing a low resolution version, and a high-pass band representing the residual between the low resolution version and the original version. By decomposing the low resolution band from the first decomposition, levels with even lower resolutions are given. Thus, each decomposition adds one more

resolution level without increasing the amount of information. As a result, each tile is available at $N_L + 1$ distinct resolutions.

4. The resolution levels are divided further into precincts.

5. The sub-bands from step 3 are divided into code-blocks, which are coded independently of each other.

The different resolutions are denoted as *sub band* LL_0 , which is the highest spatial frequency, and thus, the highest resolution, continuing this way through the resolution levels LL_1 , and onwards, to LL_D , which is the lowest resolution level.

5.3.2.1 Components

A component covers the whole image area. But the component contains only one colour. Each component is separated from the image before any other partitioning.

5.3.2.2 Tiles

Tiles are relatively large partitions of the image. The tiles can be compressed with individual parameters for each tile. It is possible to have an image where some of the tiles are not compressed at all, or some tiles are compressed with relatively low degradation of the image. A lot of image formats use tiling in order to reduce memory usage by only handling small parts of the image. It is possible to do that with the tile mechanism in JPEG2000, but it isn't necessary, since the organization of the image in precincts enables direct access to parts of the image.

5.3.2.3 Sub-bands

A tile is decomposed into the sub-bands HH (horizontally and vertically high pass), HL (horizontally high pass and vertically low pass), LH and LL .

5.3.2.4 Decomposition levels

The LL sub-band from the low pass filtering, gives a resolution level with half the resolution of the image that was filtered. The other sub-bands are discarded.

5.3.2.5 Code-blocks

Code-blocks are coded independently of each other. Common sizes are 32x32 and 64x64 pixels. The fact that the code-blocks are independently encoded, gives some advantages, among them are the fact that many code-blocks can be processed simultaneously. That means JPEG2000 processing scales well when using multiple CPUs or multi-core CPUs. Since it today seems like the most effective way to increase performance, is to use more cores in the CPUs, this is an important property in JPEG2000. The code-blocks are also used for *Rate-Allocation*. *Rate-allocation* chooses the contributions of each code-block such that an aimed bit-rate is achieved. Code-blocks are essential for the EBCOT paradigm.

5.3.2.6 Precincts

Precincts organize the compressed data in the codestream. Precincts do not affect the coding of the data. Therefore, the precincts enable direct access to parts of the image. Precinct dimensions must always be a power of two. One precinct contributes one *packet* for each *quality layer*.

5.3.2.7 Packets

The compressed data in the codestream is organized in packets. Packets are the smallest partitions of the codestream. Each precinct has one packet for every quality layer. A packet contains contributions up to that quality layer from each code-block within that precinct. Note that the contributions to any particular layer might be empty.

5.3.3 Codestream information

The codestream itself contains information for decoding the codestream.

5.3.3.1 Headers

Headers are collections of markers and marker segments, described in paragraphs. The headers are either *main header*, which has a scope encompassing the complete codestream or *tile-part header*, which has a scope encompassing a separate tile, thereby giving the option of having different compression parameters for individual tiles.

5.3.3.2 Markers and marker segments

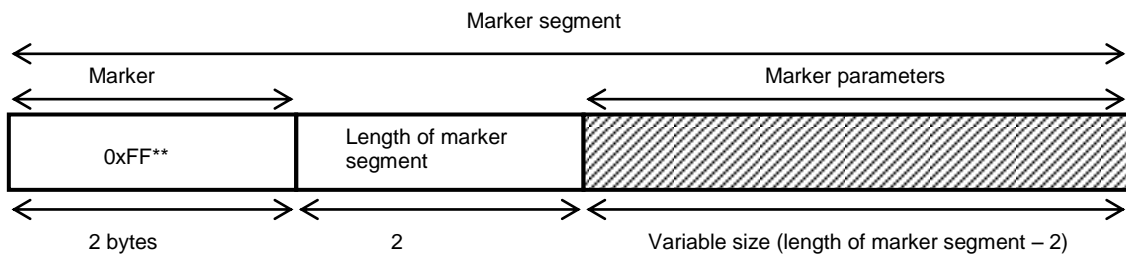


Figure 1 – Marker segment structure

The codestream contains markers for locating information in it. A marker consists of a two-byte word, 0xFF<marker code>. The first byte is always 0xFF. The second byte is the marker code. A marker segment includes a marker and parameters associated with that marker. If there are associated parameters, the marker must be followed by a two-byte word that gives the size of the marker segment, including the size, but excluding the marker. A marker without any parameters, doesn't have a *marker length* either, and is thus only two bytes.

5.3.3.3 COD - Coding style default and COC – Coding style component

The COD marker segment contains information about the encoding of an image. It is possible to override the image COD with a COD for a specific tile or a COC for a specific component or tile. The COD describes:

Items
Number of decomposition levels
Progression order
Number of quality layers
Codeblock size
Codeblock style
Transform
Multiple component transform
Packet partition size

Table 5.1 – Items in a Coding style default structure

The COC describes:

Items
Number of decomposition levels
Codeblock size
Codeblock style
Transform
Packet partition size

Table 5.2 – Items in a Coding style component structure

If more than one of the markers is used, it's always the most local of the blocks that are used. The precedence is as:

Tile COC > tile COD > main COC > main COD

5.3.4 Error resilience

Error resilience is achieved through the use of two optional markers in the codestream; SOP (Start of packet) and EPH (End of packet header). The idea is that if a codestream is interrupted, it shall be possible to resynchronize with the codestream with the help of the SOP and EPH markers, even though it is impossible to reconstruct everything. The use of SOP or EPH markers is indicated by a COD (codestream default) marker, which can appear both in the main header or the tile header.

5.3.4.1 SOP – Start of packet

The SOP marker carries a two byte counter which starts from 0 in every tile. If it rolls over, i.e. gets larger than 65535, it will start over from 0 again. As the name implies, there should be one marker for each packet. But the codestream is allowed to skip SOP markers for some packets. If some SOP markers are omitted, the next SOP marker must have its counter incremented as if the omitted SOP markers actually did exist for each packet. If a COD indicates the use of SOP markers, the markers won't necessarily appear within the scope of the COD. SOP markers are not allowed when the COD indicates that they are not used.

The use of SOP markers may require as many as six bytes for every packet, but the markers are not necessary for every packet.

5.3.4.2 Payload

The error resilience markers may use as many as eight bytes for each packet. Packet sizes vary. Packet sizes can be expected to be as small as about 100 bytes but most often they will be several KB in size. Therefore it is reasonable to assume that the payload for the error resilience markers will be negligible.

5.3.5 The EBCOT algorithm

The EBCOT (Embedded Block Coding with Optimized Truncation) algorithm is used for fast access of arbitrary regions with arbitrary resolution. The EBCOT algorithm uses the code-blocks to access regions of the image. There are also quality layers, to which the code-blocks contribute up to a specific truncation point.

5.3.5.1 PCRD-opt

Rate-allocation is done with Post-Compression Rate Distortion optimization, which is an algorithm for assigning truncation points in a code-block minimizing the distortion for a specific bit-rate. The truncation points are used for *quality layers*.

5.3.5.2 Quality layers

A *quality layer* is an abstract form of dividing the image. A *quality layer* is made up of the contributions from all code-blocks up to their respective truncation points for each *quality layer*. The PCRD optimization means that the quality layers differ in bit-rates, as opposed to *decomposition levels*, which differs in resolution. Increasing the number of quality layers, means increasing the overhead needed for the truncation points. The overhead consists of length tags inserted into the code-blocks in order to identify its contribution to a *quality layer*.

5.3.6 Progression order

The progression order determines the order of the packets in a specific tile.

Progression order	Description
LRCP	Layer-Resolution-Component-Position progression. Progressive by quality. Quality increases for each layer.
RLCP	Resolution-Layer-Component-Position progression. Progressive by resolution. The outermost loop gets a new resolution level for each stage in the loop.
RPCL	Resolution-Position-Component-Layer progression. Progressive by resolution.
PCRL	Position-Component-Resolution-Layer progression. Progressive by position.
CPRL	Component-Position-Resolution-Layer progression. Progressive by component.

Table 5.3 – List of progression orders

5.3.7 JPEG2000 Profiles

A JPEG2000 reader isn't required to be able to recover all of the information in a codestream. But there is a need to ensure that the reader can recover at least some of the information in the codestream. Therefore, the JPEG2000 standard includes a number of profiles with different restrictions of the parameters.

There are five different JPEG2000 profiles according to ISO/IEC 15444-1 to this date. The profiles are applied to the codestream. Profile 3 and Profile 4 are profiles for moving images. These profiles are outside the scope of this work. JPEG2000 codestreams are allowed to follow Profiles 0, 1 and 2 according to the DGIWG profile.

5.3.7.1 Profile-0

Profile-0 is developed for low complexity applications.

- Tiles must be squares with maximum size 128x128 pixels. Only exception is when only one tile for the whole image is used.
- Code-block size is restricted to 32 or 64 in both directions. The value must be same in both directions.
- LL_D must not exceed a spatial resolution above $w = 128$ and $h = 128$. Same restriction applies to *Profile-1*.

Profile-0 excludes features that are necessary for error detection. The restrictions also limit the possibility to optimize the image for rapid access to different image elements or for displaying the image on different devices.

5.3.7.2 Profile-1

Profile-1 is meant to be both easy to implement and allowing high performance at the same time.

- Tiles must be squares with a maximum size 1024x1024. The only exception is when the entire image is composed of only one tile.
- Code-block size is restricted to a maximum value of 64 in both directions.
- LL_D must not exceed a spatial resolution above $w = 128$ and $h = 128$. Same restriction applies to *Profile-0*.

Software implementations are strongly encouraged to support at least Profile-1. Therefore, interoperability is maximized by conforming to Profile-1.

5.3.7.3 Profile-2

No restrictions, i.e. all elements defined by ISO/IEC 15444-1 can be used. If there is no indication of profile, the codestream conforms to Profile-2.

5.3.7.4 Implications of the LL_D restriction

The *DWT* is always performed on a tile. Therefore, LL_0 , the maximum resolution level, will never exceed the tile size. If the image is divided into only one tile, that is the image size. In that case, the minimum resolution level, LL_D , will be the size of the smallest "pyramid layer". If the image is divided into several tiles, LL_D will be the size of the smallest "pyramid layer" of a tile. The whole image will be larger even at that resolution level. If the image conforms to *Profile-1*, the image could be divided into 1024x1024 tiles. In that case, all images would be required to have 5 decomposition levels, since the $h \leq 128$ restriction would mean that $LL_D = 128 \times 128$ isn't sufficiently low (the fourth level). But if the image is divided into only one tile, which is allowed by all profiles, the number of required decomposition levels will vary depending on the image's total size.

5.3.7.5 Recommendation to use Profile-1

Since there is a recommendation that software implementations shall support Profile-1, it is also recommended that a codestream conforms to Profile-1. But

certain applications can benefit from parameters outside the restrictions imposed by Profile-1. Therefore a codestream doesn't have to conform to Profile-1.

5.3.8 Multiple codestreams

JP2 format allows multiple codestreams, even though a JP2 compliant reader must ignore the other codestreams. The use of multiple codestreams requires a JPX capable reader.

5.3.8.1 Compositing layer

JPEG2000 Part 2 introduces *compositing layers*. A compositing layer is constructed from a number of codestreams in a way defined in the JPEG2000 file. Components in the codestreams are mapped to specific components in the compositing layer, which is viewed as one separate image. If this option is used, a reader which is compliant with only Part 1 will only be able to read the first codestream.

5.3.9 JPIP

JPIP (ISO/IEC 15444-9) is a protocol for streaming JPEG2000 data over a network. This protocol allows communication between the client and the server when only small portions of the codestream are transferred. If the client requests a large overview, the server will transfer a low resolution version adapted to the requested resolution. If the client requests a particular area of the image, the server will only transfer that area. Therefore, even gigapixel sized images will be usable over a network even with a relatively low bandwidth. JPIP is also used for progressively display an image while waiting for it to be completely downloaded. JPIP relies on the partition of the image into low resolution decomposition levels and precincts. JPIP can be used over IP, TCP, UDP and HTTP.

5.3.9.1 Data-bin

JPIP organizes the information in *data-bins*. Data-bins can contain precincts, tiles or metadata. It is possible to request only metadata². The data-bins that contain image elements, such as precincts and tiles, contain parts of the codestream box, and not the complete codestream box. The data-bins that contain metadata can contain any collection of boxes. A JPEG2000 box is the smallest element a metadata-bin can contain. That means it is possible to transmit multiple JPEG2000 boxes over multiple independent channels, i.e. one JPEG2000 box per channel. This allows out of band transmission to speed up metadata throughput.

5.4 JPEG2000 Box

The information in a JPEG2000 file is organized in boxes. A box can contain the codestream or information about the file.

² Here, "metadata" refers to any collection of "boxes" from a JPEG2000 family file (ISO/IEC 15444-9 A.3.6.1).

5.4.1 Box structure

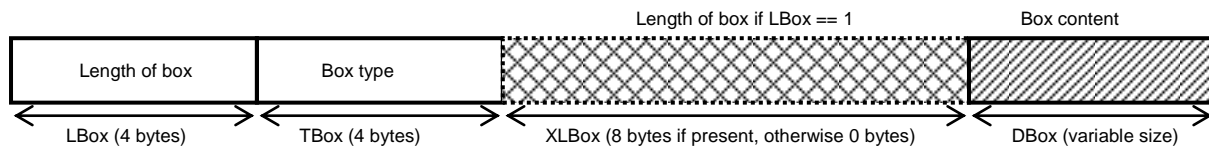


Figure 2 – JPEG2000 box structure

The box consists of:

LBox: 4-byte unsigned integer. The length of the box. This field and the box type field are included. Possible values that cannot be interpreted as box sizes, since the values are too small:

0: box length is unknown, which means that the box contains everything up to the end of file. A typical example of this is the *'jp2c'*-box in many files. If a box contained within a superbox, has this value, when the superbox size is implicitly unknown, and should therefore also have the value 0.

1: the box real length is in the *XLBox* field described below.

TBox: Actually defined as a 4-byte unsigned integer, but in practice, they are always referred to by a character string translation of the integer value. A space character is often represented as "\040" (the octal representation of its ordinal number).

XLBox: 8-byte unsigned integer. Extended field for the length. The sizes of the LBox, TBox and XLBox, are included. This field is useful if the length is larger than 4GB.

DBox: Variable length and format. This field contains the actual information in the box.

5.4.2 Superbox

The *superbox* is a box that contains other boxes. This is a mechanism that is used extensively in the *GMLJP2* standard in order to organize different GML instances.

5.5 Opacity

Opacity can be used in different ways.

5.5.1 Opacity channels (JPEG2000 Part 1)

The codestream contains an auxiliary channel with opacity information. This is the only Part 1 compatible option.

5.5.2 Opacity in a separate codestream (JPEG2000 Part 2)

Since the introduction of *compositing layers*, it is possible to store opacity information in a separate codestream. The mechanism is similar to the Part 1 case with opacity channels.

5.5.3 Opacity box (JPEG2000 Part 2)

The opacity box gives option for specifying that the last channel in a compositing layer is an opacity channel, or giving one or more chroma-key values. A chroma-key value specifies that a certain colour should be considered as transparent. A chroma-key value is useful for GMLJP2 files that have been transformed between different reference systems. Such files usually have large one-colored regions at the edges. This method of opacity, works best with the lossless mode of JPEG2000, because the compression artifacts would be visible in the regions that should be transparent.

The opacity box requires the use of *compositing layers*.

5.6 Integer values constraint of raster and gridded data

JPEG2000 allows only integer raster data. In order to store elevation data with a higher precision than one meter, it is necessary to perform an upscaling conversion of elevation values so that they all can be given as integers.

6 GML objects

(Informative)

GMLJP2 allows different kinds of data to be stored as GML in a JPEG2000 file.

6.1 Coverage

A GML *RectifiedGridCoverage* is used to georeference the image with the help of a *RectifiedGrid* (D.2.2.1.2) which handles the georeference information with the attribute *srsName* (Table D.1) with the reference systems listed in D.2.3. In addition, the coverage describes the value range of the image, which could include elevation data.

6.2 Coordinate reference system

The GML coverage refers to a coordinate reference systems (CRS) used for georeferencing the image. It could be a predefined code for the CRS, like the EPSG codes. But it is also possible to define a CRS in a *CRSDictionary.gml* box. This box could be named *CRSDictionary.gml*, or any other unique name. This profile uses a limited set of EPSG codes:

EPSG code	Reference system
urn:ogc:def:crs:EPSG::4326	GCS_WGS84
urn:ogc:def:crs:EPSG::326zz and urn:ogc:def:crs:EPSG::327zz	UTM for each zone in the northern hemisphere respectively the southern hemisphere
urn:ogc:def:ellipsoid:EPSG::7030	WGS84 ellipsoid as a vertical reference
urn:ogc:def:crs:EPSG::5773	Earth Gravity Model 1996
urn:ogc:def:crs:EPSG::3855	Earth Gravity Model 2008

Table 6.1 – EPSG codes

6.3 Axis order

There are two possible axis orders that are in use, *left hand order* and *right hand order*. The terms for these axis orders, are understood by the figure below:

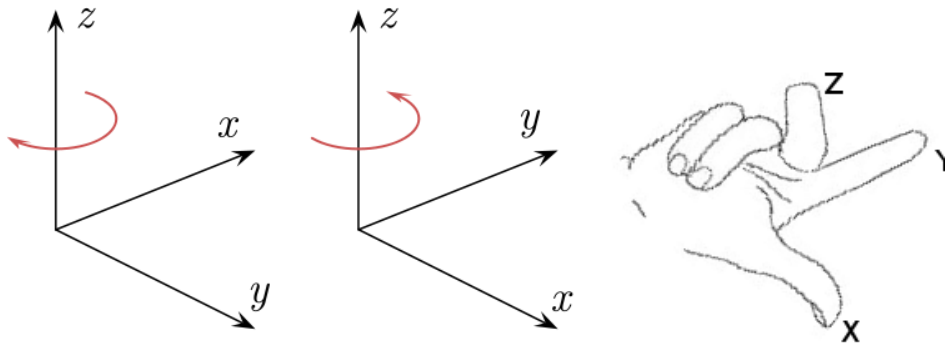


Figure 3 – Left and right hand order

The thumb denotes the first axis (typically X), the index finger denotes the second axis (Y) and the middle finger denotes the last axis (Z). If the right hand fingers are used, the first axis will be the easting axis. If the left hand fingers are used, the first axis will be the northing axis.

Mathematicians traditionally use right hand order. In the geo community, both right hand order and left hand order are used. WGS84 LL is defined as a left hand order system, while UTM is a right hand order system. Both traditions are strong. GML 3.1.1, which is used by GMLJP2, don't give a clear guideline about the axis order. This has led to a lot of confusion. GML 3.2 and later clarifies which axis order that should be used. That is to use the axis order which is defined for each reference system. OGC will provide a GMLJP2 standard based on GML 3.2.1. Therefore the DGIWG profile of GMLJP2 requires the same axis orders as the later GML versions, even though GMLJP2 uses GML 3.1.1 for now.

6.4 Annotation

Annotation includes both text and vector graphics, which can be used to point out different kinds of regions. Extended annotations can also include embedded images and video.

6.5 Feature and annotation styling

By using CSS2 styling expression grammar or SVG symbols, it is possible to describe how annotation elements should be represented visually.

6.6 Metadata

A GML element, including the coverage, can include metadata in an arbitrary schema, including ISO 19139, ISO 19139-2 and GML metadata application schema. If the GML information, like geometry or radiometric information in a GML coverage, contradicts the included metadata, the GML information will take precedence over the metadata schema. The data provider should ensure the metadata is consistent with the information provided in the GML.

7 Embedding GML in JP2 files

OGC:s GMLJP2 standard, propose that GML code for georeferencing a JPEG2000 file, should be stored in xml boxes. There is also a container entity, called *association box*, and a *label box*, which provides the means of using a hierarchical structure within a superbox. This

structure of XML boxes with GML code is described in Annex B. The box structure supports referring to different boxes in the GML structure, as well as different codestreams outside the GML structure, with the help of a *GMLJP2 URI*, described in Annex E.

7.1 JPX file format

GMLJP2 uses a number of boxes that are defined in JPEG2000 Part 2 (JPX specification). These boxes are described in Annex E.

7.2 Compatibility list in File type box

OGC GMLJP2 (05-047r3) recommends that the files are written as JP2 compatible, i.e., the file should be compatible with Part 1, and signaling this by including the string “jp2\040” in the *compatibility list* in the *File type box*. The DGIWG profile has one exception from this rule, which the paragraph about the opacity mentions (next paragraph, 7.2.1).

7.2.1 If opacity of Part 2 type is used

Opacity information according to one of the Part 2 options ([Opacity in a separate codestream \(JPEG2000 Part 2\)](#) and [Opacity box \(JPEG2000 Part 2\)](#)), requires the use of *compositing layers*. A JPEG2000 Part 1 reader won't be able to render the codestreams correctly. Therefore the compatibility list should not include the string “jp2\040”.

7.2.2 Profile

The compatibility list might also specify which [profile \(5.3.6\)](#) the first codestream should conform to. “J2P0” requires that the first codestream conforms to Profile-0, and “J2P1” requires that the first codestream conforms to Profile-1. If no profile is indicated, the codestream is allowed to follow Profile-2 (no restrictions).

7.3 Brand field in File type box

The brand field shall always be “jpx\040” for JPX files.

7.4 Reader requirement box

The presence of GML data should be signaled with a standard flag with the value 67 in a reader requirement box (C.1.3).

7.5 File suffix

The DGIWG profile uses two different file suffixes:

File suffix	Explanation
.jp2	JPEG2000 file that follows ISO/IEC 15444-1 (I.2.1 in that standard). A JPX file that is compatible with ISO/IEC 15444-1 is allowed to use this file suffix (ISO/IEC 15444-2 M.2.1). Generally, a file should use this file suffix, but they are allowed to use “.jpf” as well.
.jpf	JPEG2000 file that follows ISO/IEC 15444-2 (M.2.1 in that standard). A file that uses one of the Part 2 opacity options (7.2.1) shall use “.jpf”.

Table 7.1 – File suffixes

ISO/IEC 15444-2 M.2.1 prefer the file suffix “.jpf” for Part 2 files. But the Part 1 file suffix (“.jp2”) is already established for GMLJP2 files. Files that follow this profile can use both “.jp2” and “.jpf”, with the exception of files that uses one of the Part 2 opacity options (7.2.1). These files are required to use “.jpf”, since a Part 1 compliant reader can’t be expected to understand the opacity information. A file that uses the Part 1 opacity option (opacity channels) is allowed to use “.jp2”.

8 Annotations

(Informative)

Annotations are used for highlighting details in the JPEG2000 image. This profile uses GML for annotations.

8.1 Annotation conformance classes

There are two annotation conformance classes, A (Annotated) and XA (eXtended Annotations). A reader which only complies with conformance class B (Baseline) is not required to understand annotations.

8.1.1 Class A

Class A is the simple case of annotations. Class A compliant readers are required to understand textual annotations, simple GML elements, as points, lines, polygons and symbols.

8.1.1.1 Symbols in SVG format

The DGIWG profile uses symbols in SVG format. SVG symbols are XML instances. That means the symbols should be stored in XML boxes within the GML box structure. Since the GMLJP2 URI allows referring to XML boxes and JP2 codestreams, this constraint allows referring to SVG symbols with GMLJP2 URI. That means it is possible to embed all necessary symbols within the JPEG2000 file without extending GMLJP2.

8.1.2 Class XA

Class XA compliant readers are required to understand annotations that uses embedded images or embedded video clips. This requirement demands extensions to the profile.

8.1.2.1 Embedded images as annotations

This profile allows embedded images as annotations. The allowed file formats are JPEG (JFIF), since it is a common file format in digital cameras, PNG and GIF. The files should be embedded in ‘*uuid*’ boxes in an annotation structure which is described in C.2.1. The file type of the annotation is conveyed by a *UUID* (C.2.3 and table C.8)

8.1.2.2 Embedded video as annotations

This profile allows embedded video in MPEG-4 format as annotations. This includes H.264, which is an MPEG-4 standard. MPEG-4 organizes its data in *atoms*. An *atom* is actually identical in its structure to a *JPEG2000 box*. This makes it natural to use MPEG-4, as it can be embedded in ‘*moov*’ boxes. The ‘*moov*’ box is contained in the same annotation structure as the embedded images (C.2 and C.2.2). If MPEG-2 video is embedded, a ‘*uuid*’ box has to be used (C.2.3).

8.1.2.3 DGIWGJP2 URI

The GMLJP2 URI allows referring to two different types of information, XML blocks inside the *gml.data* structure, and JPEG2000 codestreams. The GML standard in itself, allows URI:s of the type *anyURI*. This fact allows the introduction of a new type of URI, the *DGIWGJP2 URI*. A *DGIWGJP2 URI* can refer to embedded data of type MPEG-4 and JPEG files. The *DGIWGJP2 URI* follows the convention of the *GMLJP2 URI*, and has the structure:

`dgiwgjp2://[resource type]/[label]`

where [label] is a reference to a labeled image or video clip in an annotation structure (C.2).

Examples of annotation references:

DGIWGJP2 URI	Description
<code>dgiwgjp2://annotation/photoOfScene</code>	An embedded image labeled " <i>photoOfScene</i> "
<code>dgiwgjp2://annotation/videoSeq</code>	An MPEG-4 video clip labeled " <i>videoSeq</i> "

Table 8.1 – Examples of *DGIWGJP2 URI*:s

The URI does not distinguish between image and video annotations. If the label is associated with a '*uuid*' box, the annotation is an image of the file type indicated by the *uuid*. If the label is associated with a '*moov*' box, the annotation is a video clip. The *DGIWGJP2 URI* cannot refer to JPEG2000 codestreams, since the *GMLJP2 URI* is sufficient for that purpose.

Annex A

Abstract Test Suite (normative)

A.1 Introduction

This annex provides an abstract test suite for GMLJP2 data conforming to the DGIWG profile. Some of these tests require that the tester uses software that gives a good overview of the content in a JPEG2000 file. Annex I refer to software that can be used for the abstract test suite.

A.1.1 Test classes

The test classes refer to the conformance classes that are described in Annex B.

A.2 Class B tests

Class B is described in B.3.1.

A.2.1 Georeferenced with GML according to GMLJP2

- a) Test purpose: Verify that the GMLJP2 file is georeferenced with GML according to the GMLJP2 standard.
- b) Test method: Inspect that the GML georeference exists at its proper location. The required GML structure is labeled "*gml.root-instance*".
- c) Reference: OGC 05-047r3 - GML in JPEG 2000 for Geographic Imagery (GMLJP2) Encoding
- d) Test type: Basic

A.2.2 GML code passes XML schema validation

- a) Test purpose: Verify that the GML code referencing the JPEG2000 data is valid according to the XML schema that the DGIWG profile provides. This validation is capable of catching a number of errors.
- b) Test method: Extract the GML georeference and validate the code with the xml schema provided by this profile, *DGIWGGmlJP2Profile.xsd*, which is described by Annex F.2.1.
- c) Reference: DGIWG GMLJP2 Profile, Annex F and A.5
- d) Test type: Basic.

A.2.3 GML code has RectifiedGridCoverage

- a) Test purpose: Verify that the GML code has a feature that is a *RectifiedGridCoverage*. The *RectifiedGridCoverage* is the only coverage that is allowed to be used as a georeference in *GMLJP2*.
- b) Test method: Inspect the GML reference. There should be a *RectifiedGridCoverage* at following XPath:
/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage

- c) Reference: OGC 05-047r3 - GML in JPEG 2000 for Geographic Imagery (GMLJP2) Encoding. An additional description is in this profile (DGIWG GMLJP2: D.2.1).
- d) Test type: Basic.

A.2.4 The RectifiedGrid contains all required content

- a) Test purpose: Verify that the RectifiedGrid contains all mandatory attributes and elements.
- b) Test method: Inspect the GML reference. Inspect the *RectifiedGrid* at following XPath:
/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rectifiedGridDomain/gml:RectifiedGrid

Following elements and attributes should exist:

Name	Type
limits	XML element
origin	XML element
offsetVector	XML element. Two elements are required.
axisName	XML element. Two elements are required.
srsName	XML attribute

- c) Reference: OGC 05-047r3 - GML in JPEG 2000 for Geographic Imagery (GMLJP2) Encoding and this profile (DGIWG GMLJP2: D.2.2.1.2).
- d) Test type: Basic.

A.2.5 The number of grid cells is the same as the number of pixels

- a) Test purpose: Verify that the GML grid have the same number of grid cells as the raster data it is supposed to describe. The raster is interpreted as a grid. That means the *RectifiedGrid* is a description of the geometric properties of the raster.
- b) Test method: Check the actual dimensions of the image. In the GML georeference, check the following XPath:
/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rectifiedGridDomain/gml:RectifiedGrid/gml:limits/gml:GridEnvelope
 where the number of grid cells is given as min- and max-coordinates in a coordinate system (*width, height*) with its origin at the upper left corner. This system spans from (0) to (*raster dimension size – 1*). Verify that this is the case.
- c) Reference: OGC 03-105r1 – OpenGIS Geography Markup Language (GML) Implementation Specification, v3.1.1 and OGC 05-047r3 - GML in JPEG 2000 for Geographic Imagery (GMLJP2) Encoding and *DGIWG GMLJP2 D.2.2.1.3*.

- d) Test type: Basic.

A.2.6 Coordinate lists in GML, have sufficient number of coordinates

- a) Test purpose: Verify that the GML code has sufficient number of coordinates at certain places in the GML georeference. GML uses the *XML Schema Language* type *list* for allowing more than one coordinate in one element. The *XML Schema Language* doesn't allow a schema to put any restrictions on the number of coordinates when using the *list* type. This means an *XML schema validation* will not catch that error. Therefore the number of coordinates has to be verified manually in the GML code. Typically, there should be at least two coordinates in one element, but the *gml:Point* and the *gml:offsetVector* could have three coordinates in the case of 3D data.
- b) Test method: Check that there are at least two coordinates in the GML code at following XPath's:
- a. ***/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rectifiedGridDomain/gml:RectifiedGrid/gml:limits/gml:GridEnvelope/gml:low***
 - b. ***/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rectifiedGridDomain/gml:RectifiedGrid/gml:limits/gml:GridEnvelope/gml:high***
 - c. ***/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rectifiedGridDomain/gml:RectifiedGrid/gml:origin/gml:Point/gml:pos***
 - d. ***/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rectifiedGridDomain/gml:RectifiedGrid/gml:offsetVector***
- c) Reference: OGC 05-047r3 - GML in JPEG 2000 for Geographic Imagery (GMLJP2) Encoding
- d) Test type: Basic.

A.2.7 The RectifiedGrid refers to a reference system

- a) Test purpose: Verify that the reference system is given by the attribute *srsName* in the *RectifiedGrid* element.
- b) Test method: Inspect that the attribute *srsName* is used at appropriate place in the GML, i.e. the *RectifiedGrid*. There are a few places where this attribute can be used. This profile require that this attribute is used at this location:
- /gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rectifiedGridDomain/gml:RectifiedGrid@srsName***
- c) Reference: OGC 05-047r3 - GML in JPEG 2000 for Geographic Imagery (GMLJP2) Encoding and *DGIWG GMLJP2 – D.2.3*
- d) Test type: Basic

A.2.8 Left-hand order coordinate system is used for WGS84 LL coordinates

- a) Test purpose: This profile prescribes that the axis order for WGS84 LL should follow standard practice, i.e., use *left-hand order* axis (latitude, longitude). This test is not applicable for UTM/WGS84, which uses *right-hand order* axis.
- b) Test method: It is not advisable to test this by inspecting the image in GIS software, because that might instead highlight the axis handling in the software. One important part of the problem is that both types of axis orders are common in the software. This profile tries to achieve unification in the handling of axis orders, which is in line with the recommendation of GML 3.2.1. Therefore inspect the coordinates directly in the GML code at following XPaths:
 - a. **`/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rectifiedGridDomain/gml:RectifiedGrid/gml:origin/gml:Point/gml:pos`**
While using WGS84-LL, the axis order shall be left-handed.
 - b. **`/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rectifiedGridDomain/gml:RectifiedGrid/gml:offsetVector`**
While using WGS84-LL, the axis order shall be left-handed.
- c) Reference: DGIWG GMLJP2 Profile – D.2.2.1.3 and 6.3 for an illustration
- d) Test type: Basic

A.2.9 Image coordinate system gives the coordinates in the correct order

- a) Test purpose: The image coordinates does not follow the same rule as the ground coordinates. The image coordinates shall always be given in following order (width, height).
- b) Test method: Inspect the coordinates directly in the GML code at following XPath:
`/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rectifiedGridDomain/gml:RectifiedGrid/gml:limits/gml:GridEnvelope/gml:high`
The axis order shall be given as width and height in that order. This can only be inspected in a test case there the image **is not square**.
- c) Reference: DGIWG GMLJP2 Profile – D.2.2.1.3
- d) Test type: Basic

A.2.10 Metadata extent is coherent with the GML extent

- a) Test purpose: Verify that the extent information in the metadata is coherent with the *RectifiedGridCoverage* that gives the extent of the GMLJP2 file.

- b) Test method: Inspect the GML reference and the metadata extent information. The extent should be the same area. The *RectifiedGridCoverage* is at following Xpath in the GML georeference:
/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage
- The metadata extent is at following Xpath:
/gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:extent
- if ISO/IEC 19139 is used.
- c) Reference: DGIWG GMLJP2 Profile – 6.6.1
- d) Test type: Basic.

A.2.11 Security classification in metadata

- a) Test purpose: Verify that the security classification is recorded in the metadata.
- b) Test method: Inspect the metadata to see if there is a security classification.
- a. The metadata is embedded at the following XPath:
/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:metadataProperty
 - b. In the metadata, the security classification is found at following XPath:
/gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:resourceConstraints/gmd:MD_SecurityConstraints/gmd:classification
- c) Reference: ISO 19115
- d) Test type: Basic.

A.2.12 Security classification in Intellectual property rights box

- a) Test purpose: If the file is classified in a security class more restricted than *unclassified*, the classification shall be recorded in an *Intellectual property rights* box in order to secure that information in case the metadata is lost in some way.
- b) Test method: Inspect the 'jp2i'-box (the *Intellectual property rights* box). The box shall contain an IPR element. The security classification shall be stored at following XPath:
/jp:IPR/jp:IPR_EXPLOITATION/jp:IPR_USE_RESTRICTION
- c) Reference: DGIWG GMLJP2 Profile – C.1.5
- d) Test type: Basic.

A.2.13 References to embedded codestreams and XML instances are valid

- a) Test purpose: The GML code uses *GMLJP2 URI* addresses to refer to different embedded parts of the file. These parts can be either codestreams or XML blocks within the *gml.data* structure. Codestreams are addressed by their order in the file beginning with number zero. XML blocks are addressed by their accompanying *label* boxes.

- b) Test method: Inspect that existing *GMLJP2 URI*:s refers to existing parts of the file.
 - a. One *GMLJP2 URI* in particular, is essential:
/gml:FeatureCollection/gml:featureMember/gml:FeatureCollection/gml:featureMember/gml:RectifiedGridCoverage/gml:rangeSet/gml:File/gml:fileName
This element refers to the codestream that the GML code is georeferencing. This element shall always exist and refer to an existing codestream.
 - b. There can be other *GMLJP2 URI*:s for referring to XML instances, like SVG symbols. These will occur in the annotations. This part of the test is only required for *class A* and *class XA* files.
- c) Reference: OGC 05-047r3 - GML in JPEG 2000 for Geographic Imagery (GMLJP2) Encoding
- d) Test type: Basic.

A.2.14 The Reader Requirement box signals the use of GML

- a) Test purpose: The use of GML shall be signaled by the *reader requirement* box.
- b) Test method: Inspect the file with some software that understands the *reader requirement* box. The flag with number 67 must be among the standard flags.
- c) Reference: ISO/IEC 15444-2 and DGIWG GMLJP2 Profile: C.1.3.
- d) Test type: Basic.

A.2.15 Precinct partition size

- a) Test purpose: The precinct partition size is restricted to be a power of two.
- b) Test method: Inspect the codestreams markers COD and COC. The COC information overrides the COD information. COD and COC for a tile, overrides the COD and COC for the whole codestream.
- c) Reference: ISO/IEC 15444-1 B.5, A.6.1, A.6.2 and DGIWG GMLJP2 Profile: 5.3.2.6.
- d) Test type: Basic.

A.2.16 Brand field

- a) Test purpose: GMLJP2 files uses extensions from JPEG2000 Part 2, must use a file suffix which reflect the file type that is used.
- b) Test method: Inspect the brand field in the file type box. The brand field must use the value 'jpx\040'.
- c) Reference: ISO/IEC 15444-2 M.2.1, M.8 and DGIWG GMLJP2 Profile: 7.3 and C.1.2.
- d) Test type: Basic.

A.2.17 Compatibility list

- a) Test purpose: The compatibility list contains a list about which standards and profiles the file conforms to.
- b) Test method: Inspect the compatibility list in the file type box. If none of the Part 2 options for opacity is used, the compatibility list shall contain 'jp2\040'. All files shall contain 'jpx\040'.
- c) Reference: ISO/IEC 15444-2 M.8 and DGIWG GMLJP2 Profile: 7.2 and C.1.2.
- d) Test type: Basic.

A.2.18 File suffix

- a) Test purpose: A GMLJP2 file must use a file suffix which reflects the file type that is used.
- b) Test method: Inspect the compatibility list in the file type box and the filename. If the compatibility list contains 'jp2\040', the file suffixes ".jp2" and ".jpf" are allowed. Otherwise only ".jpf" is allowed.
- c) Reference: ISO/IEC 15444-2 M.2.1 and DGIWG GMLJP2 Profile: 7.5 and C.1.2.
- d) Test type: Basic.

A.3 Class A tests

Class A is described in B.3.2.

A.3.1 Allowed style elements

- a) Test purpose: *Class A* files contains annotations which can be styled. It has to be relevant styling elements for each of the annotations. A *Label* annotation can be styled with a *LabelStyle* element, but not with a *GeometryStyle* element.
- b) Test method: Inspect the GML code that contains the annotations. Inspect the *defaultStyle* elements in annotations. Relative to the *defaultStyle* element, there can be two types of style elements, *geometryStyle* and *labelStyle*:

a. **gml:defaultStyle/gml:Style/gml:featureStyle/gml:FeatureStyle/gml:geometryStyle**

b. **gml:defaultStyle/gml:Style/gml:featureStyle/gml:FeatureStyle/gml:labelStyle**

The *labelStyle* element is used only when *defaultStyle* is a child element of the annotation content type *Label*. All other annotation types shall use *geometryStyle*. If they have labels, the labels are in their annotation content as a *Label* element, which can have a *LabelStyle* element.

- c) Reference: DGIWG GMLJP2 Profile – D.3.4.1
- d) Test type: Class A

A.3.2 References to embedded symbols are valid

- a) Test purpose: *Class A* files can contain annotations which uses symbols in SVG format embedded in the JPEG2000 file in *xml* boxes.

- b) Test method: Inspect all elements of type *gml:symbol*. The attribute *xlink:href* shall use a valid *GMLJP2 URI* for referring to the SVG symbol, which should be embedded in the *gml.data* structure in the file.
- c) Reference: DGIWG GMLJP2 Profile – D.1.3.2 and D.3.4.1.3
- d) Test type: Class A

A.4 Class XA tests

Class XA is described in B.3.3.

A.4.1 References to embedded images and video streams are valid

- a) Test purpose: *Class XA* files contains *DGIWGJP2 URI* references to embedded images and video. These parts can be either '*uuid*'-boxes or '*moov*'-boxes. They shall in both cases be in an *Annotation* structure.
- b) Test method: Inspect that the *DGIWGJP2 URI* addresses refers to actual boxes in the *Annotation* structure.
- c) Reference: DGIWG GMLJP2 Profile – C.2 and D.1.4
- d) Test type: Class XA

Annex B

Conformance

(normative)

This annex describes the conformance requirements for DGIWG GMLJP2 files as well as software that read or process a DGIWG GMLJP2 file.

B.1 Purpose

The purpose of this annex is to define different sets of requirements for a reader or a processor of GMLJP2 files as well as files.

B.2 JPEG2000 Profiles

The JPEG2000 profiles are different sets of requirements that apply to the JPEG2000 files. There are five different profiles according to ISO/IEC 15444-1 to this date. Profile-3 and Profile-4 are restrictions on JPEG2000 streams with video. These profiles are outside the scope of this work. Profile-2 means *no restrictions*, i.e. all elements defined by ISO/IEC 15444-1 can be used. If there is no indication of profile, the file conforms to Profile-2. Profile-2 is not listed in the table of the restrictions:

Restrictions	Profile-0	Profile-1
Tile size	Either smaller or equal to 128x128, or one tile for the whole image	Either smaller or equal to 1024x1024, or one tile for the whole image
Code block size	32x32 or 64x64	Smaller or equal to 64x64. Non-square codeblocks are allowed
LL_D (lowest resolution element)	Not larger than 128x128	Not larger than 128x128

Table B.1 – JPEG2000 Profiles

DGIWG GMLJP2 allows all three profiles that are within the scope of this work.

B.3 Conformance classes

The conformance classes are different sets of requirements on a reader or a processor depending on the complexity of the file. The file belongs to a certain conformance class, and can thus be said to have certain requirements on the reader or processor.

B.3.1 Class B – Baseline

A Class B file is a JPEG2000 file georeferenced with GML according to this profile.

B.3.2 Class A – GMLJP2 with Annotations

A Class A file extends Class B with the addition of annotations. Class A uses the inherent GMLJP2 support for annotations. The annotations can be:

B.3.2.1 Textual annotations

Textual annotations give textual descriptions of features in the GMLJP2 image.

B.3.2.2 Points

Points are graphical elements that highlight features in the GMLJP2 image.

B.3.2.3 Lines

Lines are graphical elements that highlight linear features in the GMLJP2 image.

B.3.2.4 Polygons

Polygons are graphical elements that highlight areal features in the GMLJP2 image.

B.3.2.5 Symbols

Symbols can be used as a styling element for annotations, mainly points. The symbols should be in SVG format. That means that the symbols can be embedded in the GMLJP2 image within the *gml.data* box, and referred to by a GMLJP2 URI, which can refer to any xml box within the *gml.data* box.

B.3.3 Class XA – GMLJP2 with extended annotations

A Class XA file extends Class A (and thus also Class B) contains additional types of annotations. The Class XA annotations use elements that are not defined in the GMLJP2 standard. The additional annotation types are:

B.3.3.1 Image annotations

Image annotations can be used in GMLJP2, but the DGIWG profile adds support for annotations with images embedded in the JP2 file. That way the image annotations will follow the file. Images should be embedded as JP2 boxes of type *uuid* in an annotation box structure (C.2). They are referred to by a *DGIWGJP2 URI*, which is an addition by the DGIWG profile.

B.3.3.2 Video annotations

Video annotations are videos embedded in the JP2 file. Video clips should be embedded as JP2 boxes of the type *moov* in an annotation box structure (C.2). They are referred to by a *DGIWGJP2 URI*. Videos must conform to the MPEG-4 standard.

B.4 Implementation Conformance Report

This chapter provides a report form for testing software conformance to the DGIWG GMLJP2 Profile.

- R = Required
- C = Conditional
- O = Optional
- S = Supported
- P = Partial support
- N = No support

Conformance test	Produce				Use				Comment	Conf. class
	R/C/O	S	P	N	R/C/O	S	P	N		
Feature	R/C/O	S	P	N	R/C/O	S	P	N		
JPEG2000 boxes										
JPEG2000 Signature box	R									B
File type box	R									B
BR field always 'jpx\040'	R									B
MinV field always 0	R									B
CL field	R									B
'jp2\040' or 'jpx\040' in the CL field	R									B
Reader requirements box	R									B

Conformance test	Produce				Use				Comment	Conf. class
	R/C/O	S	P	N	R/C/O	S	P	N		
Feature										
One of the standard flags (SF ¹) must have value 67 , which signals the presence of GML data.	R									B
JPEG2000 header box	R				R					B
Image header box	R				R					B
Height and Width	R				³					B
Number of components	R									B
Bits per component	R									B
Compression type always 7	R									B
Unknown colourspace: 0 or 1	R				R					B
Intellectual property rights This post signals that an Intellectual Property Rights box is present or absent: 0 : No Intellectual Property Rights box. 1 : An Intellectual Property Rights box exists.	R				R					B
Colour specification box	R				R					B
METH – 1 or 2	R				R					B
PREC - 0	R									B
APPROX - 0	R									B
EnumCS – 16, 17 or 18	C				R					B
PROFILE	C				R					B
Intellectual property rights	C ⁴				R					B
IPR_USE_RESTRICTION One of following keywords must be used: unclassified restricted confidential secret topSecret	C ⁵				R					B
IPR_MGMT_TYPE	O				R					B
Codestream box	R				R					B
Codestream-related properties										
Multiple codestreams	O				R					B
Compositing layers	O				R					B
Opacity channel	O				R					B
Chroma-key opacity	O				R					B
GML content										
WGS84 LL (EPSG::4326)	O ⁶				R					B

³ Height, width, number of components and number of bits per component, are information available from the codestream, so they are not a reader requirement in the context of an *Image header box*.

⁴ Required when the Image Header box signals the presence of an Intellectual Property Rights box. Additionally, the IPR box is required by this profile if the image data is classified as any of the options *restricted*, *confidential*, *secret* and *topSecret*.

⁵ Required when classified as *restricted*, *confidential*, *secret* or *topSecret*.

Conformance test	Produce				Use				Comment	Conf. class
	R/C/O	S	P	N	R/C/O	S	P	N		
Feature	R/C/O	S	P	N	R/C/O	S	P	N		
UTM/WGS84 (EPSG::32hzz)	O				R					B
WGS84 vertical reference (EPSG::7030)	O				R					B
Earth Gravity Model 1996 (EPSG::5773)	O				R					B
Earth Gravity Model 2008 (EPSG::3855)	O				R					B
One and only one GML RectifiedGridCoverage for each codestream	R				R					B
The RectifiedGrid uses the attribute <i>srsName</i> to refer to the reference system by its EPSG code	R				R					B
The origin of the RectifiedGrid must have at least two coordinates in the coordinates list	R				R					B
The coordinate order of the origin must follow the coordinate order required by D.2.2.1.4	R				R					B
The image coordinates for the <i>limits</i> element must be valid. The minimum coordinates must be <code><gml:low>0 0</gml:low></code> The maximum coordinates must not exceed the image size expressed in number of pixels in each dimension, i.e., following must apply: <code><gml:high>(width - 1) (height - 1)</gml:high></code>	R				R					B
The RectifiedGrid must have two instances of <i>axisName</i> .	R				R					B
The RectifiedGrid must have two instances of <i>offsetVector</i>	R				R					B
The instances of <i>offsetVector</i> must have two coordinates each	R				R					B
The coordinates in the <i>offsetVectors</i> must follow the coordinate order required by D.2.2.1.4	R				R					B

⁶ At least one of the horizontal reference systems WGS84 LL or UTM/WGS84 must be used.

Conformance test	Produce				Use				Comment	Conf. class
	R/C/O	S	P	N	R/C/O	S	P	N		
Feature	R/C/O	S	P	N	R/C/O	S	P	N		
The metadata extent must comply with the extent given by the RectifiedGrid	R				R					B
Annotations										
Annotation	O									A
Display annotation					R					A
Display annotation labels					R					A
Display annotation pointers					R					A
PointOfInterest	O									A
Display PointOfInterest					R					A
CurveOfInterest	O									A
Display CurveOfInterest					R					A
RegionOfInterest	O									A
Display RegionOfInterest					R					A
SVG symbols for points	O				R					A
Annotation colour	O				R					A
Font family	O				R ⁷					A
Font size	O				R					A
Font style	O				R					A
Line thickness	O				R					A
Fill pattern	O				R					A
Embedded image annotations If a <i>DGIWGJP2</i> URI refers to a <i>uuid</i> box, the referred box should be interpreted as an embedded JPEG file.	O									XA
Display embedded image annotations					R					XA
Embedded video annotations If a <i>DGIWGJP2</i> URI refers to a <i>moov</i> box, the referred box should be interpreted as an embedded MPEG-4 video.	O									XA
Display embedded video annotations Embedded video annotations should be displayed on request from the user.					R					XA
Other requirements										
Metadata	R				R					B

Table B.2 – Implementation Conformance Report

⁷ The test system is required to have the chosen font installed in order to display correct font family. If the test indicates no success, it is possible that the font family wasn't installed on the test machine.

Annex C

JP2 boxes

(normative)

C.1 Required boxes

C.1.1 JPEG 2000 Signature box

Box type is 'JP\040\040'. The box size is always 12 bytes including the box length and type. The content is always '<CR><LF><0x87><LF>'. This box is always first in a JPEG2000 file. This box could be used as a *magic number* for identifying a JPEG2000 file without a correct file suffix. The inclusion of <CR> and <LF>, serves the purpose of catching file transfer errors. It is common that these two characters are translated while transferring a file between a Windows and a UNIX system if one forgets to use *binary transfer*.

C.1.2 File type box

Box type is 'ftyp'. The box has following fields:

BR: 4-byte string for the *brand field*. Shall always be 'jpx\040' in this DGIWG profile, since ISO/IEC 15444-2 (JPEG2000 with extensions) is needed for the GMLJP2 standard.

MinV: Minor version of the ISO/IEC 15444-1 specification. The value is an integer that shall always be 0.

CL: Compatibility list. This is a list of four-byte strings denoting different standards and profiles that the file conforms to. It shall contain at least the string 'jpx\040' (ISO/IEC 15444-2). In addition the compatibility list shall contain the string 'jp2\040' (ISO/IEC 15444-1), with the exception of files that uses one of the Part 2 variants of opacity.

Field name	Size and type	Value
BR	4 byte string	'jpx\040'
MinV	4 byte unsigned integer	0
CL	List of 4 byte strings	'jpx\040' and possibly 'jp2\040' if the file uses one of the Part 2 options for opacity.

Table C.1 – File type box

C.1.3 Reader requirement box

Box type is 'rreq'. The *reader requirement* box specifies both which features the JPEG2000 file has used and which features that must be supported by a reader to fully use the file. This box must follow immediately after the *File type* box. This profile requires that the presence of GML should be signaled by this box.

ML: Mask length. The total size of the compatibility masks.

FUAM: Fully Understand Aspects Mask. Describes requirements for fully understand the image.

DCM: Decode Completely Mask. Describes requirements for displaying the image correctly.

NSF: Number of Standard Flags.

SFⁱ: Standard Flag. Standard features the file uses. This profile requires that one of them shall signal the presence of GML data by having one standard flag with the value **67**.

SMⁱ: Standard Mask. Compatibility mask with respect to the features specified by **SFⁱ**.

NVF: Number of Vendor Features.

VFⁱ: Vendor Feature.

VMⁱ: Vendor Mask.

Field name	Size and type	Value
ML	1 byte unsigned integer	
FUAM	signed integer - size specified by ML	
DCM	signed integer - size specified by ML	
NSF	2 byte unsigned integer	
SF ⁱ	2 byte unsigned integer for each flag	One of the flags must have the value 67
SM ⁱ	signed integer – size specified by ML	
NVF	2 byte unsigned integer	
VF ⁱ	8 byte (128-bit UUID) per vendor feature	
VM ⁱ	signed integer - size specified by ML	

Table C.2 – Reader requirement box

Note that the DGIWG GMLJP2 Profile doesn't have any requirements for the mask fields (FUAM, DCM, SM and VM (the vendor mask is outside the scope anyway)). There is a description about how to calculate the masks in ISO 15444-2 M.6. Since this description is quite complicated and there are hardly no good examples on how to use these masks, this profile don't require the use of the masks. However, a producer that wishes to populate these masks is recommended to follow ISO 15444-2 M.6. In that case, a suggestion is that the FUAM requires that the GML georeference is understood, and that the DCM does not require that the GML georeference is understood.

C.1.4 JPEG 2000 header box

Box type is 'jp2h'. This is a superbox, that is, it contains other boxes. The box shall at least contain:

C.1.4.1 Image header box

Box type is 'ihdr'. This box shall always be the first in a JPEG 2000 header box. The box has following fields:

Height: 4-byte unsigned integer for the image height.

Width: 4-byte unsigned integer for the image width.

NC: 2-byte unsigned integer for the number of components.

BPC: 1-byte field for the number of bits per component minus 1. The value is in the 7 lowest bits. The highest bit gives if the components contain signed or unsigned values:

0: unsigned values

1: signed values

C: 1-byte unsigned integer for the compression type. Shall always be 7.

UnkC: 1-byte unsigned integer which says if the colourspace is unknown. Allowed values:
0: known colourspace specified in the *Colourspace Specification* boxes

1: unknown colourspace (in that case, the provided *Colourspace Specification* box shall be treated as it does accurately reproduce the image even though that is not known. There is always at least one *Colourspace Specification* box present no matter if the colourspace is known or not, since it is a requirement in the JPEG2000 standard.)

IPR: 1-byte field indicating whether an IPR box with information about Intellectual Property Rights exists.

0: no information about intellectual property right information, thus no IPR box

1: there is information about intellectual property rights, so there is an IPR box

Field name	Size and type	Value
Height	4 byte unsigned integer	
Width	4 byte unsigned integer	
NC	2 byte unsigned integer	
BPC	1 byte	Number of bits per component minus 1
C	1 byte unsigned integer	7
UnkC	1 byte unsigned integer	0: known colourspace 1: unknown colourspace
IPR	1 byte	0: no information about intellectual property rights 1: there are information about intellectual property rights

Table C.3 – Image header box

C.1.4.2 Colour specification box

Box type is 'colr'. A colour specification box defines a method to interpret the colourspace of the decompressed image data. It is possible to have multiple colour specification boxes, but at least one is required. The box has following fields:

METH: 1-byte unsigned integer specifying the method used. Allowed values:

1: enumerated colourspace

2: restricted ICC profile

PREC: 1-byte signed integer. Precedence. Reserved for ISO use. Value shall be set to zero.

APPROX: 1-byte unsigned integer for colourspace approximation. Value shall be set to zero.

EnumCS: 4-byte unsigned integer for enumerated colourspaces. EnumCS shall only exist if METH == 1. Allowed values:

16: sRGB

17: greyscale

18: sYCC

PROFILE: field with variable size containing an ICC profile. This field shall only exist if METH == 2.

Field name	Size and type	Value
METH	1 byte unsigned integer	1: enumerated colourspace 2: restricted ICC profile
PREC	1 byte signed integer	0
APPROX	1 byte unsigned integer	0
EnumCS	4 byte unsigned integer	16: sRGB 17: greyscale 18: sYCC
PROFILE	Variable size	

Table C.4 – Colour specification box

C.1.4.3 Channel definition box

Box type is 'cdef'. The channel definition box specifies the meaning of each sample in each channel in the image. The box has following fields:

N: 2-byte unsigned integer. Number of channel descriptions in the image.

Cnⁱ: 2-byte unsigned integer. Channel index for channel number i.

Typⁱ: 2-byte unsigned integer. Channel type for channel number i. Allowed values:

0: Colour image data for the associated colour.

1: Opacity. A sample value of 0 indicates that the sample is completely transparent, i.e. 0% opaque. A maximum sample value, i.e. as high as the bit depth allows, indicates 100% opaque.

Asocⁱ: 2-byte unsigned integer. Channel association. Specifies the index of the colour for which the channel is associated. Allowed values:

0: This channel is associated as the image as a whole (for example, an opacity channel that should be applied to all colour channels).

1 to $(2^{16}-2)$: This channel is associated with the colour indicated by this value.

Field name	Size and type	Value
N	2 byte unsigned integer	
Cn ⁱ	2 byte unsigned integer	
Typ ⁱ	2 byte unsigned integer	0: Colour image data for the associated colour. 1: Opacity. Sample value of 0 indicates completely transparent (0% opaque). Sample with maximum value (related to the bit depth) indicates 100% opaque.
Asoc ⁱ	2 byte unsigned integer	0: This channel is associated as the image as a whole. 1 to $(2^{16}-2)$: This channel is associated with the colour indicated by this value.

Table C.5 – Channel definition box

C.1.5 Intellectual property rights box

Box type is 'jp2i'. The original purpose of this box is to store information about intellectual property rights. In addition, the DGIWG profile requires that security constraints should be stored in this box. This is a way to ensure that security constraints won't get lost in case the metadata gets lost. The only exception is if the *GMLJP2* file is *unclassified*. ISO/IEC 15444-2 provides a schema definition for metadata in XML format (ISO/IEC 15444-2 N.4.4), from which it is recommended to use the IPR element (ISO/IEC 15444-2 N.5.4 and N.6.4). This profile restricts the content further to require security classification in the IPR content.

Field name	Description
IPR_USE_RESTRICTION XPath: /jp:IPR/jp:IPR_EXPLOITATION/jp:IPR_USE_RESTRICTION	Required unless <i>unclassified</i> . IPR_USE_RESTRICTION shall contain the classification code: <i>unclassified</i> <i>restricted</i> <i>confidential</i> <i>secret</i> <i>topSecret</i> If different items in the file have different security levels, the highest security level should be used in this field.
IPR_MGMT_TYPE XPath: /jp:IPR/jp:IPR_EXPLOITATION/jp:IPR_MGMT_SYS/jp:IPR_MGMT_TYPE	Optional. IPR_MGMT_TYPE in this profile can be used to indicate the country responsible for the security classification.
IPR_PROTECTION Xpath: /jp:IPR/jp:IPR_EXPLOITATION/jp:IPR_PROTECTION	Optional. IPR_PROTECTION in this profile can be used as a releasability list. Countries that the file is releasable to are listed separated by semicolon.

Table C.6 – IPR fields

C.1.5.1 IPR_USE_RESTRICTION

The security classification shall be stored in such way that it becomes clear while reading the XML content without a parser. That means that the security classification should be signaled by the use of one of following words in the IPR_USE_RESTRICTION element:

Keyword
unclassified
restricted
confidential
secret
topSecret

Table C.7 – Security classification keywords

The keywords are used in ISO 19139. If the dataset is *unclassified*, this element is optional. Any other classification code makes this element mandatory.

C.1.5.2 IPR_MGMT_TYPE

This profile uses *IPR_MGMT_TYPE* to indicate the country responsible for the security classification.

C.1.5.3 IPR_PROTECTION

This profile uses *IPR_PROTECTION* for a releasability list. The list contains country names that the file is releasable to. The country names are separated by semicolon.

C.1.5.4 Country codes

IPR_MGMT_TYPE and *IPR_PROTECTION* shall use trigram codes from *ISO 3166-1 alpha-3* for the countries. For coalitions and other types of regions (like NATO or Asia), use trigrams from STANAG 1059 when available.

C.1.5.5 Example

Example of the content in an IPR box:

```
<?xml version="1.0"?>
<jp:IPR
  xmlns:jp=http://www.jpeg.org/jpx/1.0/xml
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <jp:IPR_EXPLOITATION>
    <jp:IPR_USE_RESTRICTION>unclassified</jp:IPR_USE_RESTRICTION>
    <jp:IPR_MGMT_SYS>
      <jp:IPR_MGMT_TYPE>SWE</jp:IPR_MGMT_TYPE>
    </jp:IPR_MGMT_SYS>
    <jp:IPR_PROTECTION>SWE;FRA;USA;GBR;ARE;ZAF;DEU;ITA;CZE
    </jp:IPR_PROTECTION>
  </jp:IPR_EXPLOITATION>
</jp:IPR>
```

C.1.6 Codestream box

Box type is 'jp2c'. This box contains the actual raster data, i.e. the codestream. It is common to set the *LBox* field to zero, which means the codestream box contains the rest of the file content. But if we have multiple codestreams, we have to give the true box length for at least every codestream except the last.

C.2 Annotation boxes

Annotations in the form of images and video clips, requires special boxes. The embedded images and video clips are stored in a structure similar to the *GML box* structure which is described in *D.1.1*. The reason for storing the attachments in such structure, is to be able to address different embedded images from the GML georeference with a URI. The mechanism of the URI is described in *D.1.4*. It works like the GMLJP2 URI, which is described in *D.1.3*.

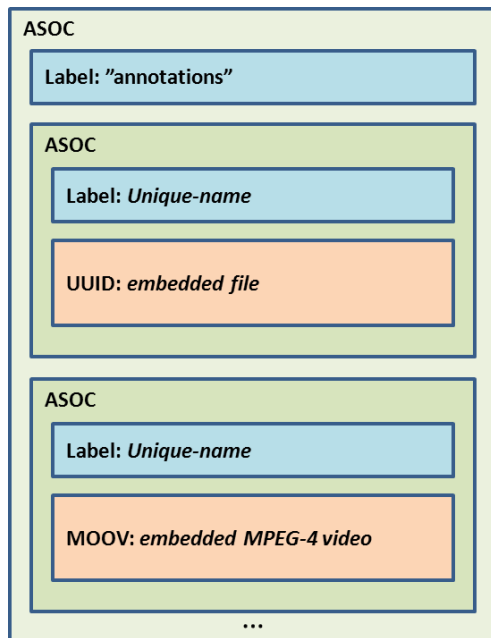


Figure C.1 – Annotation box structure

The *asoc* boxes associates embedded images and video with labels that are used for referring to the annotations. The outer *asoc* box is labeled “*annotations*”. All boxes must have unique names as their labels because these names are used in URI:s referencing the embedded images and video clips.

C.2.1 Image annotation boxes

Image annotation boxes use a *uuid* box for containing the image. The image format can be *JPEG*, *PNG* or *GIF*. *C.2.3* describes how the embedded image should be stored in the *uuid* box.

C.2.2 Video clip box

Box type is ‘*moov*’. This box was first defined as the main *atom* in Apple’s QuickTime format. It has since also been used as the main *atom* in the MPEG-4 format. The *atom* structure of QuickTime and MPEG-4 is identical with JPEG2000’s box structure. It is therefore logical to use that box for video clips embedded in JPEG2000 files. This profile uses MPEG-4.

C.2.2.1 Boxes inside the ‘*moov*’ box

The ‘*moov*’ box is a super box containing necessary sub boxes used for MPEG-4 video clips. This profile does not attempt to define any requirements for these boxes, besides that they should be kept as they are and adhere to the MPEG-4 standard.

C.2.3 UUID boxes

The *uuid* box uses a UUID for identifying the type of data that it contains.

The *uuid* box is reserved for vendor specific information. By storing embedded files in this box, this profile avoids conflict with other JPEG2000 implementations. The content of the *uuid* box shall be a *universal unique identifier (UUID)* + the embedded file:

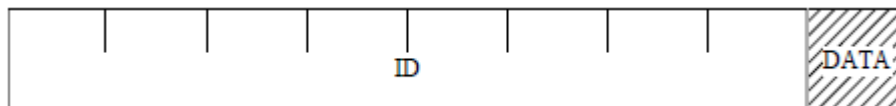


Figure C.2 – UUID box

Field name	Size and type	Value
ID	16 bytes UUID as specified by ISO/IEC 11578	Value shall indicate which file type that is used for the embedded image. Values and file types used by this profile, are listed in table C.8
DATA	Variable size	An embedded file.

Table C.8 – UUID fields

The *ID* field is a unique identifier which in this profile is used to indicate the file type of the embedded file:

File type	UUID
JPEG (JFIF)	0x652954C11E674928BF4D086D1B7EEA48
PNG	0x53F99F6240824D848AFC071230D82786
GIF	0x84F8203E2EE148209DD4D712C4A92EFC
MPEG-2	0xCA995ABABDE94CD88307E99264338CEE

Table C.9 – File types

The *UUIDs* listed in table C.9, shall not be changed for their file types. Future versions of this profile might have additional file types and *UUIDs*.

Annex D

GML encoding (normative)

D.1 GML objects

GML is used for different geographic properties in the image. OGC GMLJP2 standard (05-047r3) specifies that GML 3.1.1 shall be used in GMLJP2. This profile uses georeferencing and annotation.

D.1.1 GML box structure

The GML code is embedded in JP2 boxes:



Figure D.1 - GML box structure

The *asoc* boxes associates the labels with their GML instances, which are contained in *xml* boxes. The outer *asoc* box is labeled "*gml.data*". The inner *asoc* box labeled "*gml.root-instance*" is required. All boxes must have unique names as their labels because these names are used in URI:s referencing GML instances in the file.

D.1.2 *gml.root-instance*

This box shall contain a coverage description. It may also contain:

- Metadata instances
- Annotation instances

It may not contain anything else.

This instance can contain coverages and annotations for multiple codestreams.

D.1.3 GMLJP2 URI

The *GMLJP2 URI* is a Universal Resource Identifier for locating objects within a JPEG2000 file. In a *rangeSet* element, it is used for locating the range of values that the *RectifiedGridCoverage* consists of. The range of values is the codestream.

A GMLJP2 URI is structured as:

```
gmljp2://[resource type]/[resource id][#fragment-identifier]
```

where *resource type* is *xml* or *codestream*.

D.1.3.1 Codestream resource type

This type of URI references a particular *codestream* within the JPEG2000 file. The URI is written as:

```
gmljp2://codestream/[codestream number]
```

where [codestream number] is an integer greater than or equal to 0, identifying a particular codestream within the file.

If one intends to have multiple codestream files, this is the only way to know which codestream a specific GML instance applies to.

D.1.3.2 XML resource type

This type of URI references a particular *xml* box in the file. The URI has the following form:

```
gmljp2://xml/[label][#id]
```

where [label] identifies a particular *xml* box with its label. That means that the labeling of the *xml* boxes serves the purpose of addressing specific boxes in the file. In order for this to work, it is mandatory to use unique names in the labels. The last part, *[#id]*, which is optional, identifies a particular XML fragment in the xml instance by its *id* attribute. This mechanism can reference any type of *xml* box inside the GML box structure. This profile uses same mechanism to reference parts of an SVG document containing symbols for the annotations. That makes it possible to store SVG symbols embedded in the file as a library inside one single XML instance.

D.1.4 DGIWGJP2 URI

The *DGIWGJP2 URI* is introduced by this profile, and outside standard GMLJP2. The purpose is to refer to embedded images and video used for annotations. The *DGIWGJP2 URI* is constructed in the same way as a *GMLJP2 URI*:

```
dgiwgjp2://[resource type]/[label]
```

where [resource type] is "*annotation*" and [label] identifies a particular annotation within an annotation structure, similar to the *gml.data* structure (C.2).

D.1.4.1 Annotation resource type

The annotation resource type points to any type of resource which is embedded in the annotation structure. It could be an embedded image (C.2.1) or a video clip (C.2.2).

D.2 GML Georeferencing

For georeferencing an image, GMLJP2 uses a *GML coverage*. One specific codestream can and must have only one coverage. The coverage is in the *gml.root-instance* box. The GML is defined by an XML schema for the DGIWG GMLJP2 profile. That schema is more restricted

than the XML schema for OGC:s GMLJP2, even though it has one additional element (Video) for video annotations.

D.2.1 GML structure

The GML is structured as:

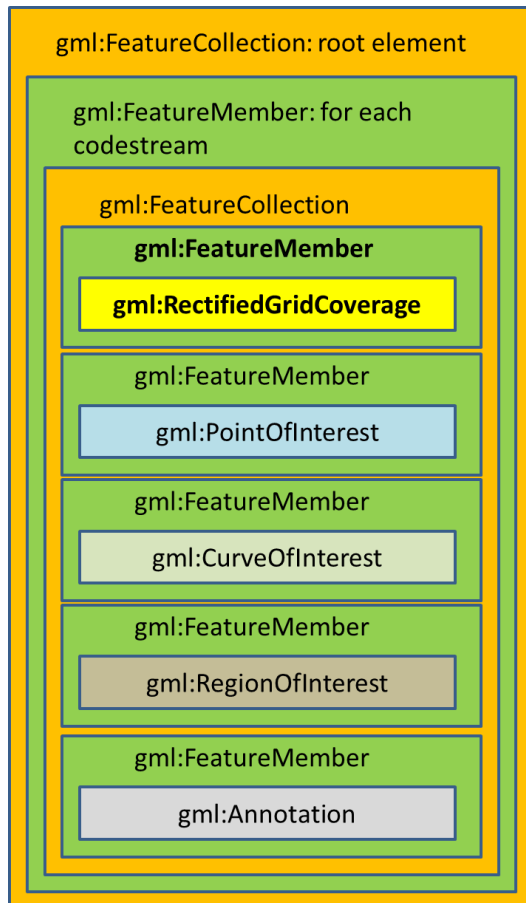


Figure D.2 – GML structure

The GML georeference uses as a root element, a *FeatureCollection* for the whole GMLJP2 file. That *FeatureCollection* has one member for each codestream. All these members are *FeatureCollections* by them self. The codestream specific *FeatureCollections*, must have at least one member which is a *RectifiedGridCoverage*. In addition, they can also contain different kinds of annotation elements, such as *PointOfInterest*, *CurveOfInterest*, *RegionOfInterest* and *Annotation*.

D.2.2 RectifiedGridCoverage

The required *gml.root-instance* box shall contain a *FeatureCollection* with a *RectifiedGridCoverage*. This element gives the extent of the image file. The rectified grid has an origin in a given coordinate reference system. The *origin's Point* element, references the CRS by either an EPSG code or a GMLJP2 URI to a CRS described in the JPEG2000 file.

D.2.2.1 Schema definition of RectifiedGridCoverage

The RectifiedGridCoverage is defined as:

```
<complexType name="RectifiedGridCoverageType">
  <complexContent>
    <restriction base="gml:AbstractCoverageType">
      <sequence>
        <group ref="gml:StandardObjectProperties"/>
        <element ref="gml:rectifiedGridDomain"/>
        <element ref="gml:rangeSet"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

D.2.2.1.1 rectifiedGridDomain

The rectifiedGridDomain is defined as:

```
<complexType name="RectifiedGridDomainType">
  <complexContent>
    <restriction base="gml:DomainSetType">
      <choice minOccurs="0">
        <element ref="gml:RectifiedGrid"/>
      </choice>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>
```

D.2.2.1.2 Image coordinates

The image coordinates uses a coordinate system with the dimensions *width* and *height* in that order, where (0, 0) is located in the upper left corner of the image.

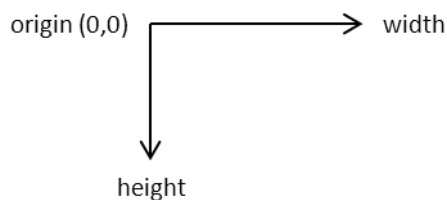


Figure D.3 – Image coordinate system

D.2.2.1.3 RectifiedGrid

Only one domain type is allowed, which is gml:RectifiedGrid:

```
<complexType name="RectifiedGridType" final="#all">
  <complexContent>
    <extension base="gml:GridType">
      <sequence>
        <element name="origin" type="gml:PointPropertyType"/>
        <element name="offsetVector" type="gml:VectorType" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

Element/attribute	Description
limits	An envelope for the grid in image coordinates (D.2.2.1.2).
origin	The origin in geo-coordinates. The origin is in the upper left corner of the grid, at the center of the upper left pixel . The axis order is defined by D.2.2.1.5.
offsetVector	The offsetVector gives the offset each pixel contributes to in each direction. Two offsetVector elements are required. The axis order is defined by D.2.2.1.5.
axisName	The axisName gives the name of the axis. Two axisName elements are required.
srsName	This attribute is used for referring to the reference system that is used. It is required in the <i>RectifiedGrid</i> by this profile. Allowed values are listed in D.2.3.

Table D.1 – Required RectifiedGrid content

D.2.2.1.4 Extent in metadata (ISO 19139)

This profile requires that the geographical extent information does not contradict the extent information given by the GMLJP2 coverage. ISO 19139 metadata can use GML for defining the geographical extent for a dataset. That means this requirement can be fulfilled by using the RectifiedGrid from previous paragraph in the metadata extent information.

Xpath to extent information in ISO 19139
/gmd:MD_Metadata/gmd:identificationInfo/gmd:MD_DataIdentification/gmd:extent/gmd:EX_Extent/gmd:geographicElement/gmd:EX_BoundingPolygon/gmd:polygon/gml:RectifiedGrid

Table D.2 – GML extent in ISO 19139 metadata

D.2.2.1.5 Axis order

The origin and the offsetVector, uses the specific axis order that is defined for the used reference system.

Reference system	Axis order
WGS84 LatLong	Left hand order (<Latitude> <Longitude>)
UTM	Right hand order (<Easting> <Northing>)

Table D.3 – Axis order for different reference systems

D.2.2.1.6 rangeSet

The rangeSet is defined as:

```
<complexType name="RangeSetType">
  <choice>
    <element ref="gml:File"/>
  </choice>
</complexType>
```

The rangeSet is the collection of values that the coverage covers. The gml:File element is the only relevant element in GMLJP2. This element is defined as:

```
<complexType name="FileType">
  <sequence>
    <element ref="gml:rangeParameters"/>
    <element name="fileName" type="anyURI"/>
    <element name="fileStructure" type="gml:FileValueModelType"/>
  </sequence>
</complexType>
```

Element	Description
gml:rangeParameters	This element describes the values of the file with GML elements. This element is mandatory in GML 3.1.1, but it is not mandatory to have any content in it, so this profile don't require any content in this element. The content is optional and can be freely chosen within the constraints imposed by GML 3.1.1.
fileName	A GMLJP2 URI pointing to the codestream that the coverage describes
fileStructure	This element is irrelevant for this profile, but it is mandatory in the gml:File element, so it must be used anyway. The only supported value in GML 3.1.1 is "Record Interleaved", so just use that string no matter what the real file structure is.

Table D.4 – Required file type content

D.2.2.2 A WGS84 LL example of a RectifiedGridCoverage:

```

<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:gmd="http://www.isotc211.org/2005/gmd"
  xmlns:gco="http://www.isotc211.org/2005/gco"
  xsi:schemaLocation="http://www.opengis.net/gml file:///D:/dgiwg/jp2/GML-
3.1.1/profiles/DGIWGgmlJP2Profile/1.1.0/DGIWGgmlJP2Profile.xsd">
  <gml:featureMember>
    <gml:FeatureCollection><!-- feature collection for a specific codestream -->
      <gml:featureMember>
        <gml:RectifiedGridCoverage>
          <gml:rectifiedGridDomain>
            <gml:RectifiedGrid dimension="2" srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:limits>
                <gml:GridEnvelope> <!-- Image coordinates -->
                  <gml:low>0 0</gml:low>
                  <gml:high>4999 9999</gml:high>
                </gml:GridEnvelope>
              </gml:limits>
              <gml:axisName>X</gml:axisName>
              <gml:axisName>Y</gml:axisName>
              <!-- The origin location in geo coordinates -->
              <gml:origin>
                <gml:Point>
                  <gml:pos>19.1234567 37.1234567</gml:pos>
                </gml:Point>
              </gml:origin>
              <!--offsetVectors says how much offset each pixel will contribute to, in practice, that is the cell size -->
              <gml:offsetVector>0.0 0.00001234</gml:offsetVector>
              <gml:offsetVector>-0.00001234 0.0</gml:offsetVector>
            </gml:RectifiedGrid>
          </gml:rectifiedGridDomain>
          <!--A RectifiedGridCoverage uses the rangeSet to describe the data below is a description of the range of values
described by the grid coverage -->
          <gml:rangeSet>
            <gml:File>
              <gml:rangeParameters></gml:rangeParameters>
              <gml:fileName>gmljp2://codestream/0</gml:fileName>
              <gml:fileStructure>Record Interleaved</gml:fileStructure>
            </gml:File>
          </gml:rangeSet>
        </gml:RectifiedGridCoverage>
      </gml:featureMember>
    </gml:FeatureCollection>
  </gml:featureMember>
</gml:FeatureCollection>

```

This example is a *RectifiedGridCoverage* describing the coverage of a rectangular raster with 10 000 pixel in the northing direction and 5000 pixels in the easting direction. The coverage is in Sudan, close to the coast. The coverage uses *WGS84 LL* coordinates, as indicated by the EPSG code (4326). The origin and the offset vectors uses left hand ordered coordinates. That means the latitude is 19.1234567 decimal degrees and the longitude is 37.1234567 decimal degrees.

D.2.2.3 A UTM example of a RectifiedGridCoverage:

```

<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:gmd="http://www.isotc211.org/2005/gmd"
  xmlns:gco="http://www.isotc211.org/2005/gco"
  xsi:schemaLocation="http://www.opengis.net/gml file:///D:/dgiwg/jp2/GML-
3.1.1/profiles/DGIWGgmlJP2Profile/1.1.0/DGIWGgmlJP2Profile.xsd">
  <gml:featureMember>
    <gml:FeatureCollection><!-- feature collection for a specific codestream -->
      <gml:featureMember>
        <gml:RectifiedGridCoverage>
          <gml:rectifiedGridDomain>
            <gml:RectifiedGrid dimension="2" srsName="urn:ogc:def:crs:EPSG::32637">
              <gml:limits>
                <gml:GridEnvelope> <!-- Image coordinates -->
                  <gml:low>0 0</gml:low>
                  <gml:high>4999 9999</gml:high>
                </gml:GridEnvelope>
              </gml:limits>
              <gml:axisName>X</gml:axisName>
              <gml:axisName>Y</gml:axisName>

              <!-- The origin location in geo coordinates -->
              <gml:origin>
                <gml:Point>
                  <gml:pos>302466.089 2115546.978</gml:pos>
                </gml:Point>
              </gml:origin>

              <!--offsetVectors says how much offset each pixel will contribute to, in practice, that is the cell size -->
              <gml:offsetVector>1.327588 0.0</gml:offsetVector>
              <gml:offsetVector>0.0 -1.372799</gml:offsetVector>
            </gml:RectifiedGrid>
          </gml:rectifiedGridDomain>

          <!--A RectifiedGridCoverage uses the rangeSet to describe the data below is a description of the range of values
          described by the grid coverage -->
          <gml:rangeSet>
            <gml:File>
              <gml:rangeParameters/>
              <gml:fileName>gmljp2://codestream/0</gml:fileName>
              <gml:fileStructure>Record Interleaved</gml:fileStructure>
            </gml:File>
          </gml:rangeSet>

        </gml:RectifiedGridCoverage>
      </gml:featureMember>
    </gml:FeatureCollection>
  </gml:featureMember>
</gml:FeatureCollection>

```

This example uses UTM coordinates. The area is in Sudan, close to the coast, like in the WGS84 example. The EPSG code (32637) indicates UTM zone 37 on the northern hemisphere. The origin and the offset vectors use right hand coordinates. That means the easting coordinate is 302466.089 m and the northing coordinate is 2115546.978 m.

D.2.3 *EPSG georeference in RectifiedGrid@srsName*

GML features can refer to coordinate reference systems with the attribute *srsName*. This profile mandates that the attribute *srsName* in the *RectifiedGrid* element shall be used for declaring reference system. This declaration shall apply globally to all GML features in the file.

The reference system used, is referred to by an OGC namespace with EPSG codes: *urn:ogc:def:<object type>:EPSG:<EPSG version>:<EPSG code>* where the following values are allowed by this profile:

EPSG code	Reference system
urn:ogc:def:crs:EPSG::4326	GCS_WGS84
urn:ogc:def:crs:EPSG::326zz and urn:ogc:def:crs:EPSG::327zz	UTM for each zone in the northern hemisphere respectively the southern hemisphere
urn:ogc:def:ellipsoid:EPSG::7030	WGS84 ellipsoid as a vertical reference
urn:ogc:def:crs:EPSG::5773	Earth Gravity Model 1996
urn:ogc:def:crs:EPSG::3855	Earth Gravity Model 2008

Table D.5 – Reference codes from the EPSG namespace

The table above doesn't use version numbers in the namespace references, but the profile allows version numbers. The profile does not mandate any particular version number. The EPSG codes used in the table will be used as above given reference systems no matter which version number is used.

D.2.4 *CRSDictionary.gml*

This is an optional box which can be used for defining a CRS in GML. With this box, it is possible to use a CRS which is not defined in any CRS namespace.

D.3 Annotation

Annotations are meant to point out and draw attention to something in the image. This profile uses labels and points/curves/areas of interest. The annotations use geocoordinates for their placement on the raster data. The annotations are visualized with common GML elements like *gml:Point*, *gml:LineString* or *gml:Polygon*. This profile requires usage of the same reference system as for the *RectifiedGrid* that makes the JPEG2000 file's georeference. A Class A or Class XA conformant image processor is required to reproject the annotations as well as the rest of the image while reprojecting the image.

D.3.1 Annotation elements

The annotation elements are associations between an annotation and an image, feature or region in the image, to which the annotation applies.

D.3.1.1 Annotation

All fundamental annotation blocks contains following elements:

Element	Description
pointer	<code>gml:CurvePropertyType</code> – For displaying an arrow or other type of symbol to point out something.
content	Can be a label, or in <i>Class XA</i> files, an image or a video clip
annotates	A geometry representing the region that is annotated. The geometry type depends on the type of annotation.
defaultStyle	The container element for all styling of the annotation. This element applies to either <i>pointer</i> or <i>annotates</i> within the same annotation element that contains the <i>defaultStyle</i> element.

Table D.6 – Annotation elements

D.3.2 Annotation types

There are four types of annotations:

Annotation type	Description
PointOfInterest	This type annotates a point. The <i>annotates</i> element uses a <i>gml:Point</i> .
CurveOfInterest	This type annotates a curve, like a road. The <i>annotates</i> element uses a <i>gml:LineString</i> .
RegionOfInterest	This type annotates a region. The <i>annotates</i> element uses a <i>gml:Polygon</i> .
Annotation	This type of annotation is a general annotation that is not intended to annotate a point/curve/region of interest in the image. This annotation type could be used for other purposes, such as visually mark the dataset with a serial number. This can use either <i>gml:Point</i> , <i>gml:LineString</i> , <i>gml:Polygon</i> or <i>gml:MultiSurface</i> .

Table D.7 – Annotation types

PointOfInterest, *CurveOfInterest* and *RegionOfInterest* are the annotations that shall be used to highlight some detail in the image. *Annotation* can be used for other purposes.

D.3.3 Content of the annotation

The annotation can contain a label, image or a video clip.

D.3.3.1 Label

The *Label* has text and anchor point:

Field	Description
<code>xima:textContent</code>	The textual content of the label.
<code>xima:anchorPoint</code>	Contains a <i>gml:Point</i> for the label.
<code>defaultStyle</code>	The container element for all styling of the annotation. In this case,

	the styling element applies to the <i>Label</i> element that contains the <i>defaultStyle</i> element.
--	--

Table D.8 – Label elements**D.3.3.2 Image**

The *Image* has a file URI and a boundary (for the border of the image):

Field	Description
xima:fileURI	Where the file is. This element uses URI:s of the type <i>DGIWGJP2 URI</i> when referring to images embedded in the JPEG2000 file. The images are embedded JPEG, PNG or GIF files. They are embedded in 'uuid' boxes in an annotation structure (C.2).
xima:boundary	The extent of the image given as a <i>gml:Envelope</i> using <i>lowerCorner</i> and <i>upperCorner</i> .

Table D.9 – Image elements**D.3.3.3 Video**

The *Video* element is outside the *GMLJP2* standard. This element is included in conformance class XA. Only an XA compliant reader can be expected to read this element. A video clip is not supposed to run at all times while using the JPEG2000 file. Instead, the video clip is supposed to be opened by user request. It is up to the user to play the video clip anywhere on the screen. That means the *Video* element needs only a file URI.

Field	Description
xima:fileURI	Where the file is. This element uses URI:s of the type <i>DGIWGJP2 URI</i> when referring to video clips embedded in the JPEG2000 file.

Table D.10 – Video element**D.3.4 Annotation styling**

Annotations can be styled with SVG symbols and CSS2 styling expression grammar.

D.3.4.1 Styling features

Both SVG and CSS2 support many different types of styling elements. The DGIWG GMLJP2 Profile does not attempt to make a profile of either SVG or CSS2. But the DGIWG GMLJP2 Profile does require that following properties should be able to control:

Property	Description and value domain
Font family	The type-face used (Helvetica, Arial, Times, etc.). The profile does not make any requirement regarding which font families that should be used. That is dependent on which font families that are installed on the system.
Font size	Use <i>typographical point</i> as the unit, as opposed to other styling measurements (commonly pixels), because the typographical point is already in common usage for font sizes. The size refers to the size on the screen which displays the image.
Font style	Following font styles should be available: <i>normal</i> <i>italic</i> <i>oblique</i> <i>bold</i>

Property	Description and value domain
Annotation colour	All kinds of annotation elements can use colours
Line thickness	The line thickness refers to the size on the screen measured in pixels.
Line pattern	Like dashed or dotted lines.
Fill pattern	Hatched fill pattern.

Table D.11 – Styling features

D.3.4.2 *gml:defaultStyle*

This is the container element for all styling. This element can be assigned to every GML feature, including the annotation elements. In this profile, the *defaultStyle* is assigned to an annotation by being defined in that annotation. The *defaultStyle* element can be a child element of:

Annotation type	Styling option
Annotation	Only <i>GeometryStyle</i> is applicable.
PointOfInterest	Only <i>GeometryStyle</i> is applicable.
CurveOfInterest	Only <i>GeometryStyle</i> is applicable.
RegionOfInterest	Only <i>GeometryStyle</i> is applicable.
Label	<i>Label</i> uses <i>LabelStyle</i> .

Table D.12 – Overview of styling options for different annotations

Note that all annotation types can have labels in their content. These labels can contain a label style.

D.3.4.2.1 *gml:FeatureStyle*

The *gml:FeatureStyle* element defines the styling of features, in this context only different annotations, since the *RectifiedGrid* should not be visualized. Following attributes are used:

Attribute	Description
queryGrammar	Specifies a query grammar used to define styles more precise than on a per feature type basis. The chosen query grammar is used for the child element <i>featureConstraint</i> . This profile uses “ <i>xpath</i> ”. The reason for having this attribute in the profile is that it is mandatory in general GML, and is therefore mandatory in this profile too.

Table D.13 – Attribute for the *gml:FeatureStyle* element

The child elements in this profile are either *gml:geometryStyle* or *gml:labelStyle*.

D.3.4.2.2 gml:GeometryStyle

The *gml:geometryStyle*, describes the appearance of geometries. This element will always contain a *gml:symbol*. GML 3.1.1 do allow the usage of a *gml:style* element, but also states that it is deprecated as early as in GML 3.1.0. The old *gml:style* element contained a CSS2 expression. The *gml:symbol* can also use CSS2. Following attributes are used:

Attribute	Description
geometryProperty	Specifies which property in an annotation that the style should apply to. It can be either <i>pointer</i> or <i>annotates</i> in this profile.
geometryType	It is necessary to specify which type of geometry that the property uses, because GML doesn't restrict that. This profile uses <i>gml:Point</i> , <i>gml:LineString</i> and <i>gml:Polygon</i> .

Table D.14 – Attributes for the *gml:geometryStyle* element**D.3.4.2.3 gml:symbol**

The *gml:symbol* element can refer to an SVG file, which in this profile can be embedded in the *gml.data* structure. This profile also allows inline SVG code. If it has inline SVG code, it can use a *style* attribute which uses a CSS2 expression, thus fulfilling the need that motivated the deprecated *style* element. Following attributes are used:

Attribute	Description
symbolType	Always “ <i>svg</i> ”.
xlink:href	Optional attribute for linking to an SVG symbol. For SVG symbols embedded in the <i>JPEG2000</i> file, a <i>GMLJP2 URI</i> should be used.

Table D.15 – Attributes for the *gml:symbol* element**D.3.4.2.4 gml:LabelStyle**

The *gml:LabelStyle* element is used if textual annotations should be rendered in a certain way. This element does not use the *gml:symbol* element, since it is irrelevant for textual annotations. Instead it uses the *gml:style* element, which is not deprecated in this context.

D.3.4.2.5 gml:style

This element uses CSS2 styling expression for styling the text content.

D.3.4.2.6 gml:label

The *gml:label* element uses an expression to compose a visual annotation of the content in *xima:textContent*. This profile allows only a *LabelExpression* element.

D.3.4.2.7 gml:LabelExpression

The *LabelExpression* element uses an *XPath* to select a property to display as a textual annotation. The *XPath* is either absolute or relative to the hierarchical level of the *gml:defaultStyle* element. This profile imposes such restrictions on the association between the annotations and their styling elements that the relative *XPath* will always be “.*xima:textContent*”.

D.4 Metadata

Metadata can be connected to any kind of GML entity with a *metadataProperty* element. At least, the *RectifiedGridCoverage* should have its *metadataProperty* element populated. This element is defined as:

```
<complexType name="GenericMetaDataType" mixed="true">
  <complexContent mixed="true">
    <extension base="gml:AbstractMetaDataType">
      <sequence>
        <any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

This element can contain metadata according to external schemas. The schemas from ISO19139 are used by this profile.

Annex E

Metadata

(normative)

E.1 Metadata location

This profile requires that the metadata may be either embedded in the GML georeference in the JPEG2000 file (connected to the *RectifiedGridCoverage* (D.4)) or supplied as an external file. There are good reasons for both practices. Metadata that is embedded in the file will not be lost even though a user forgets the external metadata file. Metadata in an external file enables a GIS system to handle metadata in the same way regardless of file format. Not all file formats allow metadata embedded in the file. If there is a conflict between the embedded metadata and the metadata in the external file, the embedded metadata will take precedence.

E.2 Metadata requirements

This profile has two requirements on the metadata.

E.2.1 *Metadata conforms to DMF*

Instead, the profile only requires that the metadata shall conform to DMF (DGIWG Metadata Foundation). For detailed requirements and conformance tests, this profile refers to the DMF work, in particular following parts:

Requirement class	Description
DMF/Core	Minimum set of metadata elements
DMF/Common	Additional set of metadata elements
DMF/Data	Metadata for datasets, series and tiles
DMF/Data+	Metadata for coverage results and a number of elements from ISO 19115-2 (extensions for imagery and gridded data)
DMF/Imagery	This class is not yet defined, but it is foreseen as an extension of DMF/Data for imagery.

Table E.1 – DMF requirement classes

E.2.2 *Coverage information in metadata*

Metadata that follows ISO 19139 usually contains information about the geographical extent of the dataset. The extent defined by the metadata, shall not contradict the extent given by the GML coverage in the GMLJP2 georeference. ISO 19139 has the possibility to use GML objects, such as the *RectifiedGrid*, which is also used in the *RectifiedGridCoverage* in GMLJP2.

E.3 Security classification

The part of the metadata that describes the security classification shall be stored in an *Intellectual Property Rights* box in addition to the security classification information in the ISO 19139 metadata. This is described more in C.1.5. The reason for this additional location is to ensure that the security classification information doesn't get lost from the GMLJP2 file. The only exception from this requirement is if the file is *unclassified*.

Annex F

XML Schemas

(normative)

This annex describes XML schemas that enable validation of the GML georeference as described in A.2.2. The schemas are available as separate appendices outside this document because of the size of the schemas.

F.1 GML Schemas

In order to validate a GML structure with these schemas, it is required to have all standard GML schemas downloaded locally. The standard GML schemas can be downloaded from:

http://schemas.opengis.net/gml/gml-3_1_1.zip

F.2 The DGIWG GMLJP2 schemas

The DGIWG GMLJP2 Profile schemas are used to validate that the GML structure follows this profile. The schemas can be downloaded from http://portal.dgiwg.org/files/?artifact_id=8105 by members of DGIWG. The schemas use elements from GML 3.1.1. In order for the schema validation to work, copy the profile schemas to following locations in the GML 3.1.1 folder structure:

Schema location
<GML Schema directory>/profiles/DGIWGgmlJP2Profile/1.0.0/DGIWGgmlJP2Profile.xsd
<GML Schema directory>/profiles/DGIWGgmlJP2Profile/1.0.0/annotation/DGIWGannotation.xsd

Table F.1 – Schema locations

F.2.1 DGIWGgmlJP2Profile.xsd

This schema is a modified version of the *gmlJP2Profile.xsd* which is published by OGC. The schema is a simplified version of the original schema. It imposes a lot of restrictions compared to the original.

F.2.2 DGIWGannotation.xsd

The annotation schema is a modified version of the *annotation.xsd* that is published by OGC. This schema is for the most part a simplified version of the original schema. Like the other schema, it imposes several restrictions. But this schema also adds the *Video* annotation element.

Annex G

Use cases (informative)

G.1 Use cases

This appendix lists the use cases for this profile:

UC#	Use case	Initiating actor	Receiving actor	Description
1	Raster types			
1.1	Gray tone	Raster image processor	File system	Raster image with a single band is compressed.
1.2	RGB	Raster image processor	File system	Colour raster image with three bands denoting Red, Green and Blue respectively, is compressed.
1.3	Multispectral	Raster image processor	File system	A multispectral image is compressed.
1.4	Discrete raster	Raster image processor	File system	An image with large solid colour areas is compressed.
1.5	Palletized image	Raster image processor	File system	An image which uses a palette for the colours, i.e. indexed colours, is compressed.
1.6	Elevation grid	Raster image processor	File system	An elevation grid is compressed.
1.7	Raster map	Raster image processor	File system	A thematic map where the pixel values denotes different objects in the map.
2	GML			
2.1	Geo-rectified	Photogrammetric expert	User	A geo-rectified image has its geo-rectification parameters encoded in GML.
2.2	Ortho-rectified	Photogrammetric expert	User	An ortho-rectified image has its ortho-rectification parameters encoded in GML.
3	Metadata			
3.1	Embed metadata	Geodata handler	User	The image's metadata is embedded in the image file.
3.2	Extract metadata	User	User	The image's metadata is read from the image file.
4	Annotations			
4.1	Annotate an image	Image interpreter	User	An image interpreter annotates an image. The image and its annotations are read by a user.
4.2	Annotate a set of related images	Image interpreter	User	Image interpreters annotate a set of related images. The images and their annotations are assembled into a package. The user read the images and the annotations from the package.
4.3	Annotate a stereo pair	Image interpreter	User	Image interpreter annotates images in a stereo pair.
4.4	Annotate a set of images in a triangulation block	Image interpreter	User	Image interpreter annotates images in a triangulation block.
4.5	Archive a set of images with annotations	User	DB or file archive	User archives a set of images and their annotations.
4.6	Reproject an image with annotations in geo-coordinates	User	File system	An image with annotations should be reprojected to a different reference system. The annotation location should be reprojected as well as the image coverage.

UC#	Use case	Initiating actor	Receiving actor	Description
5	Data access			
5.1	Query WMS or WCS and getting a GMLJP2 file as a response	WMS- or WCS-client	WMS- or WCS-server	An OWS client sends a GetMap request to a WMS-server or a GetCoverage request to a WCS-server, and gets a GMLJP2 file as a response.
5.2	Query metadata from a JPIP-server	JPIP-client	JPIP-server	A JPIP-client request metadata from a JPIP-server, and gets a JPIP data-bin containing only metadata.

Table G.1 – Use case

Annex H

Extensions used by DGIWG Profile (informative)

H.1 Extensions used by DGIWG Profile

Both GMLJP2 and the DGIWG profile of GMLJP2 uses extensions from ISO/IEC 15444-2.

H.1.1 *Extensions required by baseline GMLJP2*

The extensions listed in this paragraph, are used for georeferencing the image with GML. A JPEG2000 reader which conforms to ISO/IEC 15444-1 is able to render the codestream. It is only the GML georeference that would be ignored.

H.1.1.1 *Association box*

The purpose of the association box is to associate certain boxes with other boxes into logical groups. It is possible to consider the association box as a *folder* in a file system. GMLJP2 uses this mechanism to group together different boxes with GML data.

H.1.1.2 *Label box*

The label box contains only text. GMLJP2 uses the label box to identify certain GML instances in a JPEG2000 file.

H.1.1.3 *Reader requirement box*

The reader requirement box specifies which features that have been used in a file and which features the reader must support to fully use the file.

H.1.2 *Additional extensions used by DGIWG GMLJP2*

The extensions listed in this paragraph, are used for *compositing layers*, which is a feature for combining different codestreams, and for MPEG-4 video, which is used for video annotations. A JPEG2000 reader which conforms to ISO/IEC 15444-1 is unable to render the compositing layer. The reader will only read the first codestream. That means that if a file uses any of the Part 2 options for opacity (5.5.2 and 5.5.3), the reader wouldn't render the opacity.

H.1.2.1 *Opacity box*

The opacity box gives the option to use chroma-key based transparency (ISO/IEC 15444-2, M.11.7.6).

H.1.2.2 *Compositing layer header box*

The compositing layer box is required for combining different codestreams (ISO/IEC 15444-2, M.11.7).

H.1.2.3 *Codestream registration box*

Codestreams that shall be included in a compositing layer must be registered in the codestream registration box (ISO/IEC 15444-2, M.11.7.7).

H.1.2.4 *MPEG-4 boxes*

This paragraph applies only to *Class XA* files with embedded MPEG-4 video. It is beyond the scope of this work to make a profile for MPEG-4. Therefore, this paragraph will only say that

the superbox '*moov*' should be there containing other boxes that are necessary for MPEG-4-encoded video.

Annex I

Software used for testing purposes (informative)

I.1 Software recommendations

Examination of GMLJP2 files can be done with the help of a few different software packages.

I.2 Kdu_show

Kdu_show let the user inspect a JPEG2000 file. The program is included in the demonstration package from *Kakadu Software* which can be downloaded from:

http://www.kakadusoftware.com/index.php?option=com_content&task=view&id=26&Itemid=22

This program has a metadata viewer (opened by *Metadata* -> *Open Metashow*), which can be used for inspecting the metadata that is stored in the JPEG2000 boxes, which includes the GML georeference.

The program can also display the codestream properties (*File – Properties*) for viewing information about the codestream.

I.3 XML Spy

XML Spy is a tool from one of the leading XML tool developers, Altova. This tool is used for validating the GML georeference. *XML Spy* can't validate the GML georeference in a GMLJP2 file directly. In order to validate the GML georeference, the user have to use a tool, such as *kdu_show*, which lets the user copy and paste the GML code to an appropriate location for the validation, i.e. an xml file.

I.3.1 XML Schema reference

Note that in order to validate an XML file with *XML Spy*, it is necessary to add a reference to the XML schema that the file shall follow. Therefore the tester has to add the following attribute to the root element in the GML georeference:

```
xsi:schemaLocation="http://www.opengis.net/gml file:///D:/dgiwg/jp2/GML-3.1.1/profiles/DGIWGgmlJP2Profile/1.0.0/DGIWGgmJP2Profile.xsd"
```

(the file path in bold, shall be changed to point to the files location on the particular system where the validation takes place).

I.4 Microsoft .Net Framework 4.0

This is not a particular application ready to run, but rather a framework for building applications. The XML library in this framework can validate XML code. In this case, it is not necessary to add a schema reference to the root element. In fact, it is not even necessary to copy the GML code into an XML file. It is sufficient to let the application built with *.Net Framework 4.0* read the GML code into a string, and perform an XML validation on that string. It requires a great deal of software development from the user, but once done, the user could have a tool which lets the user batch validate a large number of GMLJP2 files.

Bibliography

David S. Taubman, Michael W. Marcellin. 2002. *JPEG2000 – Image Compression Fundamentals, Standards and Practice*, third printing, Norwell, Massachusetts: Kluwer Academic Publishers.

Peter Schelkens, Athanassios Skodras, Touradj Ebrahimi. 2009. *The JPEG 2000 Suite*, John Wiley & Sons Ltd.